User's Manual

32

RH850/F1M Group

User's Manual: Hardware

Renesas microcontroller
RH850 Family

Renesas Electronics
www.renesas.com

Rev.1.03 May 2016

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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended 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; and industrial robots etc.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

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

Notes for CMOS devices

(1) 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 VIL (MAX) and VIH (MIN) due to noise, etc., 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 VIL (MAX) and VIH (MIN).

(2) Handling of unused input pins:

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to power supply or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) Precaution against ESD:

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that 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 should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

(4) Status before initialization:

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

(5) Power ON/OFF sequence:

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

(6) Input of signal during power off state:

Do not input signals or an I/O pull-up power supply while the device is not powered. 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. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

How to Use This Manual

Readers This manual is intended for users who wish to understand the functions of the RH850/F1M and design application systems using the RH850/F1M microcontrollers.

Purpose This manual is intended to give users an understanding of the hardware functions of the RH850/F1M shown in the *Organization* below.

Organization This manual is divided into two parts: Hardware (this manual) and Software (RH850G3M User's Manual: Software).

Hardware
Pin functions
CPU function
On-chip peripheral functions
Flash memory programming

Software
Overview
Processor Model
Register Reference
Exceptions and Interrupts
Memory Management
Instruction Reference
Reset
Appendix

How to read this manual It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

To understand the overall functions of the RH850/F1M

→ Read this manual according to the Contents.

To understand the details of an instruction function

→ See RH850G3M User's Manual: Software available separately.

The "yyy bit of the xxx register" is described as the "xxx.yyy bit" in this manual. Note with caution that if "xxx.yyy" is described as is in a program, however, the compiler/assembler cannot recognize it correctly.

Conventions

Data significance: Higher digits on the left and lower digits on the right

Active low representation: xxx (overscore over pin or signal name)

Memory map address: Higher addresses on the top and lower addresses on the bottom

Note: Footnote for item marked with Note in the text

Caution: Information requiring particular attention

Remark: Supplementary information

Numeric representation: Binary ... xxxx or $xxxx_B$

Decimal ... xxxx

Hexadecimal ... $xxxx_H$

Prefix indicating power of 2 (address space, memory capacity):

K (kilo): $2^{10} = 1,024$

M (mega): $2^{20} = 1,024^2$

G (giga): $2^{30} = 1,024^3$

Description of Registers

Each register description includes the register access, register address, and register value after a reset, a bit chart illustrating the arrangement of bits and a table of bits, describing the meaning of the bit settings.

The standard format for bit charts and tables is shown below.

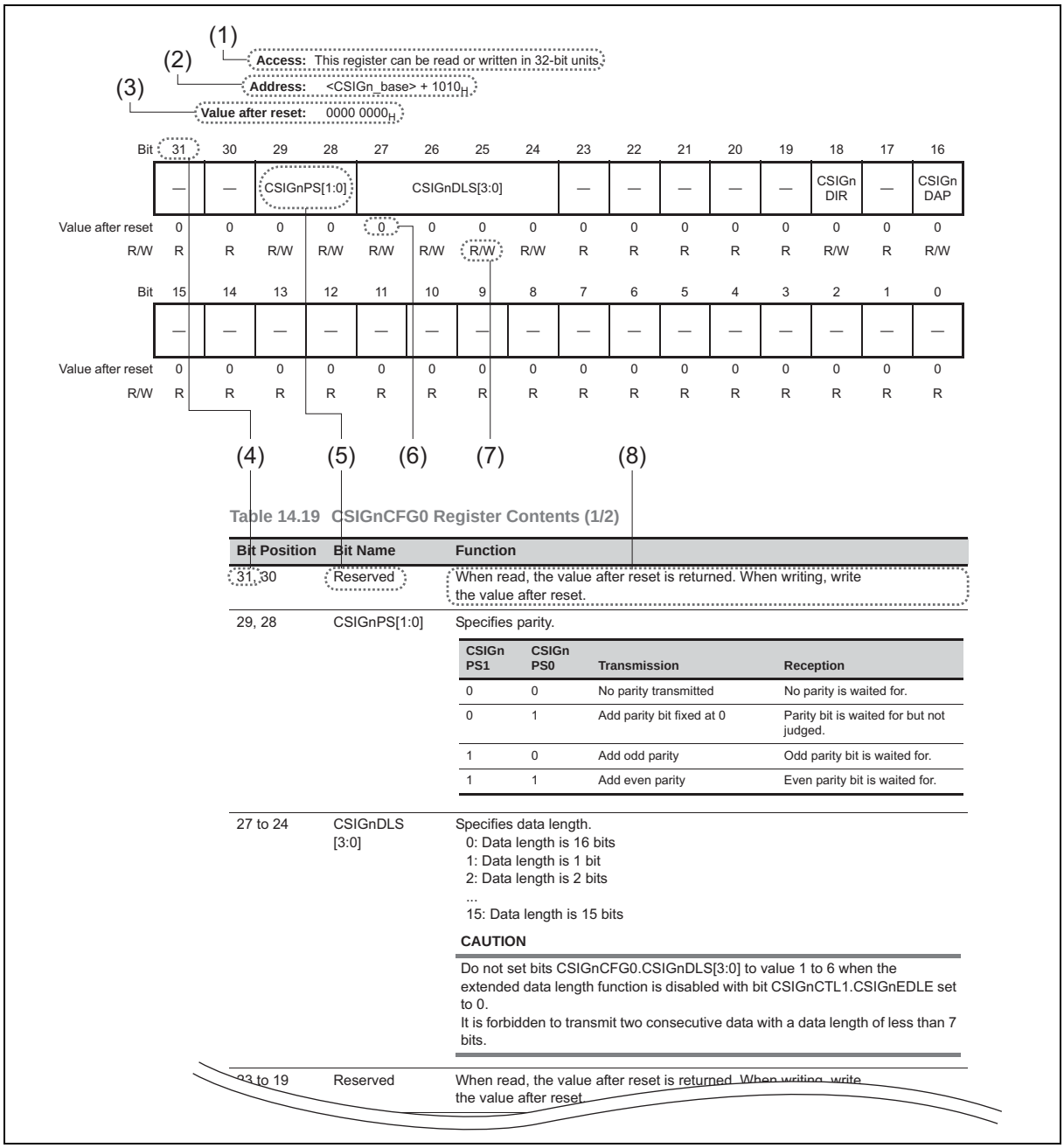


Table 14.19 CSIGnCFG0 Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29, 28	CSIGnPS[1:0]	Specifies parity.
27 to 24	CSIGnDLS [3:0]	Specifies data length. 0: Data length is 16 bits 1: Data length is 1 bit 2: Data length is 2 bits ... 15: Data length is 15 bits **CAUTION** Do not set bits CSIGnCFG0.CSIGnDLS[3:0] to value 1 to 6 when the extended data length function is disabled with bit CSIGnCTL1.CSIGnEDLE set to 0. It is forbidden to transmit two consecutive data with a data length of less than 7 bits.
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CSIGn PS1	CSIGn PS0	Transmission	Reception
0	0	No parity transmitted	No parity is waited for.
0	1	Add parity bit fixed at 0	Parity bit is waited for but not judged.
1	0	Add odd parity	Odd parity bit is waited for.
1	1	Add even parity	Even parity bit is waited for.

(1) Access

The register can be accessed in the bit unit indicated here.

(2) Address

This is the register address.
For base address, see description of base address in each section.

(3) Value after a reset (in hexadecimal notation)

This is the value of all bits of the register after a reset. Values for bytes are given as numbers in the range from 0 to 9 and letters from A to F or as X where they are undefined.

(4) Bit position

This is the bit number.
The bits are numbered from 31 to 0 for 32-bit registers, 15 to 0 for 16-bit registers, and 7 to 0 for 8-bit registers.

(5) Bit name

Bit name or field name is indicated.
If it is required to clearly identify the digits of a bit field, this is done by using a form such as CSIGnDLS[3:0] in the example above.
Reserved bits are indicated by a dash (—).

(6) Value after a reset (in binary notation)

This is the bit value after a reset.

0	:	The value after a reset is 0.
1	:	The value after a reset is 1.
—	:	The value after a reset is undefined.

(7) R/W

This is the bit attribute of all bits of the register.

R/W	:	The bit or field is readable and writable.
R	:	The bit or field is readable. Note that all reserved bits are indicated as R. When written, the value specified in the bit chart or the value after a reset should be written.
W	:	This bit or field is writable. When read, the value is undefined. If a value is indicated in the bit chart, the value is returned.

(8) Function

This is function of the bit.

Table of Contents

Section 1 Overview

1.1 RH850/F1M Product Features

The features of the RH850/F1M are described below.

The RH850/F1M is a 32-bit single-chip microcontroller with one G3M core. The key features of the F1M are low power consumption, high computational processing power, and a wide variety of internal peripheral functions. To reduce supply current in a variety of applications, a wide range of power reducing measures are available. For example, there is a Low Power Sampler (LPS), that can poll signals input to the analog and digital input pins without using the CPU, and DeepSTOP mode, in which the power supply to the most circuits of the microcontroller can be turned off.

Applications

The RH850/F1M is ideal for automotive electronics such as BCM (Body Control Module), gateway, HVAC, lighting modules, and many others.

1.1.1 RH850/F1M Functions

Table 1.1 Overview of Product (1/2)

Product Name			RH850/F1M		
			144 pin	176 pin	233 pin
Memory			See **Table 1.2, Product Lineup**.		
External memory interface (MEMC)			Not provided	Provided (20 bit Address Bus)	
CPU	CPU System		G3M (Single Core)		
	CPU Frequency		120 MHz Max.		
	FPU		Double-Precision		
	Flash Cache		8 KB / 4 way		
	Protection Function	Memory Protection Unit (MPU)	Provided		
		Internal Peripheral-device Guard (IPG)	Not provided		
		Processor Element Guard (PEG)	Provided		
DMA			16 ch		
Operating clock	Main Oscillator (MainOSC)		8 to 24 MHz		
	Low Speed Internal Oscillator (LS IntOSC)		240 kHz (typ.)		
	High Speed Internal Oscillator (HS IntOSC)		8 MHz (typ.)		
	PLL	for CPU clock (with SSCG)	120 MHz (Max.)		
		for Peripheral clock	80 MHz (Max.)		
	Sub Oscillator (SubOSC)		32.768 kHz		
I/O port			118	148	178
A/D converter	ADC0	Physical input channels	34 ch (12 bit resolution: 16 ch + 10 bit resolution 18 ch)	34 ch (12 bit resolution: 16 ch + 10 bit resolution 18 ch)	34 ch (12 bit resolution: 16 ch + 10 bit resolution 18 ch)
		External multiplexer support for channel number extension	Provided	Provided	Provided
		Channels with T&H	6 ch	6 ch	6 ch
	ADC1	Physical input channels	12 ch (12 bit resolution: 8 ch + 10 bit resolution 4 ch)	24 ch (12 bit resolution: 16 ch + 10 bit resolution 8 ch)	36 ch (12 bit resolution: 16 ch + 10 bit resolution 20 ch)
		External multiplexer support for channel number extension	Not provided	Not provided	Not provided
		Channels with T&H	Not provided	Not provided	Not provided
Timer	Timer Array Unit D (TAUD)		1 unit (16 bit resolution timers × 16 channels /unit)		
	Timer Array Unit B (TAUB)		1 unit (16 bit resolution timers × 16 channels /unit)	2 units (16 bit resolution timers × 16 channels /unit)	
	Timer Array Unit J (TAUJ)		2 units (32 bit resolution timers × 4 channels /unit)		
	Operating System Timer (OSTM)		5 units		
	Real-Time Counter (RTCA)		1 unit		
	Encoder Timer (ENCA)		1 unit		
	Window Watchdog Timer A (WDTA)		2 units		

Table 1.1 Overview of Product (2/2)

Product Name			RH850/F1M		
			144 pin	176 pin	233 pin
Serial interfaces	Clocked Serial Interface G (CSIG)		2 ch		
	Clocked Serial Interface H (CSIH)		4 ch		
	LIN Master Interface (RLIN2)		6 ch	10 ch	
	LIN/UART Interface (RLIN3)		6 ch		
	I2C Interface (RIIC)		1 ch		
	CAN Interface (RSCAN)		6 ch		
	FlexRay Interface (FLXA)		2 ch (A ch, B ch)		
	EthernetAVB (ETNB)		Not provided		
External Interrupts	Maskable (INTP)		16 ch		
	Non-maskable (NMI)		1 ch		
Other functions	Clock Monitors (CLMA)		For PLL, HS IntOSC, MainOSC		
	Data CRC (DCRA)		4 ch		
	Low-Voltage Indicator (LVI)		Provided		
	Power-On-Clear (POC)		Provided		
	Core Voltage Monitors (CVM)		Provided		
	Error Correction Coding (ECC)		For code flash, data flash, local RAM, retention RAM, cache RAM, tag RAM, CSIH, RS-CAN, FLXA		
	Low Power Sampling (LPS)		Provided		
	PWM diagnosis (PWM-Diag)		64 ch	72 ch	80 ch
	Motor Control		1 unit		
	Key Return (KR)		8 ch		
	CLOCK OUTPUT (FOUT)		Provided		
	RESET OUTPUT ($\overline{\text{RESETOUT}}$)		Provided		
	ICUSC*3		Not provided (ECO)*1 Provided (ADVANCED)*1		
	SWDTA*4		Not provided (ECO)*1 Provided (ADVANCED)*1		
	On-Chip debug (OCD)		Provided		
	Boundary Scan		Provided		
Voltage supply	Internal supply		V_{POC}*2 to 5.5 V		
	Input/output buffer supplies		V_{POC}*2 to 5.5 V		
	A/D Converter supplies		3.0 to 5.5 V		
Package	Operating Temperature (Ta)	–40°C to +105°C:	LQFP	LQFP	FPBGA
		–40°C to +125°C:			

Note 1. ECO (products in the eco range) is the product having a RSCAN.
ADVANCED (products in the advanced range) is the product having a RSCAN, an ICUSC and SWDTA.
Refer to **Table 1.2, Product Lineup**, for details.

Note 2. V_{POC} is the voltage level of power-on-clear circuit.
Refer to the data sheet for detailed specifications of electrical values.

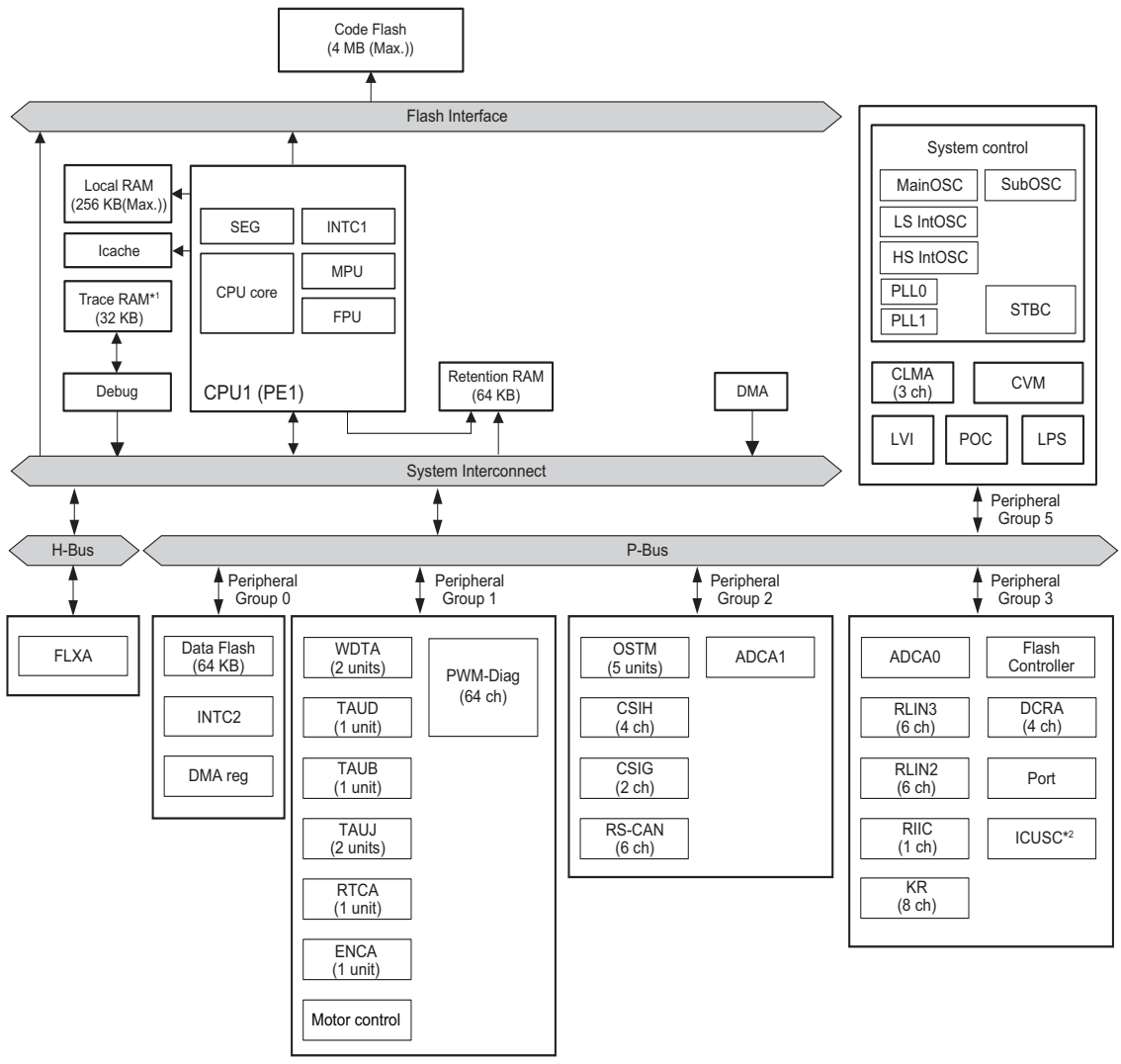
Note 3. ICUSC: Intelligent Cryptographic Unit Slave C

Note 4. SWDTA: Secure Watchdog Timer A

Table 1.2 Product Lineup

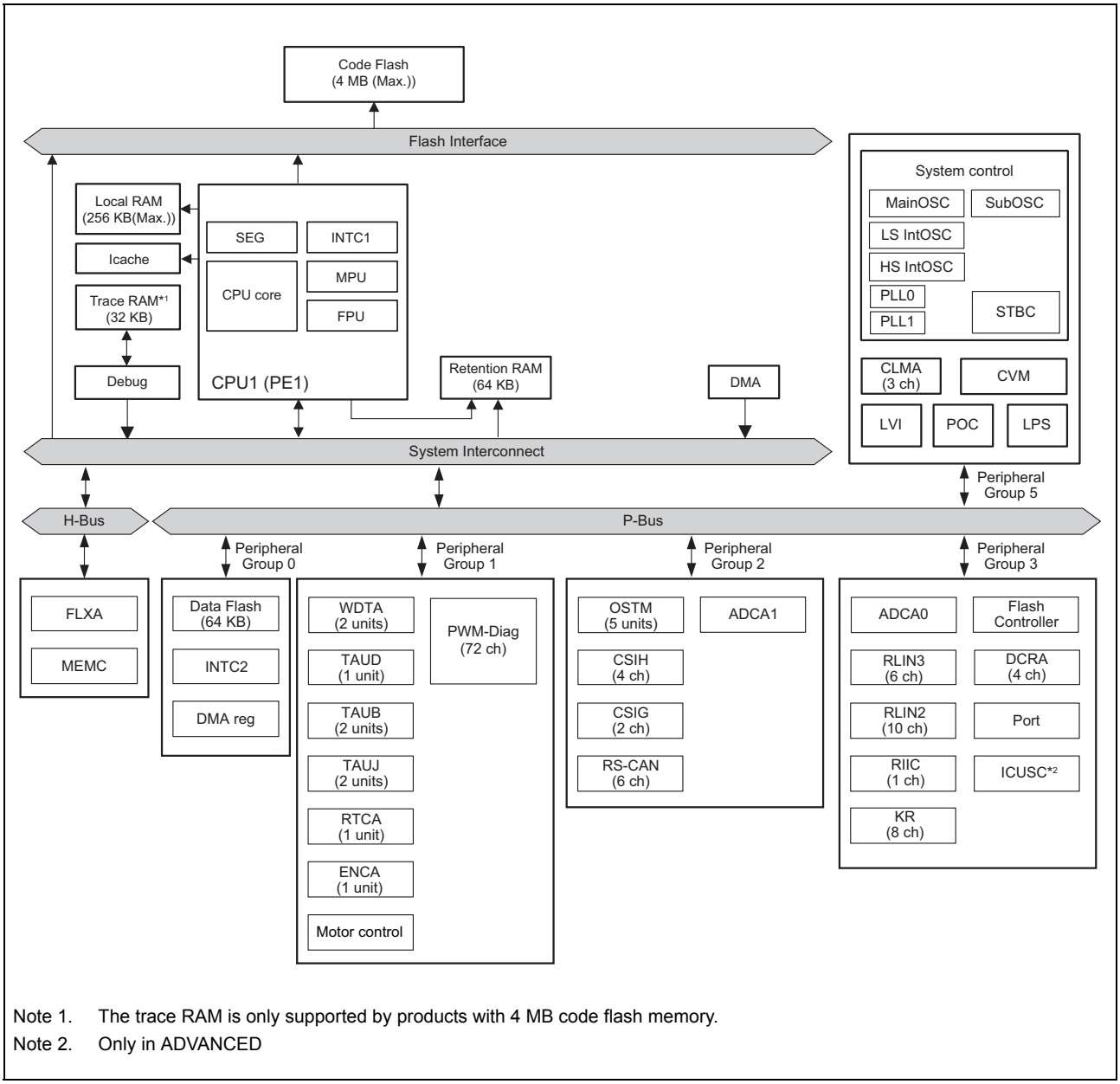
Pin Count	Memory					ICUS	CAN Interface	Part Name		Line Name
	Code Flash	CPU1 (Core #1)	Global Memory					Operating Temperature (Ta)		
		Local RAM	Data Flash	Global RAM	Retention RAM (RRAM)			–40°C to +105°C	–40°C to +125°C	
144 pin	3 M	192 KB	64 KB	Not provided	64 KB	Not provided	RS-CAN	R7F7015443AFP	R7F7015444AFP	ECO
	4 M	256 KB	64 KB	Not provided	64 KB	Not provided	RS-CAN	R7F7015453AFP	R7F7015454AFP	ECO
176 pin	3 M	192 KB	64 KB	Not provided	64 KB	Not provided	RS-CAN	R7F7015483AFP	R7F7015484AFP	ECO
	4 M	256 KB	64 KB	Not provided	64 KB	Not provided	RS-CAN	R7F7015493AFP	R7F7015494AFP	ECO
233 pin	3 M	192 KB	64 KB	Not provided	64 KB	Not provided	RS-CAN	R7F7015523ABG	R7F7015524ABG	ECO
	4 M	256 KB	64 KB	Not provided	64 KB	Not provided	RS-CAN	R7F7015533ABG	R7F7015534ABG	ECO
144 pin	3 M	192 KB	64 KB	Not provided	64 KB	ICUSC	RS-CAN	R7F7015643AFP	R7F7015644AFP	ADVANCED
	4 M	256 KB	64 KB	Not provided	64 KB	ICUSC	RS-CAN	R7F7015653AFP	R7F7015654AFP	ADVANCED
176 pin	3 M	192 KB	64 KB	Not provided	64 KB	ICUSC	RS-CAN	R7F7015683AFP	R7F7015684AFP	ADVANCED
	4 M	256 KB	64 KB	Not provided	64 KB	ICUSC	RS-CAN	R7F7015693AFP	R7F7015694AFP	ADVANCED
233 pin	3 M	192 KB	64 KB	Not provided	64 KB	ICUSC	RS-CAN	R7F7015723ABG	R7F7015724ABG	ADVANCED
	4 M	256 KB	64 KB	Not provided	64 KB	ICUSC	RS-CAN	R7F7015733ABG	R7F7015734ABG	ADVANCED

1.1.2 Internal Block Diagram



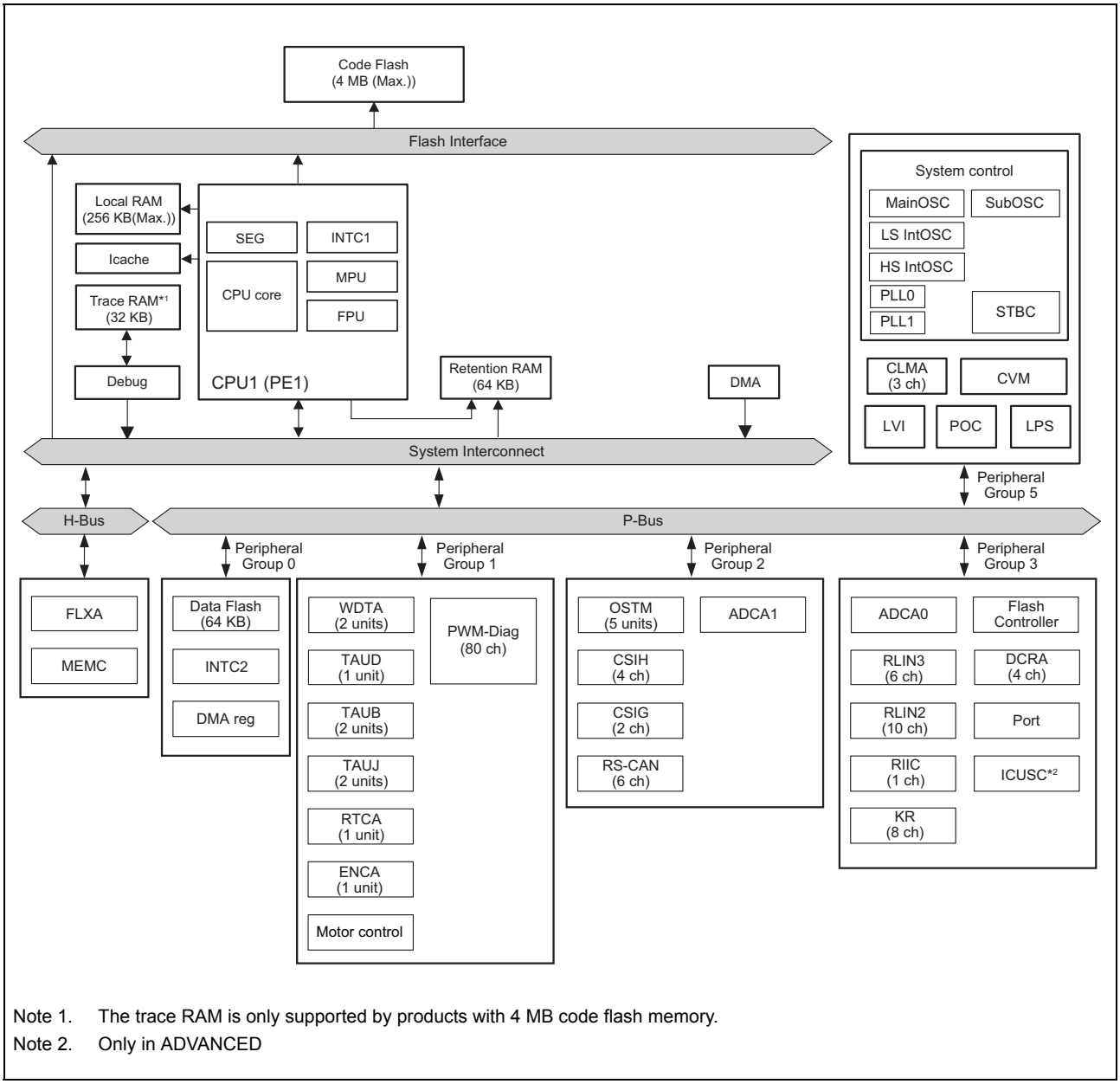
Note 1. The trace RAM is only supported by products with 4 MB code flash memory.
Note 2. Only in ADVANCED

Figure 1.1 Internal Block Diagram (RH850/F1M 144 pin)



Note 1. The trace RAM is only supported by products with 4 MB code flash memory.
Note 2. Only in ADVANCED

Figure 1.2 Internal Block Diagram (RH850/F1M 176 pin)



Note 1. The trace RAM is only supported by products with 4 MB code flash memory.
Note 2. Only in ADVANCED

Figure 1.3 Internal Block Diagram (RH850/F1M 233 pin)

Section 2 Pin Function

This section describes the pin and port functions.

Section 2.1 to **Section 2.5** describe the pin connections and respective pins.

Section 2.6 to **Section 2.13** describe the general port functions.

2.1 Pin Connection Diagram

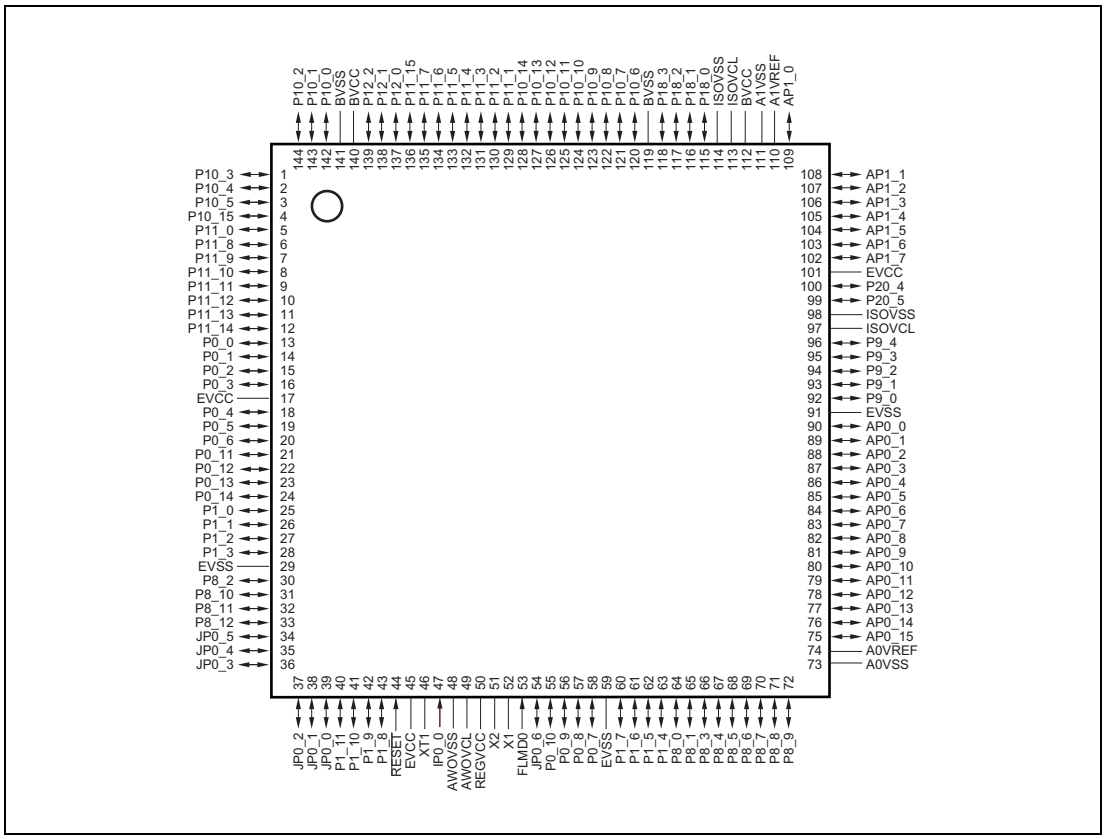


Figure 2.1 Pin Connection Diagram (144 pin LQFP)

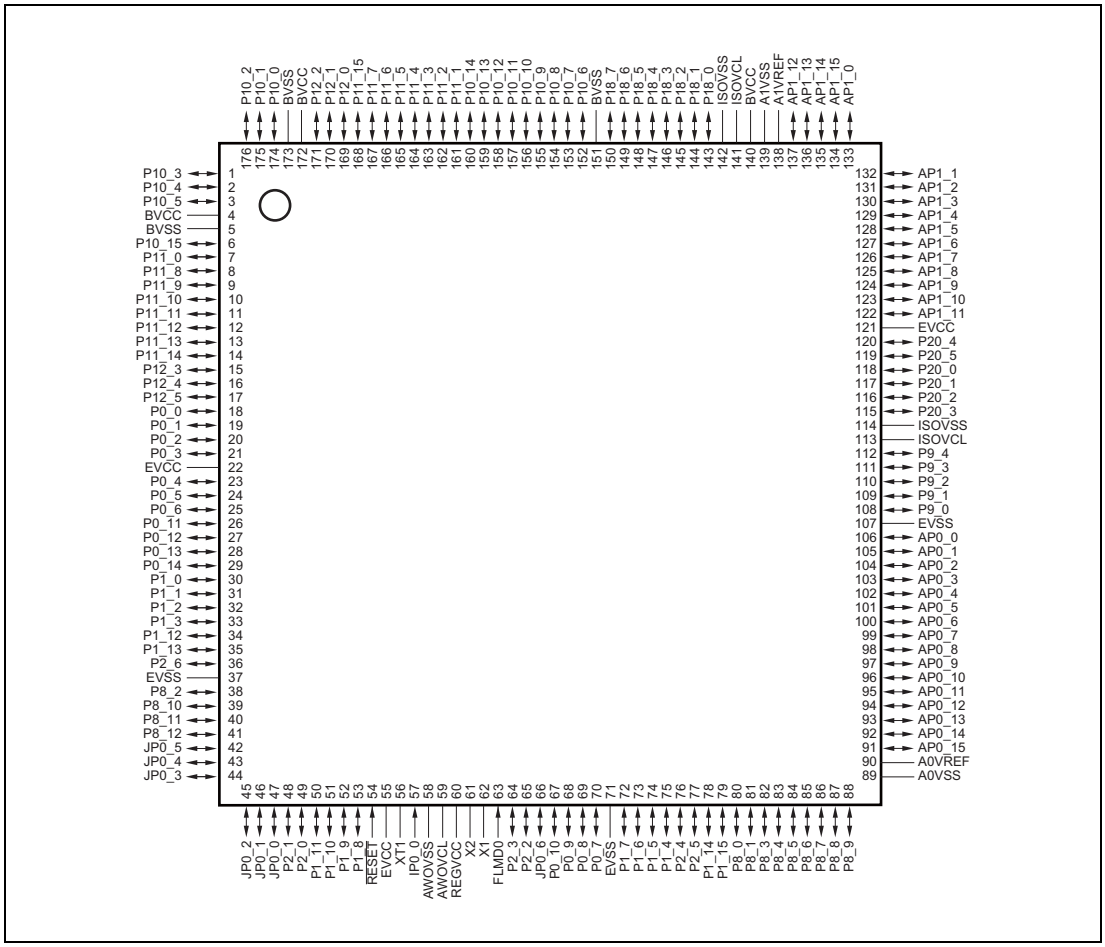
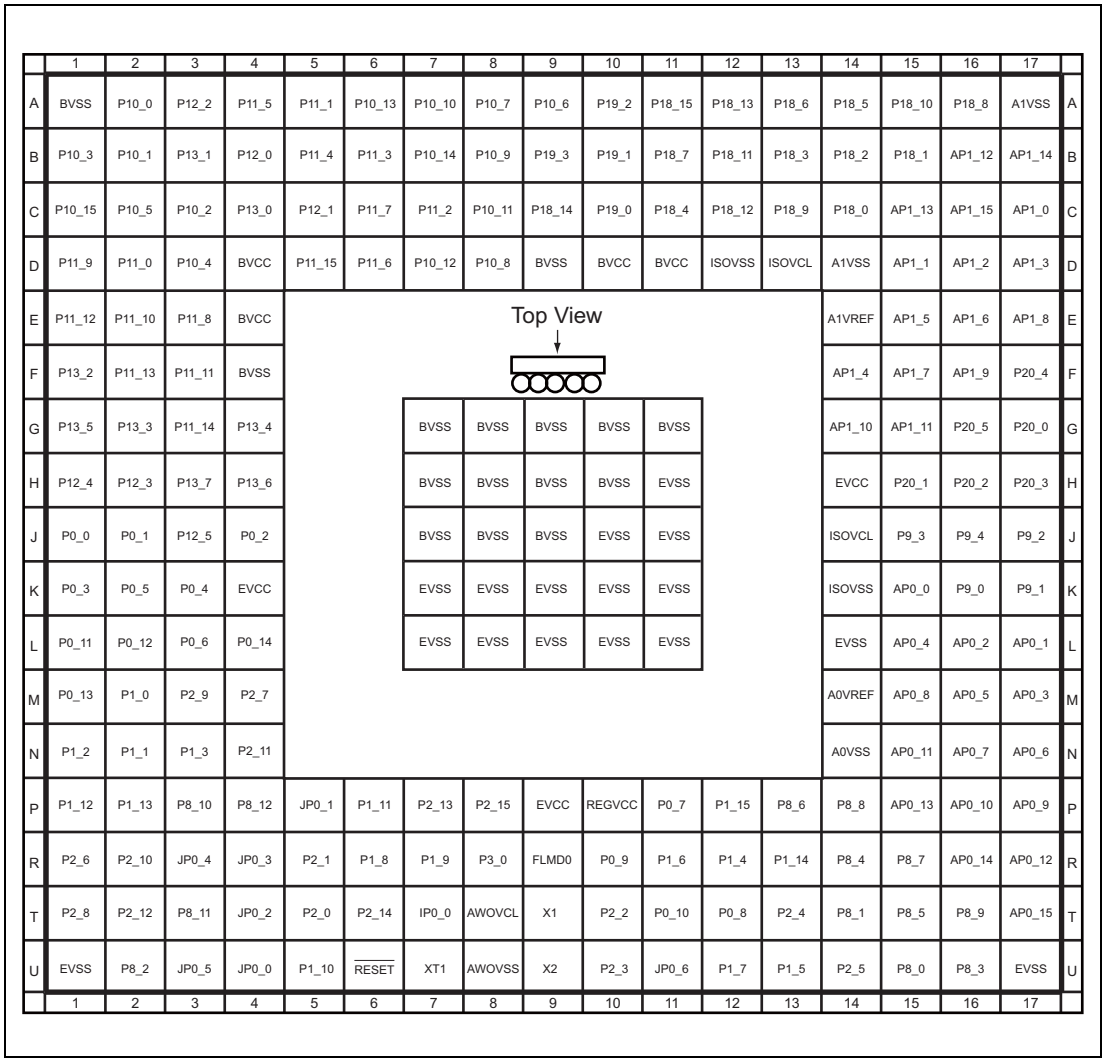


Figure 2.2 Pin Connection Diagram (176 pin LQFP)



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	BVSS	P10_0	P12_2	P11_5	P11_1	P10_13	P10_10	P10_7	P10_6	P19_2	P18_15	P18_13	P18_6	P18_5	P18_10	P18_8	A1VSS	A
B	P10_3	P10_1	P13_1	P12_0	P11_4	P11_3	P10_14	P10_9	P19_3	P19_1	P18_7	P18_11	P18_3	P18_2	P18_1	AP1_12	AP1_14	B
C	P10_15	P10_5	P10_2	P13_0	P12_1	P11_7	P11_2	P10_11	P18_14	P19_0	P18_4	P18_12	P18_9	P18_0	AP1_13	AP1_15	AP1_0	C
D	P11_9	P11_0	P10_4	BVCC	P11_15	P11_6	P10_12	P10_8	BVSS	BVCC	BVCC	ISOVSS	ISOVCL	A1VSS	AP1_1	AP1_2	AP1_3	D
E	P11_12	P11_10	P11_8	BVCC										A1VREF	AP1_5	AP1_6	AP1_8	E
F	P13_2	P11_13	P11_11	BVSS										AP1_4	AP1_7	AP1_9	P20_4	F
G	P13_5	P13_3	P11_14	P13_4				BVSS	BVSS	BVSS	BVSS	BVSS		AP1_10	AP1_11	P20_5	P20_0	G
H	P12_4	P12_3	P13_7	P13_6				BVSS	BVSS	BVSS	BVSS	EVSS		EVCC	P20_1	P20_2	P20_3	H
J	P0_0	P0_1	P12_5	P0_2				BVSS	BVSS	BVSS	EVSS	EVSS		ISOVCL	P9_3	P9_4	P9_2	J
K	P0_3	P0_5	P0_4	EVCC				EVSS	EVSS	EVSS	EVSS	EVSS		ISOVSS	AP0_0	P9_0	P9_1	K
L	P0_11	P0_12	P0_6	P0_14				EVSS	EVSS	EVSS	EVSS	EVSS		EVSS	AP0_4	AP0_2	AP0_1	L
M	P0_13	P1_0	P2_9	P2_7										A0VREF	AP0_8	AP0_5	AP0_3	M
N	P1_2	P1_1	P1_3	P2_11										A0VSS	AP0_11	AP0_7	AP0_6	N
P	P1_12	P1_13	P8_10	P8_12	JP0_1	P1_11	P2_13	P2_15	EVCC	REGVCC	P0_7	P1_15	P8_6	P8_8	AP0_13	AP0_10	AP0_9	P
R	P2_6	P2_10	JP0_4	JP0_3	P2_1	P1_8	P1_9	P3_0	FLMD0	P0_9	P1_6	P1_4	P1_14	P8_4	P8_7	AP0_14	AP0_12	R
T	P2_8	P2_12	P8_11	JP0_2	P2_0	P2_14	IP0_0	AWOVCL	X1	P2_2	P0_10	P0_8	P2_4	P8_1	P8_5	P8_9	AP0_15	T
U	EVSS	P8_2	JP0_5	JP0_0	P1_10	$\overline{\text{RESET}}$	XT1	AWOVSS	X2	P2_3	JP0_6	P1_7	P1_5	P2_5	P8_0	P8_3	EVSS	U
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

Figure 2.3 Pin Connection Diagram (233 pin BGA)

Table 2.1 Pin Assignment 144 pin LQFP (1/4)

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / $\overline{\text{CSIH1SSI}}$
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
4	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9
5	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11
6	P11_8 / $\overline{\text{CSIG1SSI}}$ / RLIN35TX / PWGA48O
7	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O
8	P11_10 / CSIG1SC / PWGA50O
9	P11_11 / CSIG1SI / RLIN25TX / PWGA51O
10	P11_12 / RLIN25RX / PWGA52O
11	P11_13 / RLIN24RX / PWGA53O
12	P11_14 / RLIN24TX / PWGA54O
13	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / $\overline{\text{RESETOUT}}$
14	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
15	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
16	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / INTP10
17	EVCC
18	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 /DPIN8
19	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO
20	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
21	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / PWGA34O
22	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI
23	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
24	P0_14 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
25	P1_0 / RLIN33RX / INTP13
26	P1_1 / RLIN33TX
27	P1_2 / CAN3RX / INTP3
28	P1_3 / CAN3TX / DPIN23
29	EVSS
30	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
31	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / ADCA0I17S
32	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / ADCA0I18S
33	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / ADCA0I19S
34	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
35	JP0_4 / $\overline{\text{DCUTRST}}$
36	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
37	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
38	JP0_1 / INTP1 /TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
39	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
40	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22
41	P1_10 / RLIN24RX / DPIN21

Table 2.1 Pin Assignment 144 pin LQFP (2/4)

Pin No.	Pin Name
42	P1_9 / RLIN34TX / DPIN20
43	P1_8 / RLIN34RX / INTP14
44	$\overline{\text{RESET}}$
45	EVCC
46	XT1
47	IP0_0 / XT2
48	AWOVSS
49	AWOVCL
50	REGVCC
51	X2
52	X1
53	FLMD0
54	JP0_6 / $\overline{\text{EVTO}}$*1
55	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
56	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
57	P0_8/ RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX
58	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
59	EVSS
60	P1_7 / ADCA1TRG1 / RLIN25TX / DPIN19
61	P1_6 / RLIN25RX / DPIN18
62	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17
63	P1_4 / RLIN35RX / INTP15
64	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / ADCA0I0S
65	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / ADCA0I1S
66	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / ADCA0I5S
67	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / ADCA0I6S
68	P8_5 / TAUJ0I3 / TAUJ0O3 / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
69	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S
70	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0I14S
71	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0I15S
72	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0I16S
73	A0VSS
74	A0VREF
75	AP0_15 / ADCA0I15
76	AP0_14 / ADCA0I14
77	AP0_13 / ADCA0I13
78	AP0_12 / ADCA0I12
79	AP0_11 / ADCA0I11
80	AP0_10 / ADCA0I10
81	AP0_9 / ADCA0I9
82	AP0_8 / ADCA0I8

Table 2.1 Pin Assignment 144 pin LQFP (3/4)

Pin No.	Pin Name
83	AP0_7 / ADCA0I7
84	AP0_6 / ADCA0I6
85	AP0_5 / ADCA0I5
86	AP0_4 / ADCA0I4
87	AP0_3 / ADCA0I3
88	AP0_2 / ADCA0I2
89	AP0_1 / ADCA0I1
90	AP0_0 / ADCA0I0
91	EVSS
92	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S
93	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S
94	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
95	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / ADCA0I10S
96	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
97	ISOVCL
98	ISOVSS
99	P20_5 / RLIN23TX / PWGA60O
100	P20_4 / RLIN23RX / PWGA59O
101	EVCC
102	AP1_7 / ADCA1I7
103	AP1_6 / ADCA1I6
104	AP1_5 / ADCA1I5
105	AP1_4 / ADCA1I4
106	AP1_3 / ADCA1I3
107	AP1_2 / ADCA1I2
108	AP1_1 / ADCA1I1
109	AP1_0 / ADCA1I0
110	A1VREF
111	A1VSS
112	BVCC
113	ISOVCL
114	ISOVSS
115	P18_0 / CSIG1RYI / CSIG1RYO / PWGA61O / ADCA1I0S
116	P18_1 / PWGA62O / ADCA1I1S
117	P18_2 / PWGA63O / ADCA1I2S
118	P18_3 / ADCA1I3S
119	BVSS
120	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX /INTP1
121	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX
122	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / FLXA0TXDB / ENCA0EC / PWGA5O / FLMD1
123	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / FLXA0RXDB

Table 2.1 Pin Assignment 144 pin LQFP (4/4)

Pin No.	Pin Name
124	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1
125	P10_11 / PWGA16O / RLIN31RX / INTP11 / FLXA0TXENA / CSIH1CSS0 / TAUB0I1 / TAUB0O1
126	P10_12 / PWGA17O / FLXA0STPWT / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3
127	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / FLXA0TXENB / TAUB0I5 / TAUB0O5
128	P10_14 / ADCA1TRG0 / PWGA19O / FLXA0RXDA / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7
129	P11_1 / $\overline{\text{CSIH2SSI}}$ / FLXA0TXDA / RLIN20RX / CSIH0CSS7 / PWGA26O / TAUB0I13 / TAUB0O13
130	P11_2 / CSIH2SO / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15
131	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O
132	P11_4 / CSIH2SI / CAN3TX / PWGA29O
133	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI
134	P11_6 / RLIN33RX / INTP13 / CAN5TX / ADCA1TRG1 / PWGA31O / CSIH3SO
135	P11_7 / INTP5 / PWGA32O / CSIH3SC
136	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O
137	P12_0 / CAN2TX / PWGA56O
138	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O
139	P12_2 / RLIN34TX / PWGA58O
140	BVCC
141	BVSS
142	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX /INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI
143	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / MODE0
144	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1

Note 1. The $\overline{\text{EVTO}}$ pin is only available in devices with 4-MB code flash memory.

Table 2.2 Pin Assignment 176 pin LQFP (1/4)

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / $\overline{\text{CSIH1SSI}}$ / MEMC0CLK
2	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$
3	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
4	BVCC
5	BVSS
6	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9 / $\overline{\text{MEMC0RD}}$
7	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11 / $\overline{\text{MEMC0WR}}$
8	P11_8 / $\overline{\text{CSIG1SSI}}$ / RLIN35TX / PWGA48O / TAUB1I11 / TAUB1O11 / $\overline{\text{MEMC0CS0}}$
9	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O / TAUB1I13 / TAUB1O13 / $\overline{\text{MEMC0CS1}}$
10	P11_10 / CSIG1SC / PWGA50O / TAUB1I15 / TAUB1O15 / $\overline{\text{MEMC0CS2}}$
11	P11_11 / CSIG1SI / RLIN25TX / PWGA51O / TAUB1I0 / TAUB1O0 / $\overline{\text{MEMC0CS3}}$
12	P11_12 / RLIN25RX / PWGA52O / TAUB1I2 / TAUB1O2 / $\overline{\text{MEMC0WAIT}}$
13	P11_13 / RLIN24RX / PWGA53O / TAUB1I4 / TAUB1O4 / $\overline{\text{MEMC0BEN0}}$
14	P11_14 / RLIN24TX / PWGA54O / TAUB1I6 / TAUB1O6 / $\overline{\text{MEMC0BEN1}}$
15	P12_3 / RLIN27RX / PWGA68O
16	P12_4 / RLIN27TX / PWGA69O
17	P12_5 / PWGA70O
18	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / $\overline{\text{RESETOUT}}$
19	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO
20	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO
21	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO
22	EVCC
23	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8
24	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO
25	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
26	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / RLIN26RX / PWGA34O
27	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI / RLIN26TX
28	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
29	P0_14 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
30	P1_0 / RLIN33RX / INTP13
31	P1_1 / RLIN33TX
32	P1_2 / CAN3RX / INTP3
33	P1_3 / CAN3TX / DPIN23
34	P1_12 / CAN4RX / INTP4
35	P1_13 / CAN4TX
36	P2_6 / ADCA0SEL2
37	EVSS
38	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
39	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / ADCA0I17S
40	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / ADCA0I18S
41	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / ADCA0I19S
42	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
43	JP0_4 / $\overline{\text{DCUTRST}}$
44	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
45	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
46	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
47	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
48	P2_1 / RLIN27TX
49	P2_0 / RLIN27RX

Table 2.2 Pin Assignment 176 pin LQFP (2/4)

Pin No.	Pin Name
50	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22
51	P1_10 / RLIN24RX / DPIN21
52	P1_9 / RLIN34TX / DPIN20
53	P1_8 / RLIN34RX / INTP14
54	$\overline{\text{RESET}}$
55	EVCC
56	XT1
57	IP0_0 / XT2
58	AWOVSS
59	AWOVCL
60	REGVCC
61	X2
62	X1
63	FLMD0
64	P2_3 / RLIN28TX
65	P2_2 / RLIN28RX
66	JP0_6 / $\overline{\text{EVTO}}$*1
67	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
68	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
69	P0_8 / RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX
70	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
71	EVSS
72	P1_7 / ADCA1TRG1 / RLIN25TX / DPIN19
73	P1_6 / RLIN25RX / DPIN18
74	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17
75	P1_4 / RLIN35RX / INTP15
76	P2_4 / RLIN29RX / ADCA0SEL0
77	P2_5 / RLIN29TX / ADCA0SEL1
78	P1_14 / RLIN23RX
79	P1_15 / RLIN23TX
80	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / ADCA0I0S
81	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / ADCA0I1S
82	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / ADCA0I5S
83	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / ADCA0I6S
84	P8_5 / TAUJ0I3 / TAUJ0O3 / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
85	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S
86	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0I14S
87	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0I15S
88	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0I16S
89	A0VSS
90	A0VREF
91	AP0_15 / ADCA0I15
92	AP0_14 / ADCA0I14
93	AP0_13 / ADCA0I13
94	AP0_12 / ADCA0I12
95	AP0_11 / ADCA0I11
96	AP0_10 / ADCA0I10
97	AP0_9 / ADCA0I9
98	AP0_8 / ADCA0I8

Table 2.2 Pin Assignment 176 pin LQFP (3/4)

Pin No.	Pin Name
99	AP0_7 / ADCA0I7
100	AP0_6 / ADCA0I6
101	AP0_5 / ADCA0I5
102	AP0_4 / ADCA0I4
103	AP0_3 / ADCA0I3
104	AP0_2 / ADCA0I2
105	AP0_1 / ADCA0I1
106	AP0_0 / ADCA0I0
107	EVSS
108	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S
109	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S
110	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
111	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / ADCA0I10S
112	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
113	ISOVCL
114	ISOVSS
115	P20_3 / CAN4TX / PWGA67O / RLIN29TX
116	P20_2 / CAN4RX / INTP4 / PWGA66O / RLIN29RX
117	P20_1 / RLIN26TX / PWGA65O
118	P20_0 / RLIN26RX / PWGA64O
119	P20_5 / RLIN23TX / PWGA60O
120	P20_4 / RLIN23RX / PWGA59O
121	EVCC
122	AP1_11 / ADCA1I11
123	AP1_10 / ADCA1I10
124	AP1_9 / ADCA1I9
125	AP1_8 / ADCA1I8
126	AP1_7 / ADCA1I7
127	AP1_6 / ADCA1I6
128	AP1_5 / ADCA1I5
129	AP1_4 / ADCA1I4
130	AP1_3 / ADCA1I3
131	AP1_2 / ADCA1I2
132	AP1_1 / ADCA1I1
133	AP1_0 / ADCA1I0
134	AP1_15 / ADCA1I15
135	AP1_14 / ADCA1I14
136	AP1_13 / ADCA1I13
137	AP1_12 / ADCA1I12
138	A1VREF
139	A1VSS
140	BVCC
141	ISOVCL
142	ISOVSS
143	P18_0 / CSIG1RYI / CSIG1RYO / PWGA61O / ADCA1I0S
144	P18_1 / PWGA62O / ADCA1I1S
145	P18_2 / PWGA63O / ADCA1I2S
146	P18_3 / PWGA71O / ADCA1I3S
147	P18_4 / CSIH1CSS4 / ADCA1I4S

Table 2.2 Pin Assignment 176 pin LQFP (4/4)

Pin No.	Pin Name
148	P18_5 / CSIH1CSS5 / ADCA1I5S
149	P18_6 / ADCA1I6S
150	P18_7 / ADCA1I7S
151	BVSS
152	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MEMC0AD0
153	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / MEMC0AD1
154	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / FLXA0TXDB / ENCA0EC / PWGA5O / MEMC0AD2 / FLMD1
155	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / MEMC0AD3 / FLXA0RXDB
156	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / MEMC0AD4
157	P10_11 / PWGA16O / RLIN31RX / INTP11 / FLXA0TXENA / CSIH1CSS0 / TAUB0I1 / TAUB0O1 / MEMC0AD5
158	P10_12 / PWGA17O / FLXA0STPWT / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3 / MEMC0AD6
159	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / FLXA0TXENB / TAUB0I5 / TAUB0O5 / MEMC0AD7
160	P10_14 / ADCA1TRG0 / PWGA19O / FLXA0RXDA / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7 / MEMC0AD8
161	P11_1 / $\overline{\text{CSIH2SSI}}$ / FLXA0TXDA / RLIN20RX / CSIH0CSS7 / PWGA26O / TAUB0I13 / TAUB0O13 / MEMC0AD9
162	P11_2 / CSIH2SO / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15 / MEMC0AD10
163	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / TAUB1I1 / TAUB1O1 / MEMC0AD11
164	P11_4 / CSIH2SI / CAN3TX / PWGA29O / TAUB1I3 / TAUB1O3 / MEMC0AD12
165	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI / TAUB1I5 / TAUB1O5 / MEMC0AD13
166	P11_6 / RLIN33RX / INTP13 / CAN5TX / ADCA1TRG1 / PWGA31O / CSIH3SO / TAUB1I7 / TAUB1O7 / MEMC0AD14
167	P11_7 / INTP5 / PWGA32O / CSIH3SC / TAUB1I9 / TAUB1O9 / MEMC0AD15
168	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O / TAUB1I8 / TAUB1O8 / $\overline{\text{MEMC0ASTB}}$
169	P12_0 / CAN2TX / PWGA56O / TAUB1I10 / TAUB1O10 / MEMC0A16
170	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O / TAUB1I12 / TAUB1O12 / MEMC0A17
171	P12_2 / RLIN34TX / PWGA58O / TAUB1I14 / TAUB1O14 / MEMC0A18
172	BVCC
173	BVSS
174	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI / MEMC0A19
175	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / MODE0
176	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1

Note 1. The $\overline{\text{EVTO}}$ pin is only available in devices with 4-MB code flash memory.

Table 2.3 Pin Assignment 233 pin BGA (1/5)

Pin No.	Pin Name
A1	BVSS
A2	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX / INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI / MEMC0A19
A3	P12_2 / RLIN34TX / PWGA58O / TAUB1I14 / TAUB1O14 / MEMC0A18
A4	P11_5 / CAN5RX / INTP5 / RLIN33TX / PWGA30O / CSIH3SI / TAUB1I5 / TAUB1O5 / MEMC0AD13
A5	P11_1 / $\overline{\text{CSIH2SSI}}$ / FLXA0TXDA / RLIN20RX / CSIH0CSS7 / PWGA26O / TAUB0I13 / TAUB0O13 / MEMC0AD9
A6	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / FLXA0TXENB / TAUB0I5 / TAUB0O5 / MEMC0AD7
A7	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / MEMC0AD4
A8	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX / MEMC0AD1
A9	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX / INTP1 / MEMC0AD0
A10	P19_2 / ADCA1I18S
A11	P18_15 / ADCA1I15S
A12	P18_13 / ADCA1I13S
A13	P18_6 / ADCA1I6S
A14	P18_5 / CSIH1CSS5 / ADCA1I5S
A15	P18_10 / ADCA1I10S
A16	P18_8 / ADCA1I8S
A17	A1VSS
B1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN / $\overline{\text{CSIH1SSI}}$ / MEMC0CLK
B2	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / MODE0
B3	P13_1
B4	P12_0 / CAN2TX / PWGA56O / TAUB1I10 / TAUB1O10 / MEMC0A16
B5	P11_4 / CSIH2SI / CAN3TX / PWGA29O / TAUB1I3 / TAUB1O3 / MEMC0AD12
B6	P11_3 / CSIH2SC / CAN3RX / INTP3 / PWGA28O / TAUB1I1 / TAUB1O1 / MEMC0AD11
B7	P10_14 / ADCA1TRG0 / PWGA19O / FLXA0RXDA / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7 / MEMC0AD8
B8	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / MEMC0AD3 / FLXA0RXDB
B9	P19_3 / ADCA1I19S
B10	P19_1 / ADCA1I17S
B11	P18_7 / ADCA1I7S
B12	P18_11 / ADCA1I11S
B13	P18_3 / PWGA71O / ADCA1I3S
B14	P18_2 / PWGA63O / ADCA1I2S
B15	P18_1 / PWGA62O / ADCA1I1S
B16	AP1_12 / ADCA1I12
B17	AP1_14 / ADCA1I14
C1	P10_15 / CSIH3RYI / CSIH3RYO / PWGA24O / RLIN22RX / TAUB0I9 / TAUB0O9 / $\overline{\text{MEMC0RD}}$
C2	P10_5 / TAUD0I11 / TAUD0O11 / RLIN21TX / KR0I3 / ADCA0SEL1 / TAPA0WN / CSIG0RYI / CSIG0RYO
C3	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1
C4	P13_0 / MEMC0A19
C5	P12_1 / RLIN34RX / INTP14 / CSIH2CSS5 / PWGA57O / TAUB1I12 / TAUB1O12 / MEMC0A17
C6	P11_7 / INTP5 / PWGA32O / CSIH3SC / TAUB1I9 / TAUB1O9 / MEMC0AD15
C7	P11_2 / CSIH2SO / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15 / MEMC0AD10
C8	P10_11 / PWGA16O / RLIN31RX / INTP11 / FLXA0TXENA / CSIH1CSS0 / TAUB0I1 / TAUB0O1 / MEMC0AD5
C9	P18_14 / ADCA1I14S
C10	P19_0 / ADCA1I16S
C11	P18_4 / CSIH1CSS4 / ADCA1I4S
C12	P18_12 / ADCA1I12S
C13	P18_9 / ADCA1I9S
C14	P18_0 / CSIG1RYI / CSIG1RYO / PWGA61O / ADCA1I0S

Table 2.3 Pin Assignment 233 pin BGA (2/5)

Pin No.	Pin Name
C15	AP1_13 / ADCA1I13
C16	AP1_15 / ADCA1I15
C17	AP1_0 / ADCA1I0
D1	P11_9 / CSIG1SO / RLIN35RX / INTP15 / PWGA49O / TAUB1I13 / TAUB1O13 / $\overline{\text{MEMC0CS1}}$
D2	P11_0 / CSIH2RYI / CSIH2RYO / ADCA1TRG2 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11 / $\overline{\text{MEMC0WR}}$
D3	P10_4 / TAUD0I9 / TAUD0O9 / RLIN21RX / KR0I2 / ADCA0SEL0 / ADCA0TRG2 / TAPA0WP / $\overline{\text{CSIG0SSI}}$
D4	BVCC
D5	P11_15 / CAN2RX / INTP2 / CSIH2CSS4 / PWGA55O / TAUB1I8 / TAUB1O8 / $\overline{\text{MEMC0ASTB}}$
D6	P11_6 / RLIN33RX / INTP13 / CAN5TX / ADCA1TRG1 / PWGA31O / CSIH3SO / TAUB1I7 / TAUB1O7 / MEMC0AD14
D7	P10_12 / PWGA17O / FLXA0STPWT / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3 / MEMC0AD6
D8	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / FLXA0TXDB / ENCA0EC / PWGA5O / MEMC0AD2 / FLMD1
D9	BVSS
D10	BVCC
D11	BVCC
D12	ISOVSS
D13	ISOVCL
D14	A1VSS
D15	AP1_1 / ADCA1I1
D16	AP1_2 / ADCA1I2
D17	AP1_3 / ADCA1I3
E1	P11_12 / RLIN25RX / PWGA52O / TAUB1I2 / TAUB1O2 / $\overline{\text{MEMC0WAIT}}$
E2	P11_10 / CSIG1SC / PWGA50O / TAUB1I15 / TAUB1O15 / $\overline{\text{MEMC0CS2}}$
E3	P11_8 / $\overline{\text{CSIG1SSI}}$ / RLIN35TX / PWGA48O / TAUB1I11 / TAUB1O11 / $\overline{\text{MEMC0CS0}}$
E4	BVCC
E14	A1VREF
E15	AP1_5 / ADCA1I5
E16	AP1_6 / ADCA1I6
E17	AP1_8 / ADCA1I8
F1	P13_2
F2	P11_13 / RLIN24RX / PWGA53O / TAUB1I4 / TAUB1O4 / $\overline{\text{MEMC0BEN0}}$
F3	P11_11 / CSIG1SI / RLIN25TX / PWGA51O / TAUB1I0 / TAUB1O0 / $\overline{\text{MEMC0CS3}}$
F4	BVSS
F14	AP1_4 / ADCA1I4
F15	AP1_7 / ADCA1I7
F16	AP1_9 / ADCA1I9
F17	P20_4 / RLIN23RX / PWGA59O
G1	P13_5
G2	P13_3
G3	P11_14 / RLIN24TX / PWGA54O / TAUB1I6 / TAUB1O6 / $\overline{\text{MEMC0BEN1}}$
G4	P13_4
G7	BVSS
G8	BVSS
G9	BVSS
G10	BVSS
G11	BVSS
G14	AP1_10 / ADCA1I10
G15	AP1_11 / ADCA1I11
G16	P20_5 / RLIN23TX / PWGA60O

Table 2.3 Pin Assignment 233 pin BGA (3/5)

Pin No.	Pin Name
G17	P20_0 / RLIN26RX / PWGA64O
H1	P12_4 / RLIN27TX / PWGA69O
H2	P12_3 / RLIN27RX / PWGA68O
H3	P13_7 / PWGA73O
H4	P13_6 / PWGA72O
H7	BVSS
H8	BVSS
H9	BVSS
H10	BVSS
H11	EVSS
H14	EVCC
H15	P20_1 / RLIN26TX / PWGA65O
H16	P20_2 / CAN4RX / INTP4 / PWGA66O / RLIN29RX
H17	P20_3 / CAN4TX / PWGA67O / RLIN29TX
J1	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / $\overline{\text{RESETOUT}}$
J2	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / INTP0 / RLIN20TX / PWGA11O / CSIH0SI / APO
J3	P12_5 / PWGA70O
J4	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / INTP1 / RLIN30TX / PWGA12O / CSIH0SC / DPO
J7	BVSS
J8	BVSS
J9	BVSS
J10	EVSS
J11	EVSS
J14	ISOVCL
J15	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / TAUJ1I1 / TAUJ1O1 / ADCA0I10S
J16	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
J17	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
K1	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / INTP10 / CAN1TX / DPIN1 / PWGA13O / CSIH0SO
K2	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO
K3	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 / DPIN8
K4	EVCC
K7	EVSS
K8	EVSS
K9	EVSS
K10	EVSS
K11	EVSS
K14	ISOVSS
K15	AP0_0 / ADCA0I0
K16	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S
K17	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S
L1	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / RLIN26RX / PWGA34O
L2	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI / RLIN26TX
L3	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC / PWGA35O
L4	P0_14 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX
L7	EVSS
L8	EVSS
L9	EVSS
L10	EVSS
L11	EVSS

Table 2.3 Pin Assignment 233 pin BGA (4/5)

Pin No.	Pin Name
L14	EVSS
L15	AP0_4 / ADCA0I4
L16	AP0_2 / ADCA0I2
L17	AP0_1 / ADCA0I1
M1	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX / INTP5
M2	P1_0 / RLIN33RX / INTP13
M3	P2_9 / PWGA77O
M4	P2_7
M14	A0VREF
M15	AP0_8 / ADCA0I8
M16	AP0_5 / ADCA0I5
M17	AP0_3 / ADCA0I3
N1	P1_2 / CAN3RX / INTP3
N2	P1_1 / RLIN33TX
N3	P1_3 / CAN3TX / DPIN23
N4	P2_11 / PWGA79O
N14	A0VSS
N15	AP0_11 / ADCA0I11
N16	AP0_7 / ADCA0I7
N17	AP0_6 / ADCA0I6
P1	P1_12 / CAN4RX / INTP4
P2	P1_13 / CAN4TX
P3	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / ADCA0I17S
P4	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / ADCA0I19S
P5	JP0_1 / INTP1 / TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
P6	P1_11 / ADCA1TRG2 / RLIN24TX / DPIN22
P7	P2_13
P8	P2_15 / PWGA75O
P9	EVCC
P10	REGVCC
P11	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX / INTP3
P12	P1_15 / RLIN23TX
P13	P8_6 / NMI / CSIH0CSS4 / PWGA38O / RTCA0OUT / ADCA0I8S
P14	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0I15S
P15	AP0_13 / ADCA0I13
P16	AP0_10 / ADCA0I10
P17	AP0_9 / ADCA0I9
R1	P2_6 / ADCA0SEL2
R2	P2_10 / PWGA78O
R3	JP0_4 / $\overline{\text{DCUTRST}}$
R4	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
R5	P2_1 / RLIN27TX
R6	P1_8 / RLIN34RX / INTP14
R7	P1_9 / RLIN34TX / DPIN20
R8	P3_0 / PWGA76O
R9	FLMD0
R10	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX / INTP4
R11	P1_6 / RLIN25RX / DPIN18
R12	P1_4 / RLIN35RX / INTP15

Table 2.3 Pin Assignment 233 pin BGA (5/5)

Pin No.	Pin Name
R13	P1_14 / RLIN23RX
R14	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / ADCA0I6S
R15	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0I14S
R16	AP0_14 / ADCA0I14
R17	AP0_12 / ADCA0I12
T1	P2_8
T2	P2_12
T3	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / ADCA0I18S
T4	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
T5	P2_0 / RLIN27RX
T6	P2_14 / PWGA74O
T7	IP0_0 / XT2
T8	AWOVCL
T9	X1
T10	P2_2 / RLIN28RX
T11	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX
T12	P0_8 / RLIN21TX / DPIN6 / CSIH0CSS6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX
T13	P2_4 / RLIN29RX / ADCA0SEL0
T14	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / ADCA0I1S
T15	P8_5 / TAUJ0I3 / TAUJ0O3 / CSIH0CSS3 / INTP9 / PWGA37O / ADCA0I7S
T16	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0I16S
T17	AP0_15 / ADCA0I15
U1	EVSS
U2	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
U3	JP0_5 / NMI / RTCA0OUT / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
U4	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
U5	P1_10 / RLIN24RX / DPIN21
U6	$\overline{\text{RESET}}$
U7	XT1
U8	AWOVSS
U9	X2
U10	P2_3 / RLIN28TX
U11	JP0_6 / $\overline{\text{EVTO}}$*1
U12	P1_7 / ADCA1TRG1 / RLIN25TX / DPIN19
U13	P1_5 / ADCA1TRG0 / RLIN35TX / DPIN17
U14	P2_5 / RLIN29TX / ADCA0SEL1
U15	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / ADCA0I0S
U16	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / ADCA0I5S
U17	EVSS

Note 1. The $\overline{\text{EVTO}}$ pin is only available in devices with 4-MB code flash memory.

2.2 Pin Description

Table 2.4 Pin Functions (1/5)

Pin Name	No. of Pins			I/O	Pin Function	Unit
	144 pins	176 pins	233 pins			
AnVREF	√	√	√	—	ADCAn voltage supply and reference voltage	ADCAn
	n = 0, 1	n = 0, 1	n = 0, 1			
AnVSS	√	√	√	—	ADCAn ground	
	n = 0, 1	n = 0, 1	n = 0, 1			
ADCAnIm	√	√	√	I	ADCAn input channel m with 12-bit resolution	
	n = 0, m = 0 to 15 n = 1, m = 0 to 7	n = 0, 1, m = 0 to 15	n = 0, 1, m = 0 to 15			
ADCA0ImS	√	√	√	I	ADCA0 input channel m with 10-bit resolution	
	m = 0 to 11, 14 to 19	m = 0 to 11, 14-19	m = 0 to 11, 14-19			
ADCA1ImS	√	√	√	I	ADCA1 input channel m with 10-bit resolution	
	m = 0 to 3	m = 0 to 7	m = 0 to 19			
ADCA0SELy	√	√	√	O	External MPX selection pin y for ADCA0 input	
	y = 0 to 2	y = 0 to 2	y = 0 to 2			
ADCAnTRGy	√	√	√	I	ADCAn external trigger pin y	
	n = 0, 1, y = 0 to 2	n = 0, 1, y = 0 to 2	n = 0, 1, y = 0 to 2			
AP0_m	√	√	√	IO	Analog port 0_m	Port
	m = 0 to 15	m = 0 to 15	m = 0 to 15			
AP1_m	√	√	√	IO	Analog port 1_m	
	m = 0 to 7	m = 0 to 15	m = 0 to 15			
APO	√	√	√	O	Port output signal for analog input	LPS
AWOVCL	√	√	√	—	Voltage regulator for AWO area capacitor connection	Power
AWOVSS	√	√	√	—	Internal logic for AWO area ground	
BVCC	√	√	√	—	Port buffer voltage supply	
BVSS	√	√	√	—	Port buffer ground	
CANmRX	√	√	√	I	CANm receive data input	RS-CANn
	m = 0 to 5	m = 0 to 5	m = 0 to 5			
CANmTX	√	√	√	O	CANm transmit data output	
	m = 0 to 5	m = 0 to 5	m = 0 to 5			
CSCXFOUT	√	√	√	O	Clock output	Clock
CSIGnRYI	√	√	√	I	CSIGn ready (1) / busy (0) input signal	CSIGn
	n = 0, 1	n = 0, 1	n = 0, 1			
CSIGnRYO	√	√	√	O	CSIGn ready (1) / busy (0) output signal	
	n = 0, 1	n = 0, 1	n = 0, 1			
CSIGnSC	√	√	√	IO	CSIGn serial clock signal	
	n = 0, 1	n = 0, 1	n = 0, 1			
CSIGnSI	√	√	√	I	CSIGn serial data input	
	n = 0, 1	n = 0, 1	n = 0, 1			
CSIGnSO	√	√	√	O	CSIGn serial data output	
	n = 0, 1	n = 0, 1	n = 0, 1			
$\overline{\text{CSIGnSSI}}$	√	√	√	I	CSIGn SS function control input signal	
	n = 0, 1	n = 0, 1	n = 0, 1			

Table 2.4 Pin Functions (2/5)

Pin Name	No. of Pins			I/O	Pin Function	Unit
	144 pins	176 pins	233 pins			
CSIHnCSS0	√	√	√	O	CSIHn serial peripheral chip select signal 0	CSIHn
	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS1	√	√	√	O	CSIHn serial peripheral chip select signal 1	
	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS2	√	√	√	O	CSIHn serial peripheral chip select signal 2	
	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS3	√	√	√	O	CSIHn serial peripheral chip select signal 3	
	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS4	√	√	√	O	CSIHn serial peripheral chip select signal 4	
	n = 0 to 2	n = 0 to 2	n = 0 to 2			
CSIHnCSS5	√	√	√	O	CSIHn serial peripheral chip select signal 5	
	n = 0 to 2	n = 0 to 2	n = 0 to 2			
CSIHnCSS6	√	√	√	O	CSIHn serial peripheral chip select signal 6	
	n = 0	n = 0	n = 0			
CSIHnCSS7	√	√	√	O	CSIHn serial peripheral chip select signal 7	
	n = 0	n = 0	n = 0			
CSIHnRYI	√	√	√	I	CSIHn ready (1) / busy (0) input signal	
	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnRYO	√	√	√	O	CSIHn ready (1) / busy (0) output signal	
	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSC	√	√	√	IO	CSIHn serial clock signal	
	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSI	√	√	√	I	CSIHn serial data input	
	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSO	√	√	√	O	CSIHn serial data output	
	n = 0 to 3	n = 0 to 3	n = 0 to 3			
$\overline{\text{CSIHnSSI}}$	√	√	√	I	CSIHn slave select input signal	
	n = 0 to 3	n = 0 to 3	n = 0 to 3			
$\overline{\text{DCURDY}}$	√	√	√	O	Debug ready	OCD
DCUTCK	√	√	√	I	Debug clock	
DCUTDI	√	√	√	I	Debug data input	
DCUTDO	√	√	√	O	Debug data output	
DCUTMS	√	√	√	I	Debug mode select	
$\overline{\text{DCUTRST}}$	√	√	√	I	Debug reset	
DPINm	√	√	√	I	Digital port input m	LPS
	m = 0 to 23	m = 0 to 23	m = 0 to 23			
DPO	√	√	√	O	Output for digital input port	
ENCA0E0	√	√	√	I	ENCAn encoder input 0	ENCA0
ENCA0E1	√	√	√	I	ENCAn encoder input 1	
ENCA0TINm	√	√	√	I	ENCA0 capture trigger input m	
	m = 0, 1	m = 0, 1	m = 0, 1			
ENCA0EC	√	√	√	I	ENCAn encoder clear input	
EVCC	√	√	√	—	Port buffer voltage supply	Power
EVSS	√	√	√	—	Port buffer ground	
$\overline{\text{EVTO}}$	√*1	√*1	√*1	O	Event output	TEU_OUT

Table 2.4 Pin Functions (3/5)

Pin Name	No. of Pins			I/O	Pin Function	Unit
	144 pins	176 pins	233 pins			
FLMD0	√	√	√	I	Operating mode select pin 0	Mode
FLMD1	√	√	√	I	Operating mode select pin 1	
FLXA0RXDA	√	√	√	I	FlexRay 0 data receive input A	FLXA0
FLXA0RXDB	√	√	√	I	FlexRay 0 data receive input B	
FLXA0STPWT	√	√	√	I	FlexRay 0 stop watch trigger	
FLXA0TXDA	√	√	√	O	FlexRay 0 data transmit output A	
FLXA0TXDB	√	√	√	O	FlexRay 0 data transmit output B	
FLXA0TXENA	√	√	√	O	FlexRay 0 data transmit enable A	
FLXA0TXENB	√	√	√	O	FlexRay 0 data transmit enable B	
INTPm	√	√	√	I	External interrupt input m	INTC
	m = 0 to 15	m = 0 to 15	m = 0 to 15			
IP0_0	√	√	√	I	Input port 0_0	Port
ISOVCL	√	√	√	—	Voltage regulator for ISO area capacitor connection	Power
ISOVSS	√	√	√	—	Internal logic for ISO area ground	
JP0_m	√	√	√	IO	JTAG port 0_m	Port
	m = 0 to 6	m = 0 to 6	m = 0 to 6			
KR0Im	√	√	√	I	KR0 key input signal	KR0
	m = 0 to 7	m = 0 to 7	m = 0 to 7			
LPDCLK	√	√	√	I	LPD clock input (4-pin)	LPD
LPDCLKOUT	√	√	√	O	LPD clock output (4-pin)	
LPDI	√	√	√	I	LPD data input (4-pin)	
LPDIO	√	√	√	IO	LPD data input / output (1-pin)	
LPDO	√	√	√	O	LPD data output (4-pin)	
MEMC0Am	—	√	√	O	MEMC0 address m	MEMC0
		m = 16 to 19	m = 16 to 19			
MEMC0ADm	—	√	√	IO	MEMC0 address / data m	
		m = 0 to 15	m = 0 to 15			
$\overline{\text{MEMC0ASTB}}$	—	√	√	O	MEMC0 address strobe	
$\overline{\text{MEMC0BENm}}$	—	√	√	O	MEMC0 byte enable m	
		m = 0, 1	m = 0, 1			
MEMC0CLK	—	√	√	O	MEMC0 clock output	
$\overline{\text{MEMC0CSm}}$	—	√	√	O	MEMC0 chip select m	
		m = 0 to 3	m = 0 to 3			
$\overline{\text{MEMC0RD}}$	—	√	√	O	MEMC0 read strobe	
$\overline{\text{MEMC0WAIT}}$	—	√	√	I	MEMC0 wait input	
$\overline{\text{MEMC0WR}}$	—	√	√	O	MEMC0 write strobe	
MODEm	√	√	√	I	Sub operating mode select (Boundary scan)	Mode
	m = 0, 1	m = 0, 1	m = 0, 1			
NMI	√	√	√	I	External non-maskable interrupt input	INTC
P0_m	√	√	√	IO	Port 0_m	Port
	m = 0 to 14	m = 0 to 14	m = 0 to 14			
P1_m	√	√	√	IO	Port 1_m	
	m = 0 to 11	m = 0 to 15	m = 0 to 15			
P2_m	—	√	√	IO	Port 2_m	
		m = 0 to 6	m = 0 to 15			

Table 2.4 Pin Functions (4/5)

Pin Name	No. of Pins			I/O	Pin Function	Unit
	144 pins	176 pins	233 pins			
P3_m	—	—	√	IO	Port 3_m	Port
			m = 0			
P8_m	√	√	√	IO	Port 8_m	
	m = 0 to 12	m = 0 to 12	m = 0 to 12			
P9_m	√	√	√	IO	Port 9_m	
	m = 0 to 4	m = 0 to 4	m = 0 to 4			
P10_m	√	√	√	IO	Port 10_m	
	m = 0 to 15	m = 0 to 15	m = 0 to 15			
P11_m	√	√	√	IO	Port 11_m	
	m = 0 to 15	m = 0 to 15	m = 0 to 15			
P12_m	√	√	√	IO	Port 12_m	
	m = 0 to 2	m = 0 to 5	m = 0 to 5			
P13_m	—	—	√	IO	Port 13_m	
			m = 0 to 7			
P18_m	√	√	√	IO	Port 18_m	
	m = 0 to 3	m = 0 to 7	m = 0 to 15			
P19_m	—	—	√	IO	Port 19_m	
			m = 0 to 3			
P20_m	√	√	√	IO	Port 20_m	
	m = 4, 5	m = 0 to 5	m = 0 to 5			
PWGAnO	√	√	√	O	PWGAn output signal	PWM-Diag
	n = 0 to 63	n = 0 to 71	n = 0 to 79			
REGVCC	√	√	√	—	Voltage regulators voltage supply	Power
$\overline{\text{RESET}}$	√	√	√	I	External reset input	Reset
$\overline{\text{RESETOUT}}$	√	√	√	O	Reset output	
RIIC0SCL	√	√	√	IO	RIIC0 serial clock	RIIC0
RIIC0SDA	√	√	√	IO	RIIC0 serial data	
RLIN2mRX	√	√	√	I	RLIN2m receive data input	RLIN2
	m = 0 to 5	m = 0 to 9	m = 0 to 9			
RLIN2mTX	√	√	√	O	RLIN2m transmit data output	
	m = 0 to 5	m = 0 to 9	m = 0 to 9			
RLIN3nRX	√	√	√	I	RLIN3n receive data input	RLIN3
	n = 0 to 5	n = 0 to 5	n = 0 to 5			
RLIN3nTX	√	√	√	O	RLIN3n transmit data output	
	n = 0 to 5	n = 0 to 5	n = 0 to 5			
RTCA0OUT	√	√	√	O	RTCA0 1Hz output	RTCA0
SELDPk	√	√	√	O	External multiplexer selection output signal k for digital port	LPS
	k = 0 to 2	k = 0 to 2	k = 0 to 2			
TAPA0ESO	√	√	√	I	Hi-Z control	Motor control
TAPA0UN	√	√	√	O	Motor control output U phase (negative)	
TAPA0UP	√	√	√	O	Motor control output U phase (positive)	
TAPA0VN	√	√	√	O	Motor control output V phase (negative)	
TAPA0VP	√	√	√	O	Motor control output V phase (positive)	
TAPA0WN	√	√	√	O	Motor control output W phase (negative)	
TAPA0WP	√	√	√	O	Motor control output W phase (positive)	

Table 2.4 Pin Functions (5/5)

Pin Name	No. of Pins			I/O	Pin Function	Unit
	144 pins	176 pins	233 pins			
TAUBnIm	√	√	√	I	TAUBn channel input m	TAUBn
	n = 0, m = 0 to 15	n = 0, 1, m = 0 to 15	n = 0, 1, m = 0 to 15			
TAUBnOm	√	√	√	O	TAUBn channel output m	
	n = 0, m = 0 to 15	n = 0, 1, m = 0 to 15	n = 0, 1, m = 0 to 15			
TAUD0Im	√	√	√	I	TAUD0 channel input m	TAUD0
	m = 0 to 15	m = 0 to 15	m = 0 to 15			
TAUD0Om	√	√	√	O	TAUD0 channel output m	
	m = 0 to 15	m = 0 to 15	m = 0 to 15			
TAUJnIm	√	√	√	I	TAUJn channel input m	TAUJn
	n = 0, 1, m = 0 to 3	n = 0, 1, m = 0 to 3	n = 0, 1, m = 0 to 3			
TAUJnOm	√	√	√	O	TAUJn channel output m	
	n = 0, 1, m = 0 to 3	n = 0, 1, m = 0 to 3	n = 0, 1, m = 0 to 3			
X1, X2	√	√	√	—	MainOSC connections	MOSC
XT1, XT2	√	√	√	—	SubOSC connections	SOSC

Note 1. The $\overline{\text{EVTO}}$ pin is only available in devices with 4-MB code flash memory.

CAUTION

- **When pin functions for a peripheral module are allocated to multiple pins, use the pins from the same port group or nearby pins as the pins for a given channel.**
  (e.g.) When RS-CAN channel 0 is used:
  CAN0TX P0_0 P10_1
  CAN0RX P0_1 P10_0
  Use one of the following pin combinations:
  - **P0_0 and P0_1**
  - **P10_0 and P10_1**
- **The combinations of P0_0 and P10_0, and P0_1 and P10_1 are not allowed.**

2.3 Pin Functions During and After Reset

Table 2.5 Pin Functions During and After Reset

Pins	During Reset*4	After Reset*4
JP0_0	High impedance	JP0_0 : High impedance Serial programming mode: FPDR, FPDT (1 wire UART) FPDR (2 wire UART) Nexus I/F : DCUTDI Input LPD (4-pin) : LPDI Input LPD (1-pin) : LPDIO input/output
JP0_1	High impedance	JP0_1 : High impedance Serial programming mode:FPDT Nexus I/F : DCUTDO output LPD (4-pin) : LPDO output LPD (1-pin) : High impedance
JP0_2	High impedance	JP0_2 : High impedance Serial programming mode:FPCK Nexus I/F : DCUTCK Input LPD (4-pin) : LPDCLK Input LPD (1-pin) : High impedance
JP0_3	High impedance	JP0_3 : High impedance Serial programming mode: High impedance Nexus I/F : DCUTMS Input LPD (4-pin) : High impedance LPD (1-pin) : High impedance
JP0_4	Input*3	JP0_4 : High impedance Serial programming mode: High impedance Nexus I/F : $\overline{\text{DCUTRST}}$ Input*1 LPD (4-pin) : High impedance LPD (1-pin) : High impedance
JP0_5	High impedance	JP0_5 : High impedance Serial programming mode: High impedance Nexus I/F : $\overline{\text{DCURDY}}$ output LPD (4-pin) : LPDCLKOUT output LPD (1-pin) : High impedance
JP0_6	High impedance	JP0_6 : High impedance Serial programming mode: High impedance Nexus I/F : $\overline{\text{EVTO}}$ output*5 LPD (4-pin) : High impedance LPD (1-pin) : High impedance
P0_0	Output*2	Output*2
P0_1 to P0_14	High impedance	High impedance
P1 to P3, P8 to P13, P18 to P20 (Except P10_1, P10_2 and P10_8)	High impedance	High impedance
P10_1	High impedance	High impedance (FLMD = 0) High impedance (FLMD0 = 1, FLMD1 = 0) MODE0 Input (FLMD0 = 1, FLMD1 = 1)
P10_2	High impedance	High impedance (FLMD = 0) High impedance (FLMD0 = 1, FLMD1 = 0) MODE1 Input (FLMD0 = 1, FLMD1 = 1)
P10_8	High impedance	High impedance (FLMD = 0) FLMD1 Input (FLMD0 = 1)
FLMD0	Input	Input
$\overline{\text{RESET}}$	Input	Input
AP0, AP1	High impedance	High impedance

Note 1. When Nexus is enabled and no external device is connected, the level of the pin must always be fixed to low.
Note 2. $\overline{\text{RESETOUT}}$ is output. For details, see **Section 2.11, Port (Special I/O) Function Overview**.
Note 3. When the power is turned on or when a reset is deasserted, the JP0_4 pin should be driven low.
Note 4. It is shown that some pin status is high impedance during and after reset. However the output pin has a possibility to become unstable at the transition moment (less than 12ns) of external and internal reset. After that, pin status will be Hi-Z. The phenomenon of unstable output period and influence of the pin level after the unstable situation, it depends on customer board and measurement conditions.
Note 5. The $\overline{\text{EVTO}}$ pin is only available in devices with 4-MB code flash memory.

2.4 Port State in Standby Mode

For the port states in standby mode, see **Section 12.1.4, I/O Buffer Control**.

2.5 Recommended Connection of Unused Pins

If the pins are not used, it is recommended to connect them as shown below.

Table 2.6 Recommended Connection of Unused Pins

Pin	Recommended Connection of Unused Pins
A0VREF, A1VREF	Connected to EVCC or BVCC
A0VSS, A1VSS	Connected to EVSS or BVSS
$\overline{\text{RESET}}$	Connected to EVCC
XT1	Connected to REGVCC or AWOVSS via a resistor*1 (bit 0 of IPIBC0 = 1) Connected to AWOVSS (bit 0 of IPIBC0 = 0)
IP0_0	Connected to REGVCC or AWOVSS via a resistor*1 (bit 0 of IPIBC0 = 1) Open (bit 0 of IPIBC0 = 0)
JP0 (excluding JP0_4) P0 (excluding P0_0) P1 P2 P8 P9 P20	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVCC or EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
JP0_4	Connected to EVSS via a resistor*2
P0_0	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
P10_1, P10_2, P10_8	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to BVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
P10 (excluding P10_1, P10_2, P10_8) P11 P12 P13 P18 P19	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to BVCC or BVSS via a resistor (when the PIBCn_m or PMCn_m bits are 1) Output: Open
AP0	Input: Open (when the PIBCn_m bit is 0) Connected to A0VREF or A0VSS via a resistor (when the PIBCn_m bit is 1) Output: Open
AP1	Input: Open (when the PIBCn_m bit is 0) Connected to A1VREF or A1VSS via a resistor (when the PIBCn_m bit is 1) Output: Open
Nexus/LPD I/F (JP0)	DCUTDI/LPDI/LPDIO (JP0_0): Connected to EVCC via a resistor DCUTDO/LPDO (JP0_1): Open DCUTCK/LPDCLK (JP0_2): Open DCUTMS (JP0_3): Connected to EVCC via a resistor $\overline{\text{DCUTRST}}$ (JP0_4): Connected to EVSS via a resistor*2 $\overline{\text{DCURDY}}$/LPDCLKOUT (JP0_5):Open $\overline{\text{EVTO}}$ (JP0_6): Open*3

Note 1. XT1 = IP0_0 (XT2) = REGVCC or AWOVSS should be set.
XT1 is connected to IP0_0 (XT2) through an internal resistor. Therefore, it is necessary to maintain equal voltage level in order not to make a current path.
Note 2. For details, see the specifications of the development tool.
Note 3. The $\overline{\text{EVTO}}$ pin is only available in devices with 4-MB code flash memory.

2.6 RH850/F1M Port Features

2.6.1 Port Group

The RH850/F1M provides the following port groups, indicated by the numbers in the table below.

Table 2.7 Port Groups in RH850/F1M

No. of Pins	Port Group	RH850/F1M
144 pins	Number	13
	Name	P0, P1, P8 to P12, P18, P20, JP0, AP0, AP1, IP0
176 pins	Number	14
	Name	P0 to P2, P8 to P12, P18, P20, JP0, AP0, AP1, IP0
233 pins	Number	17
	Name	P0 to P3, P8 to P13, P18 to P20, JP0, AP0, AP1, IP0

2.6.2 Port Group Index n

Throughout this section, the port groups are identified by using the index “n” (n = 0 to 3, 8 to 13, 18 to 20). For example, the port mode control register of the Pn pin is PMCn.

2.6.3 Register Base Address

Port and JTAG port base addresses are listed in the following table.

Port and JTAG port register addresses are given as offsets from the base addresses.

Table 2.8 Register Base Addresses

Base Address Name	Base Address
<PORTn_base>	FFC1 0000_H
<JPORT0_base>	FFC2 0000_H

2.6.4 Clock Supply

The clock supply to ports is shown in the following table.

Table 2.9 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
Port	Register access clock	CPUCLK4

2.7 Port Functions

The microcontroller has various pins for input/output functions, known as ports. The ports are organized in port groups.

The RH850/F1M also has several control registers to enable pins to be used as other than general purpose input/output pins.

For a description of the terms pin, port, or port group, see **Section 2.7.2, Terms**.

2.7.1 Functional Overview

- All the port settings can be specified individually.
- The maximum number of bits (pins) in a port is 16.
- The output level of any pin can be set independently without affecting the other pins in the same port.
- Input buffers are enabled through registers settings.
- Pin level is read by dedicated port-pin-read register (PPR)
- All possible port functions are shown in the tables listed below.
  Section 2.9.2, Pin Function Configuration: **Table 2.37**, **Table 2.39**, **Table 2.41**, **Table 2.43**, **Table 2.45**, **Table 2.47**, **Table 2.49**, **Table 2.51**, **Table 2.53**, **Table 2.55**, **Table 2.57**, **Table 2.59**, **Table 2.61**, **Table 2.63**, **Table 2.65**, **Table 2.67** and **Table 2.69**.

CAUTION

Some input or output functions may be assigned to more than one port. Only activate a given function on a single pin. Do not activate a function on multiple pins at the same time.This also applies in cases where multiple peripheral functions are assigned to a single multiplexed function and only one of these functions is used.

[Example]

INTP0 is assigned to the following pins on this device. However, the INTP0 function should not be activated on more than one pin. After activating the function on one pin, do not activate it on another.

- **JP0_0 (1st input alternative function)**
- **P0_1 (2nd, 3rd input alternative function)**
- **P10_0 (2nd input alternative function)**

In the above case, when the 1st input alternative function (INTP0) of JP0_0 is selected, it is prohibited to activate P0_1 to use the CAN signal of the 2nd input alternative function (CAN0RX/INTP0).

2.7.2 Terms

The following terms are used in this section:

Pin

Denotes the physical pin. Every pin is denoted by a unique pin number.

A pin can be used in several modes. Each pin is assigned a name that reflects its function, which is determined by the selected mode.

Port group

Denotes a group of pins. All the pins of a specific port group are controlled by the same port control register.

Port mode and ports

A pin in port mode works as a general purpose input/output pin. It is then called “port”.

The corresponding name is Pn_m. For example, P0_7 denotes port 7 of port group 0. It is referenced as “port P0_7”.

Alternative mode

In alternative mode, a pin can be used for various non-general-purpose input/output functions, such as the input/output pin of on-chip peripherals.

The corresponding pin name depends on the selected function. For example, pin INTP0 denotes the pin for one of the external interrupt inputs.

Note that two different names can refer to the same physical pin, for example P0_0 and INTP0. The different names indicate the function of the pin at that time.

2.7.2.1 JTAG Ports

The JTAG port groups are used for connecting a debugger for on-chip debugging. These are special port groups provided because the microcontroller cannot be used for the user's application while on-chip debugging is being executed. When a debugger is not connected and the microcontroller is operating normally, these port groups can be used in the same way as the other port groups.

JTAG port group registers and bit names are prefixed by a “J”. For example, JP0 denotes JTAG port group 0, and JPM0.JPM0_m denotes the JPM0_m port mode bit of the JPM0 port mode register.

NOTE

In this section, the description about all ports and their registers other than PFCEn, PFCAEn, PIPCn, PDSCn, and PISn applies to the JTAG port unless otherwise specified.

2.7.3 Overview of Pin Functions

Pins can operate in three modes.

- Port mode (PMCn.PMCn_m bit = 0)

  A pin in port mode operates as a general purpose input/output pin. The I/O mode is selected by setting the PMn.PMn_m bit.

- Software I/O control alternative mode (PMCn.PMCn_m bit = 1, PIPCn.PIPCn_m bit = 0)

  In this mode, the pins operate as alternative functions. The I/O mode is selected by setting the PMn.PMn_m bit.

- Direct I/O control alternative mode (PMCn.PMCn_m bit = 1, PIPCn.PIPCn_m bit = 1)

  In this mode, the pins operate as alternative functions. Unlike the software I/O control alternative mode, however, the I/O mode is directly controlled by the alternative function.

An overview of the register settings is given in the tables below.

Table 2.10 Pin Function Configuration (Overview)

Mode	Bit			I/O
	PMCn_m	PMn_m	PIPCn_m	
Port mode	0	0	X	O
		1*1		I
Software I/O control alternative mode	1	0	0	O
		1	0	I
Direct I/O control alternative mode		X	1	Controlled by the alternative function

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

- Software I/O control alternative mode (PIPCn.PIPCn_m = 0)
  - Output (PMn_m = 0): Alternative output mode 1 to Alternative output mode 7
  - Input (PMn_m = 1): Alternative input mode 1 to Alternative input mode 7
- Direct I/O control alternative mode (PIPCn.PIPCn_m = 1)
  - The I/O mode for Alternative output mode 1 to Alternative output mode 7 and Alternative input mode 1 to Alternative input mode 7 is directly selected by the alternative function.

Table 2.11 Alternative Mode Selection Overview (PMCn.PMCn_m Bit = 1)

Mode	Register					I/O
	PIPC*1	PM*1	PFCAE	PFCE	PFC	
Alternative input mode 1 (ALT-IN1)	0	1	0	0	0	I
Alternative output mode 1 (ALT-OUT1)		0				O
Alternative input mode 2 (ALT-IN2)		1			1	I
Alternative output mode 2 (ALT-OUT2)		0				O
Alternative input mode 3 (ALT-IN3)		1		1	0	I
Alternative output mode 3 (ALT-OUT3)		0				O
Alternative input mode 4 (ALT-IN4)		1			1	I
Alternative output mode 4 (ALT-OUT4)		0				O
Alternative input mode 5 (ALT-IN5)		1	1	0	0	I
Alternative output mode 5 (ALT-OUT5)		0				O
Alternative input mode 6 (ALT-IN6)		1			1	I
Alternative output mode 6 (ALT-OUT6)		0				O
Alternative input mode 7 (ALT-IN7)		1		1	0	I
Alternative output mode 7 (ALT-OUT7)		0				O
Other than above	Setting prohibited					

Note 1. If PIPCn.PIPCn_m = 1, the I/O direction is directly controlled by the peripheral (alternative) function and PM is ignored.

If a pin is in alternative mode (PMCn.PMCn_m = 1), one of up to seven alternative functions can be selected for that pin by using the PFCn, PFCEn, and PFCAEn registers.

2.7.4 Pin Data Input/Output

The registers used for data input/output are described below.

The location that is read via the PPRn register differs depending on the pin mode.

2.7.4.1 Output Data

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

2.7.4.2 Input Data

When the PPRn register is read, either the value of the Pn_m pin, the value of the Pn.Pn_m bit, or the value output by the alternative function is returned.

Which value is returned depends on the pin mode and setting of several control bits.

The different PPRn read modes are shown in the table below.

Table 2.12 PPRn_m Read Values

PMC n_m	PM n_m	PIBC n_m	PIPC n_m	PODC n_m	Mode	PPRn_m Read Value
0	1	0	X	X	Port input, input buffer disabled	Pn.Pn_m bit
		1		X	Port input, input buffer enabled	Pn_m pin
	0	X		0	Port push-pull output	Pn.Pn_m bit*1
				1	Port open-drain output	
1	1	X	0	X	Software I/O control alternative input	Pn_m pin
	0			0	Software I/O control alternative push-pull output	Internal output signal from the alternative function*1
				1	Software I/O control alternative open-drain output	
	X		1	0	Direct I/O control alternative input or push-pull output	I/O port in alternative mode: • Input: Pn_m pin • Output: Internal output signal from the alternative function*1
				1	Direct I/O control alternative input or open-drain output	

Note 1. When PBDCn_m = 1, the level of the Pn_m pin is returned by the PPRn_m bit.

The control registers in the above table have the following effects:

- PMCn.PMCn_m bit
  This bit selects port mode (PMCn_m = 0) or alternative mode (PMCn_m = 1).
- PMn.PMn_m bit
  This bit selects input (PMn_m = 1) or output (PMn_m = 0) when the port mode (PMCn_m = 0) and software I/O control alternative mode (PMCn_m = 1, PIPCn_m = 0) have been selected.
- PIBCn.PIBCn_m bit
  This bit disables (PIBCn_m = 0) or enables (PIBCn_m = 1) the input buffer in input port mode (PMCn_m = 0 and PMn_m = 1). If the input buffer is disabled, PPRn_m reads the Pn.Pn_m bit; otherwise the Pn_m pin level is returned.
- PIPCn.PIPCn_m bit
  This bit selects software I/O control alternative mode or direct I/O control alternative mode.

- PODCn.PODCn_m bit
  This bit selects push-pull output (PODCn_m = 0) or open-drain output (PODCn_m = 1).
- PBDCn.PBDCn_m bit
  In output mode, when this bit is set to 1, the pin enters the bidirectional mode. In bidirectional mode, the level of the signal on a Pn_m pin can be read from by PPRn.PPRn_m.

CAUTION

When using Pn_m as an alternative output function (PMCn.PMCn_m = 1, PMn.PMn_m = 0), the level of the Pn_m pin can be read at the PPRn.PPRn_m bit by enabling bidirectional mode (PBDCn.PBDCn_m = 1). Note, however, that in this case, the level of the Pn_m pin will be input to the alternative input function that the Pn_m pin is being used as.

2.7.4.3 Writing to the Pn Register

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

Pn data can be overwritten in two ways:

- By writing data directly to the Pn register.
  In this case, new data can be written directly to the Pn register.
- By performing an indirect bitwise operation (a “set”, “reset”, or “not” operation) on the Pn register.
  An indirect bitwise operation (“set”, “reset”, or “not”) can be performed on the Pn register by using the following two registers:
  - Port set reset register PSRn
    If PSRn.PSRn (m + 16) bit = 1, the value of the Pn.Pn_m bit is determined by the value of the PSRn.PSRn_m bit.
    In other words, the Pn_m bit can be set or reset without writing directly to the Pn register.
  - Port NOT register PNOTn
    By setting PNOTn.PNOTn_m to 1, the Pn.Pn_m bit can be inverted without writing directly to the Pn register.

An indirect bitwise operation on the Pn register (“set”, “reset”, or “not”) has no effect on the bits that do not need to be updated, allowing you to overwrite only the bit or bits that need to be overwritten.

2.8 Schematic View of Port Control

The following figure is a schematic view of the port control functions.

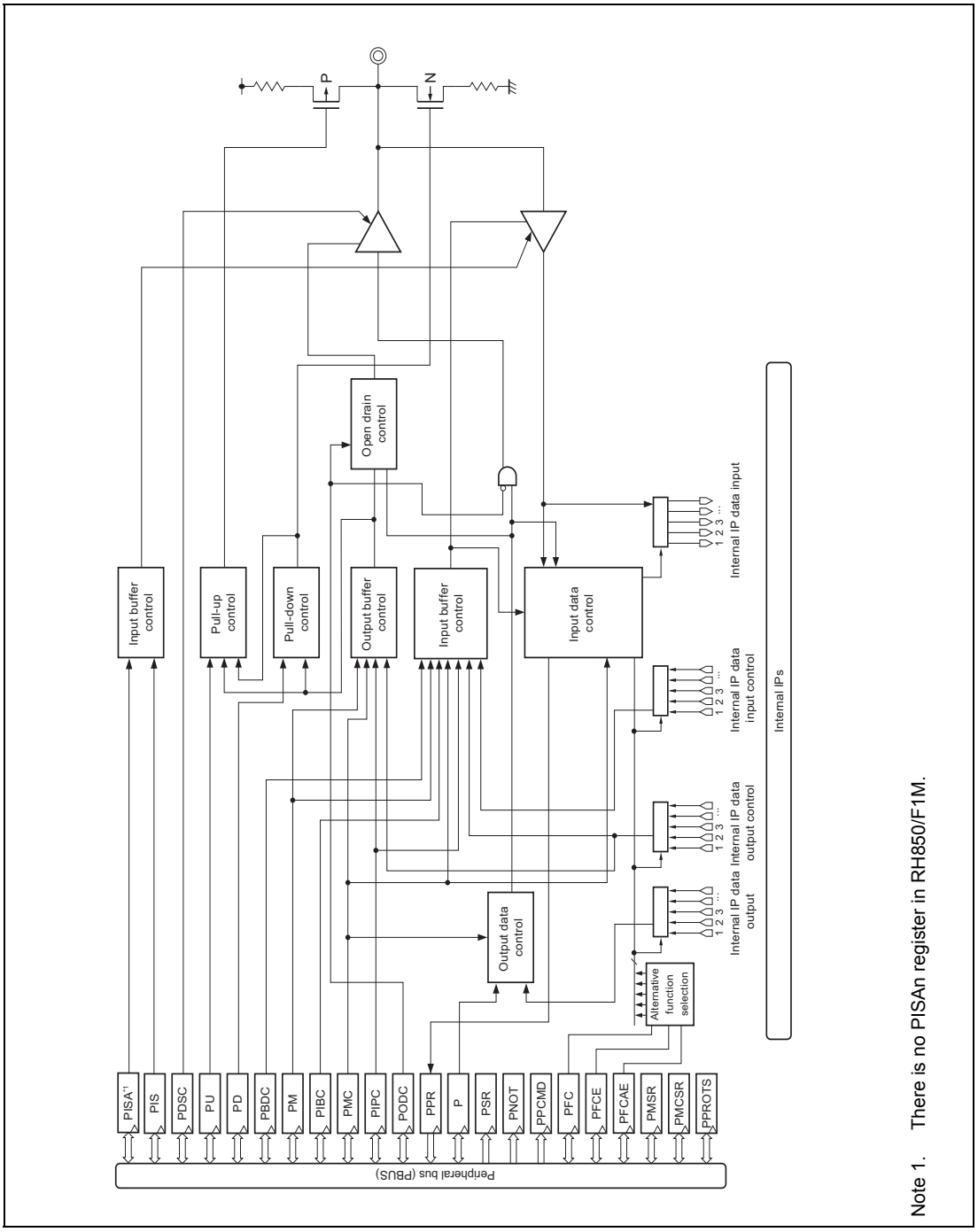


Figure 2.4 Schematic View of Port Control

CAUTION

Use documented alternative functions only.

2.9 Port Group Configuration Registers

This section starts with an overview of all configuration registers and then describes all registers in detail. The configuration registers are grouped as follows:

- **Section 2.9.2, Pin Function Configuration**
- **Section 2.9.3, Pin Data Input/Output**
- **Section 2.9.4, Configuration of Electrical Characteristics**

2.9.1 Overview

The following registers are used for setting the individual pins of the port groups.

For details on <PORTn_base> and <JPORT0_base>, see **Section 2.6.3, Register Base Address**.

Table 2.13 Port Group Configuration Registers (1/2)

Register Name	Symbol	Address
Pin function configuration		
Port mode control register	PMCn	<PORTn_base> + 0400_H + n × 4_H
	JPMC0	<JPORT0_base> + 0040_H
Port mode control set/reset register	PMCSRn	<PORTn_base> + 0900_H + n × 4_H
	JPMCSR0	<JPORT0_base> + 0090_H
Port IP control register	PIPCn	<PORTn_base> + 4200_H + n × 4_H
Port mode register	PMn	<PORTn_base> + 0300_H + n × 4_H
	APMn	<PORTn_base> + $03C8_H$ + n × 4_H
	JPM0	<JPORT0_base> + 0030_H
Port mode set/reset register	PMSRn	<PORTn_base> + 0800_H + n × 4_H
	APMSRn	<PORTn_base> + $08C8_H$ + n × 4_H
	JPMSR0	<JPORT0_base> + 0080_H
Port input buffer control register	PIBCn	<PORTn_base> + 4000_H + n × 4_H
	APIBCn	<PORTn_base> + $40C8_H$ + n × 4_H
	JPIBC0	<JPORT0_base> + 0400_H
	IPIBC0	<PORTn_base> + $40F0_H$
Port function control register	PFCn	<PORTn_base> + 0500_H + n × 4_H
	JPFC0	<JPORT0_base> + 0050_H
Port function control expansion register	PFCEn	<PORTn_base> + 0600_H + n × 4_H
Port function control additional expansion register	PFCAEn	<PORTn_base> + $0A00_H$ + n × 4_H
Pin data input/output		
Port bidirection control register	PBDCn	<PORTn_base> + 4100_H + n × 4_H
	APBDCn	<PORTn_base> + $41C8_H$ + n × 4_H
	JPBDC0	<JPORT0_base> + 0410_H
Port pin read register	PPRn	<PORTn_base> + 0200_H + n × 4_H
	APPRn	<PORTn_base> + $02C8_H$ + n × 4_H
	JPPR0	<JPORT0_base> + 0020_H
	IPPR0	<PORTn_base> + $02F0_H$
Port register	Pn	<PORTn_base> + 0000_H + n × 4_H
	APn	<PORTn_base> + $00C8_H$ + n × 4_H
	JP0	<JPORT0_base> + 0000_H

Table 2.13 Port Group Configuration Registers (2/2)

Register Name	Symbol	Address
Port NOT register	PNOTn	<PORTn_base> + 0700_H + n × 4_H
	APNOTn	<PORTn_base> + $07C8_H$ + n × 4_H
	JPNOT0	<JPORT0_base> + 0070_H
Port set/reset register	PSRn	<PORTn_base> + 0100_H + n × 4_H
	APSRn	<PORTn_base> + $01C8_H$ + n × 4_H
	JPSR0	<JPORT0_base> + 0010_H
Configuration of electrical characteristics		
Pull-up option register	PUn	<PORTn_base> + 4300_H + n × 4_H
	JPU0	<JPORT0_base> + 0430_H
Pull-down option register	PDn	<PORTn_base> + 4400_H + n × 4_H
	JPD0	<JPORT0_base> + 0440_H
Port drive strength control register	PDSCn	<PORTn_base> + 4600_H + n × 4_H
Port open drain control register	PODCn	<PORTn_base> + 4500_H + n × 4_H
	JPODC0	<JPORT0_base> + 0450_H
Port input buffer selection register	PISn	<PORTn_base> + 4700_H + n × 4_H
Port input buffer selection advanced register	JPISA0	<JPORT0_base> + $04A0_H$
Port protection		
Port protection command register	PPCMDn	<PORTn_base> + $4C00_H$ + n × 4_H
	JPPCMD0	<JPORT0_base> + $04C0_H$
Port protection status register	PPROTSn	<PORTn_base> + $4B00_H$ + n × 4_H
	JPPROTS0	<JPORT0_base> + $04B0_H$

Index n

In **Table 2.13, Port Group Configuration Registers**, the index “n” in register symbols denotes the actual indices of the individual port groups. For example, PMCn generically indicates a port mode control register for port group n (Pn). The values for n differ according to the number of pins on the device in the way shown in **Table 2.14**.

Table 2.14 Number of Pins on the Device, Name of Port Groups, and Values for “n” in Register Symbols

Number of Pins on the Device	Port Groups	Values for “n”
144 pins	P0, P1, P8, P9, P10, P11, P12, P18, P20	0, 1, 8, 9, 10, 11, 12, 18, 20
	AP0, AP1	0, 1
176 pins	P0, P1, P2, P8, P9, P10, P11, P12, P18, P20	0, 1, 2, 8, 9, 10, 11, 12, 18, 20
	AP0, AP1	0, 1
233 pins	P0, P1, P2, P3, P8, P9, P10, P11, P12, P13, P18, P19, P20	0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20
	AP0, AP1	0, 1

JTAG port registers

JTAG port registers are not explicitly described in the following register descriptions.

All descriptions (except for those of the PFCEn register, PFCAEn register, PIPCn register, PDSCn register, and PISn register) apply to JTAG port registers. Note, however, that the JTAG port register base address differs from that of regular ports.

Value after reset

The values after reset depend on the ports. For the values after reset, see the register descriptions in the following pages.

2.9.2 Pin Function Configuration

2.9.2.1 PMCn / JPMC0 — Port Mode Control Register

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

Access: PMCn: This register can be read or written in 16-bit units.
JPMC0: This register can be read or written in 8-bit units.

Address: PMCn: <PORTn_base> + 0400_H + n × 4_H (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 20)
JPMC0: <JPORT0_base> + 0040_H*1

Value after reset: 0000_H

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

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, and **Table 2.64, Control Registers (P20)**.

Table 2.15 PMCn Register Contents

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

CAUTIONS

1. I/O is not controlled by only setting alternative mode (PMCn.PMCn_m bit = 1). If the alternative function requires direct I/O control, also set the PIPCn.PIPCn_m bit to 1.
2. If a port is to be used as an input pin in alternative mode, the signals from some pins will pass through a noise filter. These pins may require the setting of the FCLA0CTLm_<name>, DNFA<name>CTL and the DNFA<name>EN register. For details, see **Section 2.12, Noise Filter & Edge/Level Detector**, and **Section 2.13, Description of Port Noise Filter & Edge/Level Detection**.

NOTE

The control bits of the JTAG port mode control register (JPMC0) are JPMC0_[7:0].

2.9.2.2 PMCSRn / JPMCSR0 — Port Mode Control Set/Reset Register

This register provides an alternative method to write data to the PMCn register.

The upper 16 bits of PMCSRn act as a mask which specifies whether or not the value of PMCn.PMCn_m is set by the corresponding bit in the lower 16 bits of PMCSRn.

Access: PMCSRn: This register can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of register PMCn.
JPMCSR0: This register can be read or written in 32-bit units. Bits 31 to 8 are always read as 000000_H. Reading bits 7 to 0 returns the value of register JPMC0.

Address: PMCSRn: <PORTn_base> + 0900_H + n × 4_H (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 20)
JPMCSR0: <JPORT0_base> + 0090_H*1

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMCSRn_31	PMCSRn_30	PMCSRn_29	PMCSRn_28	PMCSRn_27	PMCSRn_26	PMCSRn_25	PMCSRn_24	PMCSRn_23	PMCSRn_22	PMCSRn_21	PMCSRn_20	PMCSRn_19	PMCSRn_18	PMCSRn_17	PMCSRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCSRn_15	PMCSRn_14	PMCSRn_13	PMCSRn_12	PMCSRn_11	PMCSRn_10	PMCSRn_9	PMCSRn_8	PMCSRn_7	PMCSRn_6	PMCSRn_5	PMCSRn_4	PMCSRn_3	PMCSRn_2	PMCSRn_1	PMCSRn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, and **Table 2.64, Control Registers (P20)**.

Table 2.16 PMCSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMCSRn_ [31:16]	Enable bits that specify whether the value of the corresponding lower bit PMCSRn_m (PMCSRn_[15:0]) is written to PMCn_m. 0: PMCn_m is not affected by PMCSRn_m. 1: PMCn_m is PMCSRn_m. Example: If PMCSRn.PMCSRn_31 = 1, the value of bit PMCSRn.PMCSRn_15 is written to bit PMCn.PMCn_15.
15 to 0	PMCSRn_ [15:0]	Data bits that specify the value of PMCn_m if PMCSRn_m of the corresponding upper bit (PMCSRn_[31:16]) is 1. 0: PMCn_m is 0. 1: PMCn_m is 1.

NOTE

The control bits of the JTAG port mode control set/reset register (JPMCSR0) are JPMCSR0_[31:0].

2.9.2.3 PIPCn — Port IP Control Register

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

If the Pn_m pin is operated in alternative mode (PMCn.PMCn_m = 1) and the alternative function requires direct control of the I/O direction, then PIPCn.PIPCn_m must be set to 1 as well. This transfers I/O control to the alternative function and overrules the PMn.PMn_m setting.

Regarding the alternative functions for which the PIPC register must be set, see **Section 2.11, Port (Special I/O) Function Overview**

Access: This register can be read or written in 16-bit units.

Address: PIPCn: <PORTn_base> + 4200_H + n × 4_H (n = 0, 10, 11)*1

Value after reset: 0000_H

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

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.40, Control Registers (P0)**, **Table 2.52, Control Registers (P10)**, and **Table 2.54, Control Registers (P11)**.

Table 2.17 PIPCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIPCn_[15:0]	Specifies the I/O mode. 0: I/O mode is selected by PMn.PMn_m (software I/O control). 1: I/O mode is selected by the peripheral function (direct I/O control).

2.9.2.4 PMn / APMn / JPM0 — Port Mode Register

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

Access: PMn, APMn: These registers can be read or written in 16-bit units.
JPM0: This register can be read or written in 8-bit units.

Address: PMn: <PORTn_base> + 0300_H + n × 4_H (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20)
APMn: <PORTn_base> + $03C8_H$ + n × 4_H (n = 0, 1)
JPM0: <JPORT0_base> + 0030_H*1

Value after reset: $FFFF_H$*2

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

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, **Table 2.62, Control Registers (P19)**, **Table 2.64, Control Registers (P20)**, **Table 2.66, Control Registers (AP0)**, and **Table 2.68, Control Registers (AP1)**.

Note 2. The PM0 register is $FFFE_H$.

Note 3. The PM0_0 bit is 0.

Table 2.18 PMn Register Contents

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

NOTES

1. To use a port in input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), the input buffer must be enabled (PIBCn.PIBCn_m = 1).
2. By default, PMn.PMn_m specifies the I/O direction in port mode (PMCn.PMCn_m = 0) and alternative mode (PMCn.PMCn_m=1), since PIPCn.PIPCn_m = 0 after reset (I/O mode is controlled by PMn.PMn_m).
3. The control bits of the analog port register (APMn) are APMn_[15:0].
4. The control bits of the JTAG port mode register (JPM0) are JPM0_[7:0].

2.9.2.5 PMSRn / APMSRn / JPMSR0 — Port Mode Set/Reset Register

This register provides an alternative method to write data to the PMn register.

The upper 16 bits of PMSRn act as a mask which specifies whether or not the value PMn.PMn_m is set by the corresponding bit in the lower 16 bits of PMSRn.

Access: PMSRn, APMSRn, JPMSR0: These registers can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of registers PMn, APMn and JPM0.

Address: PMSRn: <PORTn_base> + 0800_H + n × 4_H (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20)
APMSRn: <PORTn_base> + $08C8_H$+ n × 4_H (n = 0, 1)
JPMSR0: <JPORT0_base> + 0080_H*1

Value after reset: 0000 $FFFF_H$*2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSRn_31	PMSRn_30	PMSRn_29	PMSRn_28	PMSRn_27	PMSRn_26	PMSRn_25	PMSRn_24	PMSRn_23	PMSRn_22	PMSRn_21	PMSRn_20	PMSRn_19	PMSRn_18	PMSRn_17	PMSRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Note 1. The valid bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, **Table 2.62, Control Registers (P19)**, **Table 2.64, Control Registers (P20)**, **Table 2.66, Control Registers (AP0)**, and **Table 2.68, Control Registers (AP1)**.

Note 2. The PMSR0 register is 0000 $FFFE_H$.

Note 3. The PMSR0_0 bit is 0.

Table 2.19 PMSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PMSRn_[31:16]	Enable bits that specify whether the value of the corresponding lower bit PMSRn_m (PMSRn_[15:0]) is written to PMn_m. 0: PMn_m is not affected by PMSRn_m. 1: PMn_m is PMSRn_m. Example: If PMSRn.PMSRn_31 = 1, the value of bit PMSRn.PMSRn_15 is written to bit PMn.PMn_15.
15 to 0	PMSRn_[15:0]	Data bits that specify the value of PMn_m if PMSRn_m of the corresponding upper bit (PMSRn_[31:16]) is 1. 0: PMn_m is 0. 1: PMn_m is 1.

NOTES

1. The control bits of the JTAG port mode set/reset register (JPMSR0) are JPMSR0_[31:0].
2. The control bits of the analog port mode set/reset register (APMSRn) are APMSRn_[31:0].

2.9.2.6 PIBCn / APIBCn / JPIBC0 / IPIBC0 — Port Input Buffer Control Register

In input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), this register enables the port pin's input buffer.

Access: PIBCn, APIBCn, IPIBC0: These registers can be read or written in 16-bit units.
JPIBC0: This register can be read or written in 8-bit units.

Address: PIBCn: <PORTn_base> + 4000_H + n × 4_H (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20)
APIBCn: <PORTn_base> + $40C8_H$ + n × 4_H (n = 0, 1)
JPIBC0: <JPORT0_base> + 0400_H
IPIBC0: <PORTn_base> + $40F0_H$*1

Value after reset: 0000_H

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

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, **Table 2.62, Control Registers (P19)**, **Table 2.64, Control Registers (P20)**, **Table 2.66, Control Registers (AP0)**, **Table 2.68, Control Registers (AP1)**, and **Table 2.70, Control Registers (IP0)**.

Table 2.20 PIBCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIBCn_[15:0]	Enables/disables the input buffer. 0: Input buffer disabled 1: Input buffer enabled

NOTES

1. When the input buffer is disabled, through current does not flow even when the pin level is Hi-Z. Thus the pin does not need to be fixed to a high or low level externally.
2. The control bits of the JTAG port input buffer control register (JPIBC0) are JPIBC0_[7:0].

CAUTION

Settings in this register are overruled in bidirectional mode (PBDCn.PBDCn_m = 1).

2.9.2.7 PFCn / JPFC0 — Port Function Control Register

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

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is selected by the peripheral function.

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

Access: PFCn: This register can be read or written in 16-bit units.
JPFC0: This register can be read or written in 8-bit units.

Address: PFCn: <PORTn_base> + 0500_H + n × 4_H (n = 0, 1, 2, 8, 9, 10, 11, 12, 13, 18, 20)
JPFC0: <JPORT0_base> + 0050_H*1

Value after reset: 0000_H

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

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, and **Table 2.64, Control Registers (P20)**.

Table 2.21 PFCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2.24, Setting Alternative Functions**.

NOTE

The control bits of the JTAG port function control register (JPFC0) are JPFC0_[7:0].

2.9.2.8 PFCEn — Port Function Control Expansion Register

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

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is specified by the peripheral function.

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

Access: This register can be read or written in 16-bit units.

Address: PFCEn: <PORTn_base> + 0600_H + n × 4_H (n = 0, 8, 9, 10, 11, 12, 20)*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCEn_15	PFCEn_14	PFCEn_13	PFCEn_12	PFCEn_11	PFCEn_10	PFCEn_9	PFCEn_8	PFCEn_7	PFCEn_6	PFCEn_5	PFCEn_4	PFCEn_3	PFCEn_2	PFCEn_1	PFCEn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.40, Control Registers (P0)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, and **Table 2.64, Control Registers (P20)**.

Table 2.22 PFCEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCEn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2.24, Setting Alternative Functions**.

2.9.2.9 PFCAEn — Port Function Control Additional Expansion Register

This register selects the alternative peripheral functions together with PFCEn, PFCn registers.

Some alternative functions directly control the I/O of the Pn_m pin. For such alternative functions, PIPCn.PIPCn_m must be set to 1 and the I/O is specified by the peripheral function.

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

Access: This register can be read or written in 16-bit units.

Address: PFCAEn: <PORTn_base> + $0A00_H$ + n × 4_H (n = 0, 10, 11)*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCAEn_15	PFCAEn_14	PFCAEn_13	PFCAEn_12	PFCAEn_11	PFCAEn_10	PFCAEn_9	PFCAEn_8	PFCAEn_7	PFCAEn_6	PFCAEn_5	PFCAEn_4	PFCAEn_3	PFCAEn_2	PFCAEn_1	PFCAEn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.40, Control Registers (P0)**, **Table 2.52, Control Registers (P10)**, and **Table 2.54, Control Registers (P11)**.

Table 2.23 PFCAEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCAEn_[15:0]	Specifies the alternative function of the pin. For details, see **Table 2.24, Setting Alternative Functions**.

Table 2.24 Setting Alternative Functions

PFCAEn_m	PFCEn_m	PFCn_m	PMn_m	Function
0	0	0	1	Alternative input mode 1
			0	Alternative output mode 1
		1	1	Alternative input mode 2
			0	Alternative output mode 2
	1	0	1	Alternative input mode 3
			0	Alternative output mode 3
		1	1	Alternative input mode 4
			0	Alternative output mode 4
1	0	0	1	Alternative input mode 5
			0	Alternative output mode 5
		1	1	Alternative input mode 6
			0	Alternative output mode 6
	1	0	1	Alternative input mode 7
			0	Alternative output mode 7
		1	X	Setting prohibited
			X	Setting prohibited

CAUTION

- After selecting the alternative function by the PFCn_m, PFCEn_m, or PFCAEn_m bit, set the PMCn_m bit to “1”.
- With this product, the I/O of some functions is assigned to two or more pins, but a specific pin function can only be set to one pin at a time. Setting the same pin function to two or more pins at the same time is prohibited.
  For example, if the a/b/c pin is used as b, the b/d/e pin cannot be used as b. In this case, the b/d/e pin must be configured as a pin function other than b.

NOTE

For more details on the assignment of each function, see **Sections 2.10.1** to **2.10.17**.

2.9.3 Pin Data Input/Output

2.9.3.1 PBDCn / APBDCn / JPBDC0 — Port Bidirection Control Register

This register enables the input buffer in output mode and sets the port to bidirectional mode. In bidirectional mode, the level of the signal on a Pn_m pin can be read from PPRn.PPRn_m.

Access: PBDCn, APBDCn: These registers can be read or written in 16-bit units.
JPBDC0: This register can be read or written in 8-bit units.

Address: PBDCn: <PORTn_base> + 4100_H + n × 4_H (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20)
APBDCn: <PORTn_base> + $41C8_H$ + n × 4_H (n = 0, 1)
JPBDC0: <JPORT0_base> + 0410_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PBDCn_15	PBDCn_14	PBDCn_13	PBDCn_12	PBDCn_11	PBDCn_10	PBDCn_9	PBDCn_8	PBDCn_7	PBDCn_6	PBDCn_5	PBDCn_4	PBDCn_3	PBDCn_2	PBDCn_1	PBDCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, **Table 2.62, Control Registers (P19)**, **Table 2.64, Control Registers (P20)**, **Table 2.66, Control Registers (AP0)**, and **Table 2.68, Control Registers (AP1)**.

Table 2.25 PBDCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PBDCn[15:0]	Enables/disables bidirectional mode of the corresponding pin. 0: Bidirectional mode disabled 1: Bidirectional mode enabled

CAUTION

When the Pn_m port is used for the alternative output function (PMCn.PMCn_m = 1, PMn.PMn_m = 0), the level of the Pn_m pin can be read from PPRn.PPRn_m by enabling the bidirectional mode (PBDCn.PBDCn_m = 1).

However, output of that alternative output function is input to the alternative input function of the same pin (the alternative input function set by PFCn.PFCn_m, PFCEn.PFCEn_m, and PFCAEn.PFCAEn_m). If the alternative input function in question is being used by another pin, the alternative input function is not guaranteed.

NOTE

The control bits of the JTAG port bidirection control register (JPBDC0) are JPBDC0_[7:0].

2.9.3.2 PPRn / APPRn / JPPR0 / IPPR0 — Port Pin Read Register

This register reflects the actual level of the Pn_m pin, whether it is the value of the Pn.Pn_m bit or the level of an alternative output function.

Access: PPRn, APPRn, IPPR0: These registers are read-only registers that can be read in 16-bit units.
JPPR0: This register is a read-only register that can be read in 8-bit units.

Address: PPRn: <PORTn_base> + 0200_H + n × 4_H (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20)
APPRn: <PORTn_base> + $02C8_H$ + n × 4_H (n = 0, 1)
JPPR0: <JPORT0_base> + 0020_H
IPPR0: <PORTn_base> + $02F0_H$*1

Value after reset: 0000_H

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

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, **Table 2.62, Control Registers (P19)**, **Table 2.64, Control Registers (P20)**, **Table 2.66, Control Registers (AP0)**, **Table 2.68, Control Registers (AP1)**, and **Table 2.70, Control Registers (IP0)**.

Table 2.26 PPRn Register Contents

Bit Position	Bit Name	Function
15 to 0	PPRn_[15:0]	The Pn_m pin, Pn.Pn_m value or alternative function output.

NOTES

1. For the read values of the PPRn register, see **Section 2.7.4, Pin Data Input/Output**.
2. The control bits of the JTAG port pin read register (JPPR0) are JPPR0_[7:0].

2.9.3.3 Pn / APn / JP0 — Port Register

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

Access: Pn, APn: These registers can be read or written in 16-bit units.
JP0: This register can be read or written in 8-bit units.

Address: Pn: <PORTn_base> + 0000_H + n × 4_H (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20)
APn: <PORTn_base> + $00C8_H$ + n × 4_H (n = 0, 1)
JP0: <JPORT0_base> + 0000_H*1

Value after reset: 0000_H

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

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, **Table 2.62, Control Registers (P19)**, **Table 2.64, Control Registers (P20)**, **Table 2.66, Control Registers (AP0)**, and **Table 2.68, Control Registers (AP1)**.

Table 2.27 Pn Register Contents

Bit Position	Bit Name	Function
15 to 0	Pn_[15:0]	Sets the output level of the Pn_m pin (m = 0 to 15). 0: Outputs low level 1: Outputs high level

NOTE

The control bits of the JTAG port register (JP0) are JP0_[7:0].

2.9.3.4 PNOTn / APNOTn / JPNOT0 — Port NOT Register

This register allows the Pn_m bit of the port register Pn to be inverted without directly writing to Pn.

Access: PNOTn, APNOTn : These registers are write-only registers that can be written in 16-bit units. When read, 0000_H is returned.
JPNOT0: This register is a write-only register that can be written in 8-bit units. When read, 00_H is returned.

Address: PNOTn: <PORTn_base> + 0700_H + n × 4_H (n = 0, 1, 2, 3. 8, 9, 10, 11, 12, 13, 18, 19, 20)
APNOTn: <PORTn_base> + $07C8_H$ + n × 4_H (n = 0, 1)
JPNOT0: <JPORT0_base> + 0070_H*1

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PNOTn_15	PNOTn_14	PNOTn_13	PNOTn_12	PNOTn_11	PNOTn_10	PNOTn_9	PNOTn_8	PNOTn_7	PNOTn_6	PNOTn_5	PNOTn_4	PNOTn_3	PNOTn_2	PNOTn_1	PNOTn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, **Table 2.62, Control Registers (P19)**, **Table 2.64, Control Registers (P20)**, **Table 2.66, Control Registers (AP0)**, and **Table 2.68, Control Registers (AP1)**.

Table 2.28 PNOTn Register Contents

Bit Position	Bit Name	Function
15 to 0	PNOTn_[15:0]	Specifies if Pn.Pn_m is inverted. 0: Pn.Pn_m is not inverted (Pn_m → Pn_m) 1: Pn.Pn_m is inverted ($\overline{Pn\_m}$ → Pn_m)

NOTE

The control bits of the JTAG port NOT register are JPNOT0_[7:0].

2.9.3.5 PSRn / APSRn / JPSR0 — Port Set/Reset Register

This register provides an alternative method to write data to the Pn register.

The upper 16 bits of PSRn act as a mask which specifies whether or not the value Pn.Pn_m is set by the corresponding bit in the lower 16 bits of PSRn.

Access: PSRn, APSRn: These registers can be read or written in 32-bit units. Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of registers Pn and APn.
JPSR0: This register can be read or written in 32-bit units. Bits 31 to 8 are always read as 000000_H. Reading bits 7 to 0 returns the value of register JP0.

Address: PSRn: <PORTn_base> + 0100_H + n × 4_H (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20)
APSRn: <PORTn_base> + $01C8_H$ + n × 4_H (n = 0, 1)
JPSR0: <JPORT0_base> + 0010_H*1

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PSRn_31	PSRn_30	PSRn_29	PSRn_28	PSRn_27	PSRn_26	PSRn_25	PSRn_24	PSRn_23	PSRn_22	PSRn_21	PSRn_20	PSRn_19	PSRn_18	PSRn_17	PSRn_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

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

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, **Table 2.62, Control Registers (P19)**, **Table 2.64, Control Registers (P20)**, **Table 2.66, Control Registers (AP0)**, and **Table 2.68, Control Registers (AP1)**.

Table 2.29 PSRn Register Contents

Bit Position	Bit Name	Function
31 to 16	PSRn_[31:16]	Specifies whether the value of the corresponding lower bit PSRn_m (PSRn_[15:0]) is written to Pn_m. 0: Pn_m is not affected by PSRn_m. 1: Pn_m is PSRn_m Example: If PSRn.PSRn_31 = 1, the value of bit PSRn.PSRn_15 is written to bit Pn.Pn_15.
15 to 0	PSRn_[15:0]	Specifies the Pn_m value if the corresponding upper bit (PSRn_[31:16]) PSRn_m is 1. 0: Pn_m = 0 1: Pn_m = 1

NOTE

The control bits of the JTAG port set/reset register (JPSR0) are JPSR0_[31:0].

2.9.4 Configuration of Electrical Characteristics

2.9.4.1 PUn / JPU0 — Pull-Up Option Register

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

Access: PUn: This register can be read or written in 16-bit units.
JPU0: This register can be read or written in 8-bit units.

Address: PUn: <PORTn_base> + 4300_H + n × 4_H (n = 0, 1, 2, 3, 8, 9, 10, 11,12, 13, 18, 19, 20)
JPU0: <JPORT0_base> + 0430_H*1

Value after reset: 0000_H

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

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, **Table 2.62, Control Registers (P19)**, and **Table 2.64, Control Registers (P20)**.

Table 2.30 PUn Register Contents

Bit Position	Bit Name	Function
15 to 0	PUn_[15:0]	Specifies whether an internal pull-up resistor is connected to the corresponding pin. 0: No internal pull-up resistor connected 1: An internal pull-up resistor connected

NOTES

1. If a pin is configured such that both an internal pull-up resistor (PUn.PUn_m = 1) and pull-down resistor (PDn.PDn_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The pull-up resistor has no effect when the pin is operated in output mode.
3. The control bits of the JTAG pull-up option register (JPU0) are JPU0_[7:0].

2.9.4.2 PDn / JPD0 — Pull-Down Option Register

This register specifies whether to connect an internal pull-down resistor to an input pin.

Access: PDn: This register can be read or written in 16-bit units.
JPD0: This register can be read or written in 8-bit units.

Address: PDn: <PORTn_base> + 4400_H + n × 4_H (n = 0, 8, 9, 10, 11)
JPD0: <JPORT0_base> + 0440_H*1

Value after reset: 0000_H

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

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, and **Table 2.54, Control Registers (P11)**.

Table 2.31 PDn Register Contents

Bit Position	Bit Name	Function
15 to 0	PDn_[15:0]	Specifies whether to connect an internal pull-down resistor to the corresponding pin: 0: No internal pull-down resistor connected 1: An internal pull-down resistor connected

NOTES

1. If a pin is configured such that both an internal pull-up resistor (PUn.PUn_m = 1) and pull-down resistor (PDn.PDn_m = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The internal pull-down resistor has no effect when the pin is operated in output mode.
3. The control bits of the JTAG pull-down option register (JPD0) are JPD0_[7:0].

2.9.4.3 PDSCn — Port Drive Strength Control Register

This register specifies the output driver strength of the port pin. This function selects the fast mode (high drive strength) or slow mode (low drive strength) of the output buffer. The correct write sequence using the PPCMDn register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**. Regarding the alternative functions for which the PDSC register needs to be set, see **Section 2.11.3.3, Output Buffer Control (PDSC)**.

Access: This register can be read or written in 32-bit units.

Address: PDSCn: <PORTn_base> + 4600_H + n × 4_H (n = 0, 10, 11, 12, 13)*1

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDSCn_15	PDSCn_14	PDSCn_13	PDSCn_12	PDSCn_11	PDSCn_10	PDSCn_9	PDSCn_8	PDSCn_7	PDSCn_6	PDSCn_5	PDSCn_4	PDSCn_3	PDSCn_2	PDSCn_1	PDSCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.40, Control Registers (P0)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, and **Table 2.58, Control Registers (P13)**.

Table 2.32 PDSCn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PDSCn_[15:0]	Specifies the port drive strength of the output buffer of the port pin. 0: Lower drive strength (when the frequency output from the pin is 10 MHz or below) 1: High drive strength (when the frequency output from the pin is 40 MHz or less)

NOTE

See **Table 5.1, Write-Protection Target Registers** when setting this register after returning from DeepSTOP mode (RESF.RESF10 = 1).

2.9.4.4 PODCn / JPODC0 — Port Open Drain Control Register

This register selects push-pull or open-drain as output buffer function. The correct write sequence using the PPCMDn and JPPCMD0 registers is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

Access: PODCn, JPODC0: These registers can be read or written in 32-bit units.

Address: PODCn: <PORTn_base> + 4500_H + n × 4_H (n = 0, 1, 2, 3, 8, 9, 10, 11, 12, 13, 18, 19, 20)
JPODC0: <JPORT0_base> + 0450_H*1

Value after reset: 0000 0000_H*2

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PODCn_15	PODCn_14	PODCn_13	PODCn_12	PODCn_11	PODCn_10	PODCn_9	PODCn_8	PODCn_7	PODCn_6	PODCn_5	PODCn_4	PODCn_3	PODCn_2	PODCn_1	PODCn_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.38, Control Registers (JP0)**, **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.46, Control Registers (P3)**, **Table 2.48, Control Registers (P8)**, **Table 2.50, Control Registers (P9)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, **Table 2.58, Control Registers (P13)**, **Table 2.60, Control Registers (P18)**, **Table 2.62, Control Registers (P19)**, and **Table 2.64, Control Registers (P20)**.

Note 2. The PODC0 register is 0000 0001_H.

Note 3. The PODC0_0 bit is 1.

Table 2.33 PODCn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PODCn_[15:0]	Specifies the output buffer function. 0: Push-pull 1: Open-drain

NOTES

1. The control bits of the JTAG port open drain control register (JPODC0) are JPODC0_[15:0].
2. See **Table 5.1, Write-Protection Target Registers** when setting this register after returning from DeepSTOP mode (RESF.RESF10 = 1).

2.9.4.5 PISn — Port Input Buffer Selection Register

This register specifies the input buffer characteristics.

Access: This register can be read or written in 16-bit units.

Address: PISn: <PORTn_base> + 4700_H + n × 4_H (n = 0, 1, 2, 10, 11, 12, 20)*1

Value after reset: $FFFF_H$

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

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview**: **Table 2.40, Control Registers (P0)**, **Table 2.42, Control Registers (P1)**, **Table 2.44, Control Registers (P2)**, **Table 2.52, Control Registers (P10)**, **Table 2.54, Control Registers (P11)**, **Table 2.56, Control Registers (P12)**, and **Table 2.64, Control Registers (P20)**.

Table 2.34 PISn Register Contents

Bit Position	Bit Name	Function
15 to 0	PISn_[15:0]	Specifies the input buffer characteristic: 0: Type 1 (SHMT1) 1: Type 2 (SHMT4)

NOTE

The definition of type 1 and type 2 is given in **Section 2.11.3.2, Input Buffer Control (PISn, JPISA0)**. Also see the data sheet for detail of input buffer characteristics.

2.9.4.6 JPISA0 — Port Input Buffer Selection Advanced Register

This register specifies the input buffer characteristics.

Access: JPISA0: This register can be read or written in 8-bit units.

Address: JPISA0: <JPORT0_base> + $04A0_H$[*1]

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	JPISA0_3	JPISA0_2	—	JPISA0_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R/W

Note 1. The effective bit positions (value for the index m) vary depending on the number of pins for each device. See the following tables in **Section 2.10, Port (General I/O) Function Overview** : **Table 2.38, Control Registers (JP0)**.

Table 2.35 JPISA0 Register Contents

Bit Position	Bit Name	Function
7 to 4, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2, 0	JPISA0_[3, 2, 0]	Specifies the input buffer characteristic: 0: Type 2 (SHMT4) 1: Type 5 (TTL)

NOTES

1. The definition type 2 and type 5 is given in **Section 2.11.3.2, Input Buffer Control (PISn, JPISA0)**.
2. See the data sheet for details of input buffer characteristics.

2.9.5 Port Register Protection

RH850/F1M has Port Protection Command Registers (PPCMDn) and Port Protection Status Registers (PPROTSn) which implement the Port Protection Cluster Function. For details on the registers, see **Section 5, Write-Protected Registers**.

2.9.6 Flowchart Example for Port Settings

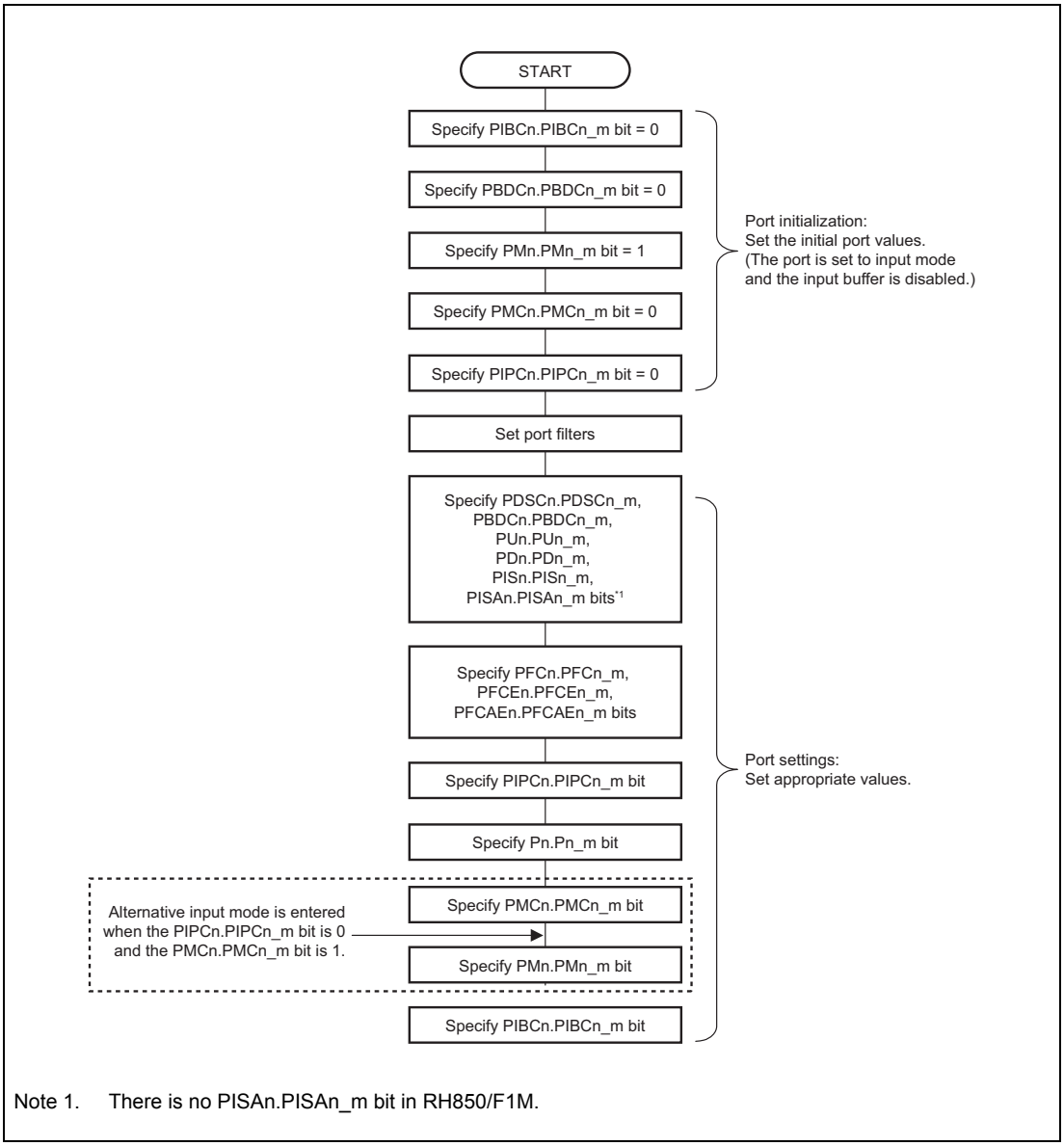
Examples of the port settings are shown in the flowchart below.

CAUTION

If the port is set to the PIPCn.PIPCn_m bit = 0 and alternative output mode, the port might briefly enter alternative input mode. This will occur between when the PMCn.PMCn_m bit is set to 1 and when the PMn.PMn_m bit is set to 0. If an interrupt-related signal is specified as an alternate function of the port, the mode will temporarily become the alternative input mode, so either disable the interrupt in question, or specify that the interrupt is ignored.

2.9.6.1 Batch Setting

An example of specifying batch port group settings is shown in the flowchart below.

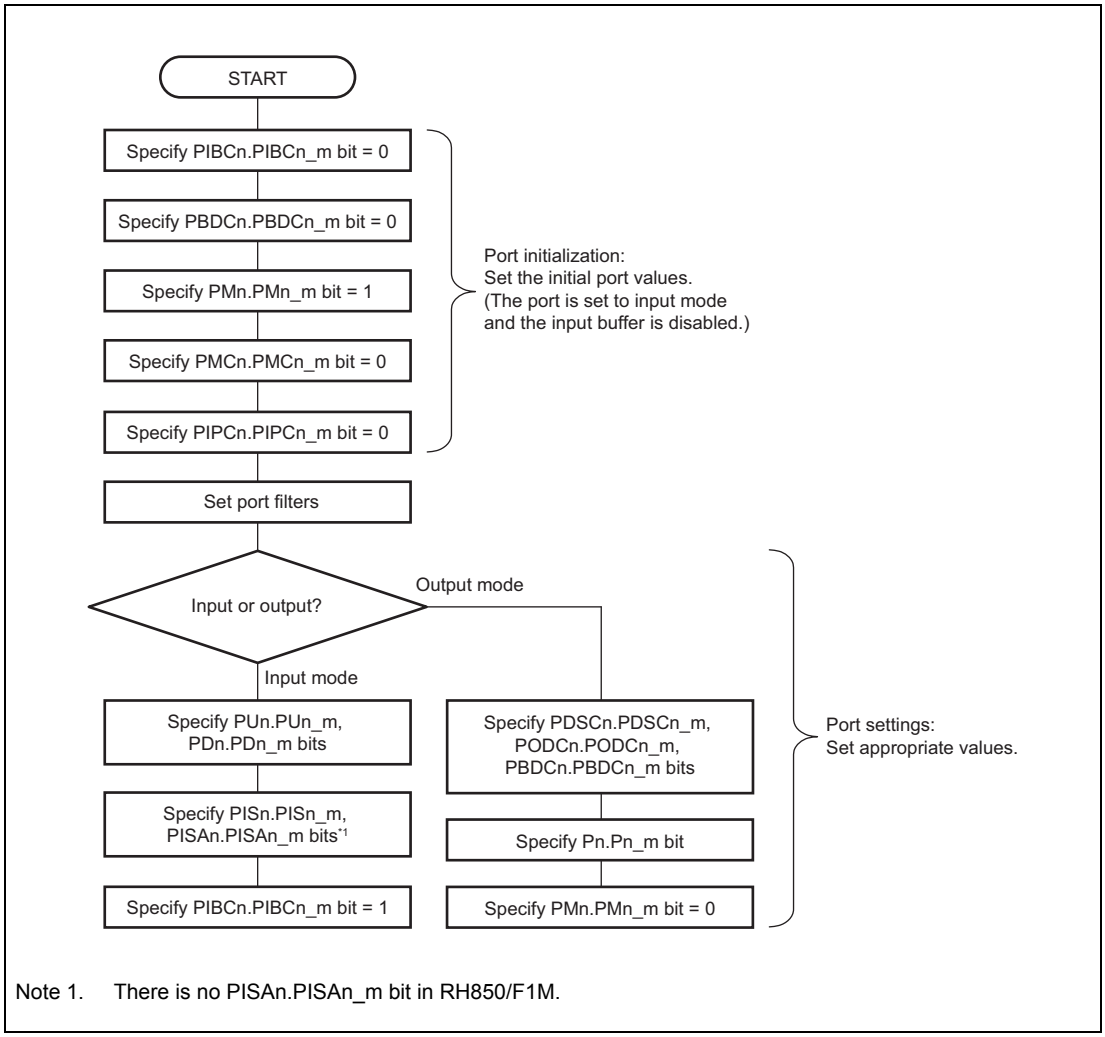


Note 1. There is no PISAn.PISAn_m bit in RH850/F1M.

Figure 2.5 Example of Port Settings (When Specified in Batch)

2.9.6.2 Individual Settings

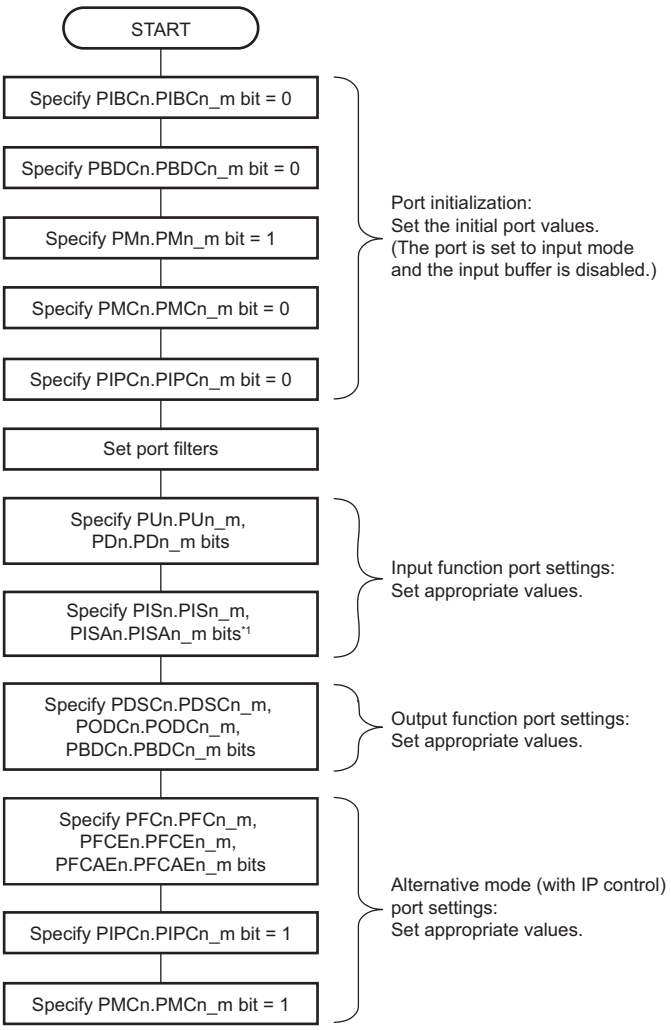
An example of specifying individual port settings is shown in the flowchart below.



Note 1. There is no PISAn.PISAn_m bit in RH850/F1M.

Figure 2.6 Example of Port Settings (in Port Mode)

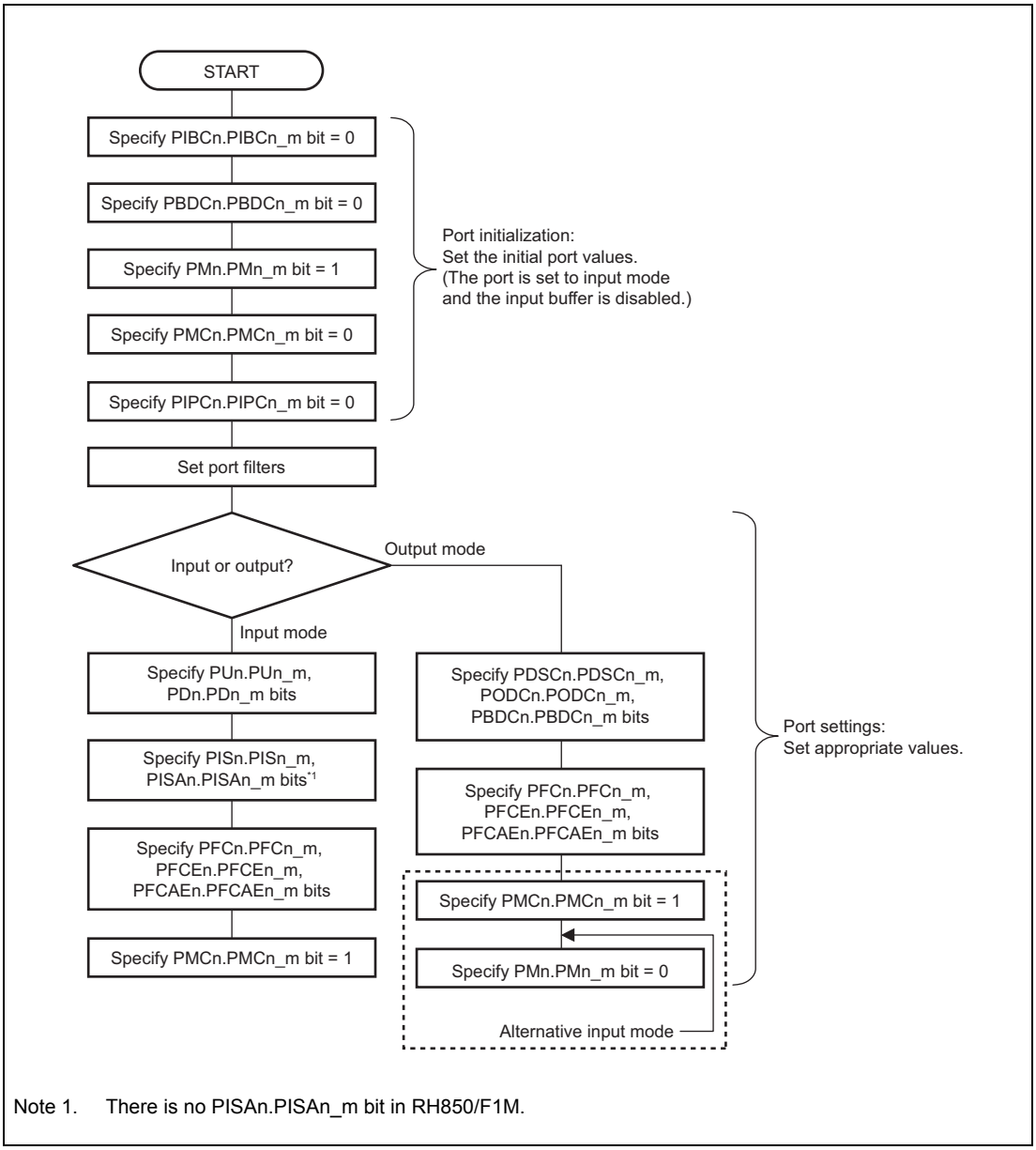
(a) With IP control



Note 1. There is no PISAn.PISAn_m bit in RH850/F1M.

Figure 2.7 Example of Port Settings (in Alternative Mode) (1/2)

(b) Without IP control



Note 1. There is no PISAn.PISAn_m bit in RH850/F1M.

Figure 2.8 Example of Port Settings (in Alternative Mode) (2/2)

2.10 Port (General I/O) Function Overview

This section explains the port (general I/O) functions and all the functions assigned to the ports. See the following pages for details.

In addition, whether the port mode is alternative mode or not can be selected by PMCn register setting. When PMCn.PMCn_m = 1, alternative functions are selected by the PFCn, PFCEn, and PFCAEn registers.

Table 2.36 Port Function

Port	Pin Name	Size	Direction	Power Domain	Special Alternative Function	Device		
						144 pins	176 pins	233 pins
JTAG Port 0	JP0_0 - 6	7 bits	In/Out	AWO	JTAG, LPD	√	√	√
Port 0	P0_0 - 14	15 bits	In/Out	AWO	$\overline{\text{RESETOUT}}$	√	√	√
Port 1	P1_0 - 11	12 bits	In/Out	AWO		√	—	—
	P1_0 - 15	16 bits	In/Out	AWO		—	√	√
Port 2	P2_0 - 6	7 bits	In/Out	AWO		—	√	—
	P2_0 - 15	16 bits				—	—	√
Port 3	P3_0	1 bit	In/Out	AWO		—	—	√
Port 8	P8_0 - 12	13 bits	In/Out	AWO	ADCA0 (10-bit resolution)	√	√	√
Port 9	P9_0 - 4	5 bits	In/Out	ISO	ADCA0 (10-bit resolution)	√	√	√
Port 10	P10_0 - 15	16 bits	In/Out	ISO		√	√	√
Port 11	P11_0 - 15	16 bits	In/Out	ISO		√	√	√
Port 12	P12_0 - 2	3 bits	In/Out	ISO		√	—	—
	P12_0 - 5	6 bits	In/Out	ISO		—	√	√
Port 13	P13_0 - 7	8 bits	In/Out	ISO		—	—	√
Port 18	P18_0 - 3	4 bits	In/Out	ISO	ADCA1 (10-bit resolution)	√	—	—
	P18_0 - 7	8 bits				—	√	—
	P18_0 - 15	16 bits				—	—	√
Port 19	P19_0 - 3	4 bits	In/Out	ISO	ADCA1 (10-bit resolution)	—	—	√
Port 20	P20_4 - 5	2 bits	In/Out	ISO		√	—	—
	P20_0 - 5	6 bits				—	√	√
Analog Port 0	AP0_0 - 15	16 bits	In/Out	AWO	ADCA0 (12/10-bit resolution)	√	√	√
Analog Port 1	AP1_0 - 7	8 bits	In/Out	ISO	ADCA1 (12/10-bit resolution)	√	—	—
	AP1_0 - 15	16 bits	In/Out	ISO	ADCA1 (12/10-bit resolution)	—	√	√
Input Port 0	IP0_0	1 bit	In	AWO	SOSC (XT2 pin)	√	√	√

2.10.1 JTAG Port 0 (JP0)

2.10.1.1 Alternative Function

The following alternative functions are available when the JTAG port 0 is configured as a general-purpose I/O by setting OPJTAG[1:0] on the corresponding option byte to 00_B.

Table 2.37 JTAG Port 0 (JP0)

Port Mode (JPMC0_m = 0)	Alternative Mode (JPMC0_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
JP0_0	INTP0														DCUTDI / LPDI / LPDIO	√	√	√
JP0_1	INTP1		TAUJ0I0	TAUJ0O0											DCUTDO / LPDO	√	√	√
JP0_2	INTP2		TAUJ0I1	TAUJ0O1											DCUTCK / LPDCLK	√	√	√
JP0_3	INTP3	CSCXFOUT	TAUJ0I2	TAUJ0O2											DCUTMS	√	√	√
JP0_4															$\overline{\text{DCUTRST}}$	√	√	√
JP0_5	NMI	RTCA0OUT	TAUJ0I3	TAUJ0O3											$\overline{\text{DCURDY}}$ / LPDCLKOUT	√	√	√
JP0_6																*1	*1	*1
															$\overline{\text{EVTO}}$	*2	*2	*2

Note 1. Available in devices except for ones with 4-MB code flash memory.

Note 2. Only available in devices with 4-MB code flash memory.

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2.10.1.2 Control Registers

Table 2.38 Control Registers (JP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*[1]			144 pins	176 pins	233 pins
JP0	JTAG port register 0	8	6-0	R/W	0000_H	00_H	√	√	√
JPSR0	JTAG port set/reset register 0	32	22-16, 6-0	R/W	0010_H	$0000\ 0000_H$	√	√	√
JPPR0	JTAG port pin read register 0	8	6-0	R	0020_H	00_H	√	√	√
JPM0	JTAG port mode register 0	8	6-0	R/W	0030_H	FF_H	√	√	√
JPMC0	JTAG port mode control register 0	8	5, 3-0	R/W	0040_H	00_H	√	√	√
JPFC0	JTAG port function control register 0	8	5, 3-1	R/W	0050_H	00_H	√	√	√
JPNOT0	JTAG port NOT register 0	8	6-0	W	0070_H	00_H	√	√	√
JPMSR0	JTAG port mode set/reset register 0	32	22-16, 6-0	R/W	0080_H	$0000\ FFFF_H$	√	√	√
JPMCSR0	JTAG port mode control set/reset register 0	32	21, 19-16, 5, 3-0	R/W	0090_H	$0000\ 0000_H$	√	√	√
JPIBC0	JTAG port input buffer control register 0	8	6-0	R/W	0400_H	00_H	√	√	√
JPBDC0	JTAG port bidirection control register 0	8	6-0	R/W	0410_H	00_H	√	√	√
JPU0	Pull-up option register 0	8	6-0	R/W	0430_H	00_H	√	√	√
JPD0	Pull-down option register 0	8	6-0	R/W	0440_H	00_H	√	√	√
JPODC0	JTAG port open drain control register 0	32	6-0	R/W	0450_H	$0000\ 0000_H$	√	√	√
JPISA0	JTAG port input buffer selection advanced register 0	8	3, 2, 0	R/W	$04A0_H$	00_H	√	√	√
JPPROTS0	JTAG port protection status register 0	32	0	R	$04B0_H$	$0000\ 0000_H$	√	√	√
JPPCMD0	JTAG port protection command register 0	32	7-0	W	$04C0_H$	$xxxx\ xx00_H$	√	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2.10.2 Port 0 (P0)

2.10.2.1 Alternative Function

Table 2.39 Port 0 (P0)

Port Mode (PMC0_m = 0)	Alternative Mode (PMC0_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative					
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output		144 pins	176 pins	233 pins
P0_0	TAUD0I2	TAUD0O2	RLIN20RX	CAN0TX		PWGA10O	$\overline{\text{CSIH0SSI}}$	DPO							$\overline{\text{RESETOUT}}$	√	√	√
P0_1	TAUD0I4	TAUD0O4	CAN0RX / INTP0	RLIN20TX	INTP0	PWGA11O	CSIH0SI	APO								√	√	√
P0_2	TAUD0I6	TAUD0O6	CAN1RX / INTP1	RLIN30TX		PWGA12O	CSIH0SC		INTP1	DPO						√	√	√
P0_3	TAUD0I8	TAUD0O8	RLIN30RX/ INTP10	CAN1TX	DPIN1	PWGA13O		CSIH0SO	INTP10							√	√	√
P0_4	RLIN31RX/ INTP11	CAN2TX	INTP11	PWGA10O	CSIH1SI	SELDP0	DPIN8									√	√	√
P0_5	CAN2RX / INTP2	RLIN31TX	DPIN9	SELDP1		CSIH1SO										√	√	√
P0_6	INTP2		DPIN10	SELDP2	CSIH1SC					PWGA35O						√	√	√
P0_7	RLIN21RX		DPIN5	CSCXFOUT	CSIH1RYI	CSIH1RYO	TAUB0I0	TAUB0O0	CAN3RX/ INTP3							√	√	√
P0_8		RLIN21TX	DPIN6	CSIH0CSS6	$\overline{\text{CSIH1SSI}}$		TAUB0I2	TAUB0O2		CAN3TX						√	√	√
P0_9	INTP12	CSIH1CSS0	DPIN7		RLIN22RX		TAUB0I4	TAUB0O4	CAN4RX/ INTP4							√	√	√
P0_10	INTP3	CSIH1CSS1	DPIN11			RLIN22TX	TAUB0I6	TAUB0O6		CAN4TX						√	√	√
P0_11	RIIC0SDA		DPIN12	CSIH1CSS2	TAUB0I8	TAUB0O8		PWGA34O								√	—	—
	RIIC0SDA		DPIN12	CSIH1CSS2	TAUB0I8	TAUB0O8	RLIN26RX	PWGA34O								—	√	√
P0_12	RIIC0SCL		DPIN13	PWGA45O	TAUB0I10	TAUB0O10	CSIG0SI									√	—	—
	RIIC0SCL		DPIN13	PWGA45O	TAUB0I10	TAUB0O10	CSIG0SI	RLIN26TX								—	√	√
P0_13	RLIN32RX/ INTP12		INTP12	PWGA46O	TAUB0I12	TAUB0O12		CSIG0SO	CAN5RX/ INTP5							√	√	√
P0_14		RLIN32TX		PWGA47O	TAUB0I14	TAUB0O14	CSIG0SC			CAN5TX						√	√	√

CAUTIONS

1. The behavior and performance are not guaranteed when alternative functions are not assigned to the register.
2. When the $\overline{\text{RESETOUT}}$ function is selected for the P0_0 pin, the P0_0 pin outputs low-level as $\overline{\text{RESETOUT}}$ signal while a reset is asserted and continues to output a low level after the reset is released. For details, see **Section 2.11.1.1, P0_0: RESETOUT**

2.10.2.2 Control Registers

Table 2.40 Control Registers (P0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*[1]			144 pins	176 pins	233 pins
P0	Port register 0	16	14-0	R/W	0000_H	0000_H	√	√	√
PSR0	Port set/reset register 0	32	30-16, 14-0	R/W	0100_H	$0000\ 0000_H$	√	√	√
PPR0	Port pin read register 0	16	14-0	R	0200_H	0000_H	√	√	√
PM0	Port mode register 0	16	14-0	R/W	0300_H	$FFFE_H$	√	√	√
PMC0	Port mode control register 0	16	14-0	R/W	0400_H	0000_H	√	√	√
PFC0	Port function control register 0	16	14-0	R/W	0500_H	0000_H	√	√	√
PFCE0	Port function control expansion register 0	16	14-0	R/W	0600_H	0000_H	√	√	√
PNOT0	Port NOT register 0	16	14-0	W	0700_H	0000_H	√	√	√
PMSR0	Port mode set/reset register 0	32	30-16, 14-0	R/W	0800_H	$0000\ FFFE_H$	√	√	√
PMCSR0	Port mode control set/reset register 0	32	30-16, 14-0	R/W	0900_H	$0000\ 0000_H$	√	√	√
PFCAE0	Port function control additional expansion register 0	16	14, 13, 10-6, 3, 2	R/W	$0A00_H$	0000_H	√	√	√
PIBC0	Port input buffer control register 0	16	14-0	R/W	4000_H	0000_H	√	√	√
PBDC0	Port bidirection control register 0	16	14-0	R/W	4100_H	0000_H	√	√	√
PIPC0	Port IP control register 0	16	14, 13, 6, 5, 3, 2	R/W	4200_H	0000_H	√	√	√
PU0	Pull-up option register 0	16	14-0	R/W	4300_H	0000_H	√	√	√
PD0	Pull-down option register 0	16	12-0	R/W	4400_H	0000_H	√	√	√
PODC0	Port open drain control register 0	32	14-0	R/W	4500_H	$0000\ 0001_H$	√	√	√
PDSC0	Port drive strength control register 0	32	14, 13, 7-5, 3, 2	R/W	4600_H	$0000\ 0000_H$	√	√	√
PIS0	Port input buffer selection register 0	16	13 - 11, 9, 7-0	R/W	4700_H	$FFFF_H$	√	√	√
PPROTS0	Port protection status register 0	32	0	R	$4B00_H$	$0000\ 0000_H$	√	√	√
PPCMD0	Port protection command register 0	32	7-0	W	$4C00_H$	$xxxx\ xx00_H$	√	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned. When writing to unused bits, write the value after reset.

2.10.3 Port 1 (P1)

2.10.3.1 Alternative Function

Table 2.41 Port 1 (P1)

Port Mode (PMC1_m = 0)	Alternative Mode (PMC1_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P1_0	RLIN33RX/ INTP13		INTP13													√	√	√
P1_1		RLIN33TX														√	√	√
P1_2	CAN3RX/ INTP3		INTP3													√	√	√
P1_3		CAN3TX	DPIN23													√	√	√
P1_4	RLIN35RX/ INTP15		INTP15													√	√	√
P1_5	ADCA1TRG0	RLIN35TX	DPIN17													√	√	√
P1_6	RLIN25RX		DPIN18													√	√	√
P1_7	ADCA1TRG1	RLIN25TX	DPIN19													√	√	√
P1_8	RLIN34RX/ INTP14		INTP14													√	√	√
P1_9		RLIN34TX	DPIN20													√	√	√
P1_10	RLIN24RX		DPIN21													√	√	√
P1_11	ADCA1TRG2	RLIN24TX	DPIN22													√	√	√
P1_12	CAN4RX/ INTP4		INTP4													—	√	√
P1_13		CAN4TX														—	√	√
P1_14	RLIN23RX															—	√	√
P1_15		RLIN23TX														—	√	√

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2.10.3.2 Control Registers

Table 2.42 Control Registers (P1)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
P1	Port register 1	16	15-0	R/W	0004_H	0000_H	—	√	√
			11-0				√	—	—
PSR1	Port set/reset register 1	32	31-16, 15-0	R/W	0104_H	$0000\ 0000_H$	—	√	√
			27-16, 11-0				√	—	—
PPR1	Port pin read register 1	16	15-0	R	0204_H	0000_H	—	√	√
			11-0				√	—	—
PM1	Port mode register 1	16	15-0	R/W	0304_H	$FFFF_H$	—	√	√
			11-0				√	—	—
PMC1	Port mode control register 1	16	15-0	R/W	0404_H	0000_H	—	√	√
			11-0				√	—	—
PFC1	Port function control register 1	16	12-2, 0	R/W	0504_H	0000_H	—	√	√
			11-2, 0				√	—	—
PNOT1	Port NOT register 1	16	15-0	W	0704_H	0000_H	—	√	√
			11-0				√	—	—
PMSR1	Port mode set/reset register 1	32	31-16, 15-0	R/W	0804_H	$0000\ FFFF_H$	—	√	√
			27-16, 11-0				√	—	—
PMCSR1	Port mode control set/reset register 1	32	31-16, 15-0	R/W	0904_H	$0000\ 0000_H$	—	√	√
			27-16, 11-0				√	—	—
PIBC1	Port input buffer control register 1	16	15-0	R/W	4004_H	0000_H	—	√	√
			11-0				√	—	—
PBDC1	Port bidirection control register 1	16	15-0	R/W	4104_H	0000_H	—	√	√
			11-0				√	—	—
PU1	Pull-up option register 1	16	15-0	R/W	4304_H	0000_H	—	√	√
			11-0				√	—	—
PODC1	Port open drain control register 1	32	15-0	R/W	4504_H	$0000\ 0000_H$	—	√	√
			11-0				√	—	—
PIS1	Port input buffer selection register 1	16	14, 12, 10, 8, 6, 4, 2,0	R/W	4704_H	$FFFF_H$	—	√	√
			10, 8, 6, 4, 2,0				√	—	—
PPROTS1	Port protection status register 1	32	0	R	$4B04_H$	$0000\ 0000_H$	√	√	√
PPCMD1	Port protection command register 1	32	7-0	W	$4C04_H$	$xxxx\ xx00_H$	√	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2.10.4 Port 2 (P2)

2.10.4.1 Alternative Function

Table 2.43 Port 2 (P2)

Port Mode (PMC2_m = 0)	Alternative Mode (PMC2_m = 0)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P2_0	RLIN27RX															—	√	√
P2_1		RLIN27TX														—	√	√
P2_2	RLIN28RX															—	√	√
P2_3		RLIN28TX														—	√	√
P2_4	RLIN29RX			ADCA0SEL0												—	√	√
P2_5		RLIN29TX		ADCA0SEL1												—	√	√
P2_6		ADCA0SEL2														—	√	√
P2_7																—	—	√
P2_8																—	—	√
P2_9		PWGA77O														—	—	√
P2_10		PWGA78O														—	—	√
P2_11		PWGA79O														—	—	√
P2_12																—	—	√
P2_13																—	—	√
P2_14		PWGA74O														—	—	√
P2_15		PWGA75O														—	—	√

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2.10.4.2 Control Registers

Table 2.44 Control Registers (P2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
P2	Port register 2	16	6-0	R/W	0008_H	0000_H	—	√	—
			15-0				—	—	√
PSR2	Port set/reset register 2	32	22-16, 6-0	R/W	0108_H	0000 0000_H	—	√	—
			31-16, 15-0				—	—	√
PPR2	Port pin read register 2	16	6-0	R	0208_H	0000_H	—	√	—
			15-0				—	—	√
PM2	Port mode register 2	16	6-0	R/W	0308_H	$FFFF_H$	—	√	—
			15-0				—	—	√
PMC2	Port mode control register 2	16	6-0	R/W	0408_H	0000_H	—	√	—
			15, 14, 11-9, 6-0				—	—	√
PFC2	Port function control register 2	16	5, 4	R/W	0508_H	0000_H	—	√	√
PNOT2	Port NOT register 2	16	6-0	W	0708_H	0000_H	—	√	—
			15-0				—	—	√
PMSR2	Port mode set/reset register 2	32	22-16, 6-0	R/W	0808_H	0000 $FFFF_H$	—	√	—
			31-16, 15-0				—	—	√
PMCSR2	Port mode control set/reset register 2	32	22-16, 6-0	R/W	0908_H	0000 0000_H	—	√	—
			31, 30, 27-25, 22-16, 15, 14, 11-9, 6-0				—	—	√
PIBC2	Port input buffer control register 2	16	6-0	R/W	4008_H	0000_H	—	√	—
			15-0				—	—	√
PBDC2	Port bidirection control register 2	16	6-0	R/W	4108_H	0000_H	—	√	—
			15-0				—	—	√
PU2	Pull-up option register 2	16	6-0	R/W	4308_H	0000_H	—	√	—
			15-0				—	—	√
PODC2	Port open drain control register 2	32	6-0	R/W	4508_H	0000 0000_H	—	√	—
			15-0				—	—	√
PIS2	Port input buffer selection register 2	16	4, 2, 0	R/W	4708_H	$FFFF_H$	—	√	√
PPROTS2	Port protection status register 2	32	0	R	$4B08_H$	0000 0000_H	—	√	√
PPCMD2	Port protection command register 2	32	7-0	W	$4C08_H$	xxxx $xx00_H$	—	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned. When writing to unused bits, write the value after reset.

2.10.5 Port 3 (P3)

2.10.5.1 Alternative Function

Table 2.45 Port 3 (P3)

Port Mode (PMC3_m = 0)	Alternative Mode (PMC3_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P3_0		PWGA76O														—	—	√

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2.10.5.2 Control Registers

Table 2.46 Control Registers (P3)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*[1]			144 pins	176 pins	233 pins
P3	Port register 3	16	0	R/W	$000C_H$	0000_H	—	—	√
PSR3	Port set/reset register 3	32	16, 0	R/W	$010C_H$	$0000\ 0000_H$	—	—	√
PPR3	Port pin read register 3	16	0	R	$020C_H$	0000_H	—	—	√
PM3	Port mode register 3	16	0	R/W	$030C_H$	$FFFF_H$	—	—	√
PMC3	Port mode control register 3	16	0	R/W	$040C_H$	0000_H	—	—	√
PNOT3	Port NOT register 3	16	0	W	$070C_H$	0000_H	—	—	√
PMSR3	Port mode set/reset register 3	32	16, 0	R/W	$080C_H$	$0000\ FFFF_H$	—	—	√
PMCSR3	Port mode control set/reset register 3	32	16, 0	R/W	$090C_H$	$0000\ 0000_H$	—	—	√
PIBC3	Port input buffer control register 3	16	0	R/W	$400C_H$	0000_H	—	—	√
PBDC3	Port bidirection control register 3	16	0	R/W	$410C_H$	0000_H	—	—	√
PU3	Pull-up option register 3	16	0	R/W	$430C_H$	0000_H	—	—	√
PODC3	Port open drain control register 3	32	0	R/W	$450C_H$	$0000\ 0000_H$	—	—	√
PPROTS3	Port protection status register 3	32	0	R	$4B0C_H$	$0000\ 0000_H$	—	—	√
PPCMD3	Port protection command register 3	32	7-0	W	$4C0C_H$	$xxxx\ xx00_H$	—	—	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned. When writing to unused bits, write the value after reset.

2.10.6 Port 8 (P8)

2.10.6.1 Alternative Function

Table 2.47 Port 8 (P8)

Port Mod (PMC8_m = 0)	Alternative Mode (PMC8_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P8_0	TAUJ0I0	TAUJ0O0	DPIN2	PWGA14O	INTP4	CSIH0CSS0									ADCA0I0S	√	√	√
P8_1	TAPA0ESO	TAUJ0O1	DPIN0	PWGA15O	INTP5	CSIH1CSS3									ADCA0I1S	√	√	√
P8_2	TAUJ0I0	TAUJ0O0	DPIN2	CSIH0CSS0	INTP6	PWGA22O									ADCA0I4S	√	√	√
P8_3	TAUJ0I1	TAUJ0O1	DPIN3	CSIH0CSS1	INTP7	PWGA23O									ADCA0I5S	√	√	√
P8_4	TAUJ0I2	TAUJ0O2	DPIN4	CSIH0CSS2	INTP8	PWGA36O									ADCA0I6S	√	√	√
P8_5	TAUJ0I3	TAUJ0O3		CSIH0CSS3	INTP9	PWGA37O									ADCA0I7S	√	√	√
P8_6	NMI	CSIH0CSS4		PWGA38O		RTCA0OUT									ADCA0I8S	√	√	√
P8_7		CSIH3CSS0		PWGA39O											ADCA0I14S	√	√	√
P8_8		CSIH3CSS1		PWGA40O											ADCA0I15S	√	√	√
P8_9		CSIH3CSS2		PWGA41O											ADCA0I16S	√	√	√
P8_10		CSIH3CSS3	DPIN14	PWGA42O											ADCA0I17S	√	√	√
P8_11	TAUJ1I2	TAUJ1O2	DPIN15	PWGA43O		CSIH1CSS4									ADCA0I18S	√	√	√
P8_12	TAUJ1I3	TAUJ1O3	DPIN16	PWGA44O		CSIH1CSS5									ADCA0I19S	√	√	√

CAUTIONS

1. The behavior and performance are not guaranteed when alternative functions are not assigned to the register.
2. Use special functions with their initial settings.

2.10.6.2 Control Registers

Table 2.48 Control Registers (P8)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
P8	Port register 8	16	12-0	R/W	0020_H	0000_H	√	√	√
PSR8	Port set/reset register 8	32	28-16, 12-0	R/W	0120_H	$0000\ 0000_H$	√	√	√
PPR8	Port pin read register 8	16	12-0	R	0220_H	0000_H	√	√	√
PM8	Port mode register 8	16	12-0	R/W	0320_H	$FFFF_H$	√	√	√
PMC8	Port mode control register 8	16	12-0	R/W	0420_H	0000_H	√	√	√
PFC8	Port function control register 8	16	12-0	R/W	0520_H	0000_H	√	√	√
PFCE8	Port function control expansion register 8	16	12, 11, 6-0	R/W	0620_H	0000_H	√	√	√
PNOT8	Port NOT register 8	16	12-0	W	0720_H	0000_H	√	√	√
PMSR8	Port mode set/reset register 8	32	28-16, 12-0	R/W	0820_H	$0000\ FFFF_H$	√	√	√
PMCSR8	Port mode control set/reset register 8	32	28-16, 12-0	R/W	0920_H	$0000\ 0000_H$	√	√	√
PIBC8	Port input buffer control register 8	16	12-0	R/W	4020_H	0000_H	√	√	√
PBDC8	Port bidirection control register 8	16	12-0	R/W	4120_H	0000_H	√	√	√
PU8	Pull-up option register 8	16	12-0	R/W	4320_H	0000_H	√	√	√
PD8	Pull-down option register 8	16	6-0	R/W	4420_H	0000_H	√	√	√
PODC8	Port open drain control register 8	32	12-0	R/W	4520_H	$0000\ 0000_H$	√	√	√
PPROTS8	Port protection status register 8	32	0	R	$4B20_H$	$0000\ 0000_H$	√	√	√
PPCMD8	Port protection command register 8	32	7-0	W	$4C20_H$	$xxxx\ xx00_H$	√	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2.10.7 Port 9 (P9)

2.10.7.1 Alternative Function

Table 2.49 Port 9 (P9)

Port Mod (PMC9_m = 0)	Alternative Mode (PMC9_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P9_0	NMI	PWGA8O	TAUD0I0	TAUD0O0	ADCA0TRG0	CSIH2CSS0	KR0I4								ADCA0I2S	√	√	√
P9_1	INTP11	PWGA9O	TAUD0I2	TAUD0O2	KR0I5	CSIH2CSS1									ADCA0I3S	√	√	√
P9_2	KR0I6	PWGA20O	TAPA0ESO	CSIH2CSS2											ADCA0I9S	√	√	√
P9_3	KR0I7	PWGA21O		CSIH2CSS3	TAUJ1I1	TAUJ1O1									ADCA0I10S	√	√	√
P9_4		CSIH0CSS5		PWGA33O	TAUJ1I0	TAUJ1O0									ADCA0I11S	√	√	√

CAUTIONS

1. The behavior and performance are not guaranteed when alternative functions are not assigned to the register.
2. Use special functions with their initial settings.

2.10.7.2 Control Registers

Table 2.50 Control Registers (P9)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
P9	Port register 9	16	4-0	R/W	0024_H	0000_H	√	√	√
PSR9	Port set/reset register 9	32	20-16, 4-0	R/W	0124_H	$0000\ 0000_H$	√	√	√
PPR9	Port pin read register 9	16	4-0	R	0224_H	0000_H	√	√	√
PM9	Port mode register 9	16	4-0	R/W	0324_H	$FFFF_H$	√	√	√
PMC9	Port mode control register 9	16	4-0	R/W	0424_H	0000_H	√	√	√
PFC9	Port function control register 9	16	4-0	R/W	0524_H	0000_H	√	√	√
PFCE9	Port function control expansion register 9	16	4, 3, 1, 0	R/W	0624_H	0000_H	√	√	√
PNOT9	Port NOT register 9	16	4-0	W	0724_H	0000_H	√	√	√
PMSR9	Port mode set/reset register 9	32	20-16, 4-0	R/W	0824_H	$0000\ FFFF_H$	√	√	√
PMCSR9	Port mode control set/reset register 9	32	20-16, 4-0	R/W	0924_H	$0000\ 0000_H$	√	√	√
PIBC9	Port input buffer control register 9	16	4-0	R/W	4024_H	0000_H	√	√	√
PBDC9	Port bidirection control register 9	16	4-0	R/W	4124_H	0000_H	√	√	√
PU9	Pull-up option register 9	16	4-0	R/W	4324_H	0000_H	√	√	√
PD9	Pull-down option register 9	16	4-0	R/W	4424_H	0000_H	√	√	√
PODC9	Port open drain control register 9	32	4-0	R/W	4524_H	$0000\ 0000_H$	√	√	√
PPROTS9	Port protection status register 9	32	0	R	$4B24_H$	$0000\ 0000_H$	√	√	√
PPCMD9	Port protection command register 9	32	7-0	W	$4C24_H$	$xxxx\ xx00_H$	√	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned. When writing to unused bits, write the value after reset.

2.10.8 Port 10 (P10)

2.10.8.1 Alternative Function

Table 2.51 Port 10 (P10) (1/2)

Port Mod (PMC10_m = 0)	Alternative Mode (PMC10_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P10_0	TAUD0I1	TAUD0O1	CAN0RX/ INTP0	CSCXFOUT		PWGA0O		TAPA0UP	CSIH1SI							√	—	—
	TAUD0I1	TAUD0O1	CAN0RX/ INTP0	CSCXFOUT		PWGA0O		TAPA0UP	CSIH1SI	MEMC0A19						—	√	√
P10_1	TAUD0I3	TAUD0O3		CAN0TX		PWGA1O		TAPA0UN	CSIH1SC						MODE0	√	√	√
P10_2	TAUD0I5	TAUD0O5	RIIC0SDA		KR0I0	PWGA2O	ADCA0TRG0	TAPA0VP		CSIH1SO					MODE1	√	√	√
P10_3	TAUD0I7	TAUD0O7	RIIC0SCL		KR0I1	PWGA3O	ADCA0TRG1	TAPA0VN	$\overline{\text{CSIH1SSI}}$							√	—	—
	TAUD0I7	TAUD0O7	RIIC0SCL		KR0I1	PWGA3O	ADCA0TRG1	TAPA0VN	$\overline{\text{CSIH1SSI}}$	MEMC0CLK						—	√	√
P10_4	TAUD0I9	TAUD0O9	RLIN21RX		KR0I2	ADCA0SEL0	ADCA0TRG2	TAPA0WP	$\overline{\text{CSIG0SSI}}$							√	√	√
P10_5	TAUD0I11	TAUD0O11		RLIN21TX	KR0I3	ADCA0SEL1		TAPA0WN	CSIG0RYI	CSIG0RYO						√	√	√
P10_6	TAUD0I13	TAUD0O13		CSIG0SO	ENCA0TIN0	ADCA0SEL2	CAN1RX/ INTP1									√	—	—
	TAUD0I13	TAUD0O13		CSIG0SO	ENCA0TIN0	ADCA0SEL2	CAN1RX/ INTP1		MEMC0AD0							—	√	√
P10_7	TAUD0I15	TAUD0O15	CSIG0SC		ENCA0TIN1	PWGA4O		CAN1TX								√	—	—
	TAUD0I15	TAUD0O15	CSIG0SC		ENCA0TIN1	PWGA4O		CAN1TX	MEMC0AD1							—	√	√
P10_8	TAUD0I10	TAUD0O10	CSIG0SI	FLXA0TXDB	ENCA0EC	PWGA5O									FLMD1	√	—	—
	TAUD0I10	TAUD0O10	CSIG0SI	FLXA0TXDB	ENCA0EC	PWGA5O	MEMC0AD2								FLMD1	—	√	√
P10_9	TAUD0I12	TAUD0O12	RLIN30RX/ INTP10		ENCA0E0	PWGA6O	CSIH0RYI	CSIH0RYO			FLXA0RXDB					√	—	—
	TAUD0I12	TAUD0O12	RLIN30RX/ INTP10		ENCA0E0	PWGA6O	CSIH0RYI	CSIH0RYO	MEMC0AD3		FLXA0RXDB					—	√	√
P10_10	TAUD0I14	TAUD0O14		RLIN30TX	ENCA0E1	PWGA7O		CSIH0CSS1								√	—	—
	TAUD0I14	TAUD0O14		RLIN30TX	ENCA0E1	PWGA7O		CSIH0CSS1	MEMC0AD4							—	√	√
P10_11		PWGA16O	RLIN31RX/ INTP11	FLXA0TXENA		CSIH1CSS0	TAUB0I1	TAUB0O1								√	—	—
		PWGA16O	RLIN31RX/ INTP11	FLXA0TXENA		CSIH1CSS0	TAUB0I1	TAUB0O1	MEMC0AD5							—	√	√
P10_12		PWGA17O	FLXA0STPWT	RLIN31TX		CSIH1CSS1	TAUB0I3	TAUB0O3								√	—	—
		PWGA17O	FLXA0STPWT	RLIN31TX		CSIH1CSS1	TAUB0I3	TAUB0O3	MEMC0AD6							—	√	√

Table 2.51 Port 10 (P10) (2/2)

Port Mod (PMC10_m = 0)	Alternative Mode (PMC10_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P10_13	$\overline{\text{CSIH0SSI}}$	PWGA18O	RLIN32RX/ INTP12	FLXA0TXENB			TAUB0I5	TAUB0O5								√	—	—
	$\overline{\text{CSIH0SSI}}$	PWGA18O	RLIN32RX/ INTP12	FLXA0TXENB			TAUB0I5	TAUB0O5	MEMC0AD7							—	√	√
P10_14	ADCA1TRG0	PWGA19O	FLXA0RXDA	RLIN32TX	$\overline{\text{CSIH3SSI}}$		TAUB0I7	TAUB0O7								√	—	—
	ADCA1TRG0	PWGA19O	FLXA0RXDA	RLIN32TX	$\overline{\text{CSIH3SSI}}$		TAUB0I7	TAUB0O7	MEMC0AD8							—	√	√
P10_15	CSIH3RYI	CSIH3RYO		PWGA24O	RLIN22RX		TAUB0I9	TAUB0O9								√	—	—
	CSIH3RYI	CSIH3RYO		PWGA24O	RLIN22RX		TAUB0I9	TAUB0O9		$\overline{\text{MEMC0RD}}$						—	√	√

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2.10.8.2 Control Registers

Table 2.52 Control Registers (P10)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
P10	Port register 10	16	15-0	R/W	0028_H	0000_H	√	√	√
PSR10	Port set/reset register 10	32	31-16, 15-0	R/W	0128_H	$0000\ 0000_H$	√	√	√
PPR10	Port pin read register 10	16	15-0	R	0228_H	0000_H	√	√	√
PM10	Port mode register 10	16	15-0	R/W	0328_H	$FFFF_H$	√	√	√
PMC10	Port mode control register 10	16	15-0	R/W	0428_H	0000_H	√	√	√
PFC10	Port function control register 10	16	15-0	R/W	0528_H	0000_H	√	√	√
PFCE10	Port function control expansion register 10	16	15-0	R/W	0628_H	0000_H	√	√	√
PNOT10	Port NOT register 10	16	15-0	W	0728_H	0000_H	√	√	√
PMSR10	Port mode set/reset register 10	32	31-16, 15-0	R/W	0828_H	$0000\ FFFF_H$	√	√	√
PMCSR10	Port mode control set/reset register 10	32	31-16, 15-0	R/W	0928_H	$0000\ 0000_H$	√	√	√
PFCAE10	Port function control additional expansion register 10	16	9, 5-0	R/W	$0A28_H$	0000_H	√	—	—
			15-9, 7-0				—	√	√
PIBC10	Port input buffer control register 10	16	15-0	R/W	4028_H	0000_H	√	√	√
PBDC10	Port bidirection control register 10	16	15-0	R/W	4128_H	0000_H	√	√	√
PIPC10	Port IP control register 10	16	7-0	R/W	4228_H	0000_H	√	—	—
			14-0				—	√	√
PU10	Pull-up option register 10	16	15-0	R/W	4328_H	0000_H	√	√	√
PD10	Pull-down option register 10	16	15-0	R/W	4428_H	0000_H	√	√	√
PODC10	Port open drain control register 10	32	15-0	R/W	4528_H	$0000\ 0000_H$	√	√	√
PDSC10	Port drive strength control register 10	32	13, 11, 8-0	R/W	4628_H	$0000\ 0000_H$	√	—	—
			15-0				—	√	√
PIS10	Port input buffer selection register 10	16	15-11, 9, 6, 4-2, 0	R/W	4728_H	$FFFF_H$	√	√	√
PPROTS10	Port protection status register 10	32	0	R	$4B28_H$	$0000\ 0000_H$	√	√	√
PPCMD10	Port protection command register 10	32	7-0	W	$4C28_H$	$xxxx\ xx00_H$	√	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2.10.9 Port 11 (P11)

2.10.9.1 Alternative Function

Table 2.53 Port 11 (P11) (1/2)

Port Mode (PMC11_m = 0)	Alternative Mode (PMC11_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P11_0	CSIH2RYI	CSIH2RYO	ADCA1TRG2	PWGA25O		RLIN22TX	TAUB0I11	TAUB0O11								√	—	—
	CSIH2RYI	CSIH2RYO	ADCA1TRG2	PWGA25O		RLIN22TX	TAUB0I11	TAUB0O11		$\overline{\text{MEMC0WR}}$						—	√	√
P11_1	$\overline{\text{CSIH2SSI}}$	FLXA0TXDA	RLIN20RX	CSIH0CSS7		PWGA26O	TAUB0I13	TAUB0O13								√	—	—
	$\overline{\text{CSIH2SSI}}$	FLXA0TXDA	RLIN20RX	CSIH0CSS7		PWGA26O	TAUB0I13	TAUB0O13	MEMC0AD9							—	√	√
P11_2		CSIH2SO		RLIN20TX		PWGA27O	TAUB0I15	TAUB0O15								√	—	—
		CSIH2SO		RLIN20TX		PWGA27O	TAUB0I15	TAUB0O15	MEMC0AD10							—	√	√
P11_3	CSIH2SC		CAN3RX/ INTP3	PWGA28O												√	—	—
	CSIH2SC		CAN3RX/ INTP3	PWGA28O	TAUB1I1	TAUB1O1	MEMC0AD11									—	√	√
P11_4	CSIH2SI	CAN3TX		PWGA29O												√	—	—
	CSIH2SI	CAN3TX		PWGA29O	TAUB1I3	TAUB1O3	MEMC0AD12									—	√	√
P11_5	CAN5RX/ INTP5	RLIN33TX		PWGA30O	CSIH3SI											√	—	—
	CAN5RX/ INTP5	RLIN33TX		PWGA30O	CSIH3SI		TAUB1I5	TAUB1O5	MEMC0AD13							—	√	√
P11_6	RLIN33RX/ INTP13	CAN5TX	ADCA1TRG1	PWGA31O		CSIH3SO										√	—	—
	RLIN33RX/ INTP13	CAN5TX	ADCA1TRG1	PWGA31O		CSIH3SO	TAUB1I7	TAUB1O7	MEMC0AD14							—	√	√
P11_7	INTP5			PWGA32O	CSIH3SC											√	—	—
	INTP5			PWGA32O	CSIH3SC		TAUB1I9	TAUB1O9	MEMC0AD15							—	√	√
P11_8	$\overline{\text{CSIG1SSI}}$	RLIN35TX		PWGA48O												√	—	—
	$\overline{\text{CSIG1SSI}}$	RLIN35TX		PWGA48O	TAUB1I11	TAUB1O11		$\overline{\text{MEMC0CS0}}$								—	√	√
P11_9		CSIG1SO	RLIN35RX/ INTP15	PWGA49O												√	—	—
		CSIG1SO	RLIN35RX/ INTP15	PWGA49O	TAUB1I13	TAUB1O13		$\overline{\text{MEMC0CS1}}$								—	√	√
P11_10	CSIG1SC			PWGA50O												√	—	—
	CSIG1SC			PWGA50O	TAUB1I15	TAUB1O15		$\overline{\text{MEMC0CS2}}$								—	√	√
P11_11	CSIG1SI	RLIN25TX		PWGA51O												√	—	—
	CSIG1SI	RLIN25TX		PWGA51O	TAUB1I0	TAUB1O0		$\overline{\text{MEMC0CS3}}$								—	√	√

Table 2.53 Port 11 (P11) (2/2)

Port Mode (PMC11_m = 0)	Alternative Mode (PMC11_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P11_12	RLIN25RX			PWGA52O												√	—	—
	RLIN25RX			PWGA52O	TAUB1I2	TAUB1O2	$\overline{\text{MEMC0WAIT}}$									—	√	√
P11_13	RLIN24RX			PWGA53O												√	—	—
	RLIN24RX			PWGA53O	TAUB1I4	TAUB1O4		$\overline{\text{MEMC0BEN0}}$								—	√	√
P11_14		RLIN24TX		PWGA54O												√	—	—
		RLIN24TX		PWGA54O	TAUB1I6	TAUB1O6		$\overline{\text{MEMC0BEN1}}$								—	√	√
P11_15	CAN2RX/ INTP2	CSIH2CSS4		PWGA55O												√	—	—
	CAN2RX/ INTP2	CSIH2CSS4		PWGA55O	TAUB1I8	TAUB1O8		$\overline{\text{MEMC0ASTB}}$								—	√	√

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2.10.9.2 Control Registers

Table 2.54 Control Registers (P11)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*[1]			144 pins	176 pins	233 pins
P11	Port register 11	16	15-0	R/W	$002C_H$	0000_H	√	√	√
PSR11	Port set/reset register 11	32	31-16, 15-0	R/W	$012C_H$	$0000\ 0000_H$	√	√	√
PPR11	Port pin read register 11	16	15-0	R	$022C_H$	0000_H	√	√	√
PM11	Port mode register 11	16	15-0	R/W	$032C_H$	$FFFF_H$	√	√	√
PMC11	Port mode control register 11	16	15-0	R/W	$042C_H$	0000_H	√	√	√
PFC11	Port function control register 11	16	15-0	R/W	$052C_H$	0000_H	√	√	√
PFCE11	Port function control expansion register 11	16	7-5, 2-0	R/W	$062C_H$	0000_H	√	—	—
			15-0				—	√	√
PNOT11	Port NOT register 11	16	15-0	W	$072C_H$	0000_H	√	√	√
PMSR11	Port mode set/reset register 11	32	31-16, 15-0	R/W	$082C_H$	$0000\ FFFF_H$	√	√	√
PMCSR11	Port mode control set/reset register 11	32	31-16, 15-0	R/W	$092C_H$	$0000\ 0000_H$	√	√	√
PFCAE11	Port function control additional expansion register 11	16	7-5, 2-0	R/W	$0A2C_H$	0000_H	—	√	√
PIBC11	Port input buffer control register 11	16	15-0	R/W	$402C_H$	0000_H	√	√	√
PBDC11	Port bidirection control register 11	16	15-0	R/W	$412C_H$	0000_H	√	√	√
PIPC11	Port IP control register 11	16	10, 9, 7, 6, 3, 2	R/W	$422C_H$	0000_H	√	—	—
			10, 9, 7-1				—	√	√
PU11	Pull-up option register 11	16	15-0	R/W	$432C_H$	0000_H	√	√	√
PD11	Pull-down option register 11	16	4-0	R/W	$442C_H$	0000_H	√	√	√
PODC11	Port open drain control register 11	32	15-0	R/W	$452C_H$	$0000\ 0000_H$	√	√	√
PDSC11	Port drive strength control register 11	32	10, 9, 7, 6, 3-1	R/W	$462C_H$	$0000\ 0000_H$	√	—	—
			15-13, 11-0				—	√	√
PIS11	Port input buffer selection register 11	16	15, 13, 12, 9, 6, 5, 3, 1	R/W	$472C_H$	$FFFF_H$	√	√	√
PPROTS11	Port protection status register 11	32	0	R	$4B2C_H$	$0000\ 0000_H$	√	√	√
PPCMD11	Port protection command register 11	32	7-0	W	$4C2C_H$	$xxxx\ xx00_H$	√	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned. When writing to unused bits, write the value after reset.

2.10.10 Port 12 (P12)

2.10.10.1 Alternative Function

Table 2.55 Port 12 (P12)

Port Mode (PMC12_m = 0)	Alternative Mode (PMC12_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P12_0		CAN2TX		PWGA56O												√	—	—
		CAN2TX		PWGA56O	TAUB1I10	TAUB1O10		MEMC0A16								—	√	√
P12_1	RLIN34RX/ INTP14	CSIH2CSS5		PWGA57O												√	—	—
	RLIN34RX/ INTP14	CSIH2CSS5		PWGA57O	TAUB1I12	TAUB1O12		MEMC0A17								—	√	√
P12_2		RLIN34TX		PWGA58O												√	—	—
		RLIN34TX		PWGA58O	TAUB1I14	TAUB1O14		MEMC0A18								—	√	√
P12_3	RLIN27RX			PWGA68O												—	√	√
P12_4		RLIN27TX		PWGA69O												—	√	√
P12_5		PWGA70O														—	√	√

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2.10.10.2 Control Registers

Table 2.56 Control Registers (P12)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
P12	Port register 12	16	2-0	R/W	0030_H	0000_H	√	—	—
			5-0				—	√	√
PSR12	Port set/reset register 12	32	18-16, 2-0	R/W	0130_H	$0000\ 0000_H$	√	—	—
			21-16, 5-0				—	√	√
PPR12	Port pin read register 12	16	2-0	R	0230_H	0000_H	√	—	—
			5-0				—	√	√
PM12	Port mode register 12	16	2-0	R/W	0330_H	$FFFF_H$	√	—	—
			5-0				—	√	√
PMC12	Port mode control register 12	16	2-0	R/W	0430_H	0000_H	√	—	—
			5-0				—	√	√
PFC12	Port function control register 12	16	2-0	R/W	0530_H	0000_H	√	—	—
			4-0				—	√	√
PFCE12	Port function control expansion register 12	16	5-0	R/W	0630_H	0000_H	—	√	√
PNOT12	Port NOT register 12	16	2-0	W	0730_H	0000_H	√	—	—
			5-0				—	√	√
PMSR12	Port mode set/reset register 12	32	18-16, 2-0	R/W	0830_H	$0000\ FFFF_H$	√	—	—
			21-16, 5-0				—	√	√
PMCSR12	Port mode control set/reset register 12	32	18-16, 2-0	R/W	0930_H	$0000\ 0000_H$	√	—	—
			21-16, 5-0				—	√	√
PIBC12	Port input buffer control register 12	16	2-0	R/W	4030_H	0000_H	√	—	—
			5-0				—	√	√
PBDC12	Port bidirection control register 12	16	2-0	R/W	4130_H	0000_H	√	—	—
			5-0				—	√	√
PU12	Pull-up option register 12	16	2-0	R/W	4330_H	0000_H	√	—	—
			5-0				—	√	√
PODC12	Port open drain control register 12	32	2-0	R/W	4530_H	$0000\ 0000_H$	√	—	—
			5-0				—	√	√
PDSC12	Port drive strength control register 12	32	2-0	R/W	4630_H	$0000\ 0000_H$	—	√	√
PIS12	Port input buffer selection register 12	16	1	R/W	4730_H	$FFFF_H$	√	—	—
			3, 1				—	√	√
PPROTS12	Port protection status register 12	32	0	R	$4B30_H$	$0000\ 0000_H$	√	√	√
PPCMD12	Port protection command register 12	32	7-0	W	$4C30_H$	$xxxx\ xx00_H$	√	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned. When writing to unused bits, write the value after reset.

2.10.11 Port 13 (P13)

2.10.11.1 Alternative Function

Table 2.57 Port 13 (P13)

Port Mode (PMC13_m = 0)	Alternative Mode (PMC13_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P13_0		MEMC0A19														—	—	√
P13_1																—	—	√
P13_2																—	—	√
P13_3																—	—	√
P13_4																—	—	√
P13_5																—	—	√
P13_6				PWGA72O												—	—	√
P13_7				PWGA73O												—	—	√

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2.10.11.2 Control Registers

Table 2.58 Control Registers (P13)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
P13	Port register 13	16	7-0	R/W	0034_H	0000_H	—	—	√
PSR13	Port set/reset register 13	32	23-16, 7-0	R/W	0134_H	$0000\ 0000_H$	—	—	√
PPR13	Port pin read register 13	16	7-0	R	0234_H	0000_H	—	—	√
PM13	Port mode register 13	16	7-0	R/W	0334_H	$FFFF_H$	—	—	√
PMC13	Port mode control register 13	16	7, 6, 0	R/W	0434_H	0000_H	—	—	√
PFC13	Port function control register 13	16	7, 6	R/W	0534_H	0000_H	—	—	√
PNOT13	Port NOT register 13	16	7-0	W	0734_H	0000_H	—	—	√
PMSR13	Port mode set/reset register 13	32	23-16, 7-0	R/W	0834_H	$0000\ FFFF_H$	—	—	√
PMCSR13	Port mode control set/reset register 13	32	23, 22, 16, 7, 6, 0	R/W	0934_H	$0000\ 0000_H$	—	—	√
PIBC13	Port input buffer control register 13	16	7-0	R/W	4034_H	0000_H	—	—	√
PBDC13	Port bidirection control register 13	16	7-0	R/W	4134_H	0000_H	—	—	√
PU13	Pull-up option register 13	16	7-0	R/W	4334_H	0000_H	—	—	√
PODC13	Port open drain control register 13	32	7-0	R/W	4534_H	$0000\ 0000_H$	—	—	√
PDSC13	Port drive strength control register 13	32	0	R/W	4634_H	$0000\ 0000_H$	—	—	√
PPROTS13	Port protection status register 13	32	0	R	$4B34_H$	$0000\ 0000_H$	—	—	√
PPCMD13	Port protection command register 13	32	7-0	W	$4C34_H$	$xxxx\ xx00_H$	—	—	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2.10.12 Port 18 (P18)

2.10.12.1 Alternative Function

Table 2.59 Port 18 (P18)

Port Mode (PMC18_m = 0)	Alternative Mode (PMC18_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P18_0	CSIG1RYI	CSIG1RYO		PWGA61O											ADCA1I0S	√	√	√
P18_1		PWGA62O													ADCA1I1S	√	√	√
P18_2		PWGA63O													ADCA1I2S	√	√	√
P18_3															ADCA1I3S	√	—	—
		PWGA71O													ADCA1I3S	—	√	√
P18_4		CSIH1CSS4													ADCA1I4S	—	√	√
P18_5		CSIH1CSS5													ADCA1I5S	—	√	√
P18_6															ADCA1I6S	—	√	√
P18_7															ADCA1I7S	—	√	√
P18_8															ADCA1I8S	—	—	√
P18_9															ADCA1I9S	—	—	√
P18_10															ADCA1I10S	—	—	√
P18_11															ADCA1I11S	—	—	√
P18_12															ADCA1I12S	—	—	√
P18_13															ADCA1I13S	—	—	√
P18_14															ADCA1I14S	—	—	√
P18_15															ADCA1I15S	—	—	√

CAUTIONS

1. The behavior and performance are not guaranteed when alternative functions are not assigned to the register.
2. Use special functions with their initial settings.

2.10.12.2 Control Registers

Table 2.60 Control Registers (P18)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
P18	Port register 18	16	3-0	R/W	0048_H	0000_H	√	—	—
			7-0				—	√	—
			15-0				—	—	√
PSR18	Port set/reset register 18	32	19-16, 3-0	R/W	0148_H	$0000\ 0000_H$	√	—	—
			23-16, 7-0				—	√	—
			31-16, 15-0				—	—	√
PPR18	Port pin read register 18	16	3-0	R	0248_H	0000_H	√	—	—
			7-0				—	√	—
			15-0				—	—	√
PM18	Port mode register 18	16	3-0	R/W	0348_H	$FFFF_H$	√	—	—
			7-0				—	√	—
			15-0				—	—	√
PMC18	Port mode control register 18	16	2-0	R/W	0448_H	0000_H	√	—	—
			5-0				—	√	√
PFC18	Port function control register 18	16	0	R/W	0548_H	0000_H	√	√	√
PNOT18	Port NOT register 18	16	3-0	W	0748_H	0000_H	√	—	—
			7-0				—	√	—
			15-0				—	—	√
PMSR18	Port mode set/reset register 18	32	19-16, 3-0	R/W	0848_H	$0000\ FFFF_H$	√	—	—
			23-16, 7-0				—	√	—
			31-16, 15-0				—	—	√
PMCSR18	Port mode control set/reset register 18	32	18-16, 2-0	R/W	0948_H	$0000\ 0000_H$	√	—	—
			20-16, 5-0				—	√	√
PIBC18	Port input buffer control register 18	16	3-0	R/W	4048_H	0000_H	√	—	—
			7-0				—	√	—
			15-0				—	—	√
PBDC18	Port bidirection control register 18	16	3-0	R/W	4148_H	0000_H	√	—	—
			7-0				—	√	—
			15-0				—	—	√
PU18	Pull-up option register 18	16	3-0	R/W	4348_H	0000_H	√	—	—
			7-0				—	√	—
			15-0				—	—	√
PODC18	Port open drain control register 18	32	3-0	R/W	4548_H	$0000\ 0000_H$	√	—	—
			7-0				—	√	—
			15-0				—	—	√
PPROTS18	Port protection status register 18	32	0	R	$4B48_H$	$0000\ 0000_H$	√	√	√
PPCMD18	Port protection command register 18	32	7-0	W	$4C48_H$	$xxxx\ xx00_H$	√	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned. When writing to unused bits, write the value after reset.

2.10.13 Port 19 (P19)

2.10.13.1 Alternative Function

Table 2.61 Port 19 (P19)

Port Mode (PMC19_m = 0)	Alternative Mode (PMC19_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P19_0															ADCA1I16S	—	—	√
P19_1															ADCA1I17S	—	—	√
P19_2															ADCA1I18S	—	—	√
P19_3															ADCA1I19S	—	—	√

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2.10.13.2 Control Registers

Table 2.62 Control Registers (P19)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
P19	Port register 19	16	3-0	R/W	$004C_H$	0000_H	—	—	√
PSR19	Port set/reset register 19	32	19-16, 3-0	R/W	$014C_H$	$0000\ 0000_H$	—	—	√
PPR19	Port pin read register 19	16	3-0	R	$024C_H$	0000_H	—	—	√
PM19	Port mode register 19	16	3-0	R/W	$034C_H$	$FFFF_H$	—	—	√
PNOT19	Port NOT register 19	16	3-0	W	$074C_H$	0000_H	—	—	√
PMSR19	Port mode set/reset register 19	32	19-16, 3-0	R/W	$084C_H$	$0000\ FFFF_H$	—	—	√
PIBC19	Port input buffer control register 19	16	3-0	R/W	$404C_H$	0000_H	—	—	√
PBDC19	Port bidirection control register 19	16	3-0	R/W	$414C_H$	0000_H	—	—	√
PU19	Pull-up option register 19	16	3-0	R/W	$434C_H$	0000_H	—	—	√
PODC19	Port open drain control register 19	32	3-0	R/W	$454C_H$	$0000\ 0000_H$	—	—	√
PPROTS19	Port protection status register 19	32	0	R	$4B4C_H$	$0000\ 0000_H$	—	—	√
PPCMD19	Port protection command register 19	32	7-0	W	$4C4C_H$	$xxxx\ xx00_H$	—	—	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2.10.14 Port 20 (P20)

2.10.14.1 Alternative Function

Table 2.63 Port 20 (P20)

Port Mode (PMC20_m = 0)	Alternative Mode (PMC20_m = 1)														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
P20_0	RLIN26RX			PWGA64O												—	√	√
P20_1		RLIN26TX		PWGA65O												—	√	√
P20_2	CAN4RX/ INTP4			PWGA66O	RLIN29RX											—	√	√
P20_3		CAN4TX		PWGA67O		RLIN29TX										—	√	√
P20_4	RLIN23RX			PWGA59O												√	√	√
P20_5		RLIN23TX		PWGA60O												√	√	√

CAUTION

The behavior and performance are not guaranteed when alternative functions are not assigned to the register.

2.10.14.2 Control Registers

Table 2.64 Control Registers (P20)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
P20	Port register 20	16	5, 4	R/W	0050_H	0000_H	√	—	—
			5-0				—	√	√
PSR20	Port set/reset register 20	32	21, 20, 5, 4	R/W	0150_H	$0000\ 0000_H$	√	—	—
			21-16, 5-0				—	√	√
PPR20	Port pin read register 20	16	5, 4	R	0250_H	0000_H	√	—	—
			5-0				—	√	√
PM20	Port mode register 20	16	5, 4	R/W	0350_H	$FFFF_H$	√	—	—
			5-0				—	√	√
PMC20	Port mode control register 20	16	5, 4	R/W	0450_H	0000_H	√	—	—
			5-0				—	√	√
PFC20	Port function control register 20	16	5, 4	R/W	0550_H	0000_H	√	—	—
			5-0				—	√	√
PFCE20	Port function control expansion register 20	16	3, 2	R/W	0650_H	0000_H	—	√	√
PNOT20	Port NOT register 20	16	5, 4	W	0750_H	0000_H	√	—	—
			5-0				—	√	√
PMSR20	Port mode set/reset register 20	32	21, 20, 5, 4	R/W	0850_H	$0000\ FFFF_H$	√	—	—
			21-16, 5-0				—	√	√
PMCSR20	Port mode control set/reset register 20	32	21, 20, 5, 4	R/W	0950_H	$0000\ 0000_H$	√	—	—
			21-16, 5-0				—	√	√
PIBC20	Port input buffer control register 20	16	5, 4	R/W	4050_H	0000_H	√	—	—
			5-0				—	√	√
PBDC20	Port bidirection control register 20	16	5, 4	R/W	4150_H	0000_H	√	—	—
			5-0				—	√	√
PU20	Pull-up option register 20	16	5, 4	R/W	4350_H	0000_H	√	—	—
			5-0				—	√	√
PODC20	Port open drain control register 20	32	5, 4	R/W	4550_H	$0000\ 0000_H$	√	—	—
			5-0				—	√	√
PIS20	Port input buffer selection register 20	16	4	R/W	4750_H	$FFFF_H$	√	—	—
			4, 2, 0				—	√	√
PPROTS20	Port protection status register 20	32	0	R	$4B50_H$	$0000\ 0000_H$	√	√	√
PPCMD20	Port protection command register 20	32	7-0	W	$4C50_H$	$xxxx\ xx00_H$	√	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2.10.15 Analog Port 0 (AP0)

2.10.15.1 Alternative Function

Table 2.65 Analog Port 0 (AP0)

Port Mode	Alternative Mode														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
AP0_0															ADCA0I0	√	√	√
AP0_1															ADCA0I1	√	√	√
AP0_2															ADCA0I2	√	√	√
AP0_3															ADCA0I3	√	√	√
AP0_4															ADCA0I4	√	√	√
AP0_5															ADCA0I5	√	√	√
AP0_6															ADCA0I6	√	√	√
AP0_7															ADCA0I7	√	√	√
AP0_8															ADCA0I8	√	√	√
AP0_9															ADCA0I9	√	√	√
AP0_10															ADCA0I10	√	√	√
AP0_11															ADCA0I11	√	√	√
AP0_12															ADCA0I12	√	√	√
AP0_13															ADCA0I13	√	√	√
AP0_14															ADCA0I14	√	√	√
AP0_15															ADCA0I15	√	√	√

CAUTION

The special function can be used by setting the pin to input mode.

2.10.15.2 Control Registers

Table 2.66 Control Registers (AP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W			144 pins	176 pins	233 pins
AP0	Analog port register 0	16	15-0	R/W	$00C8_H$	0000_H	√	√	√
APSR0	Analog port set/reset register 0	32	31-16, 15-0	R/W	$01C8_H$	$0000\ 0000_H$	√	√	√
APPR0	Analog port pin read register 0	16	15-0	R	$02C8_H$	0000_H	√	√	√
APM0	Analog port mode register 0	16	15-0	R/W	$03C8_H$	$FFFF_H$	√	√	√
APNOT0	Analog port NOT register 0	16	15-0	W	$07C8_H$	0000_H	√	√	√
APMSR0	Analog port mode set/reset register 0	32	31-16, 15-0	R/W	$08C8_H$	$0000\ FFFF_H$	√	√	√
APIBC0	Analog port input buffer control register 0	16	15-0	R/W	$40C8_H$	0000_H	√	√	√
APBDC0	Analog port bidirection control register 0	16	15-0	R/W	$41C8_H$	0000_H	√	√	√

2.10.16 Analog Port 1 (AP1)

2.10.16.1 Alternative Function

Table 2.67 Analog Port 1 (AP1)

Port Mode	Alternative Mode														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
AP1_0															ADCA1I0	√	√	√
AP1_1															ADCA1I1	√	√	√
AP1_2															ADCA1I2	√	√	√
AP1_3															ADCA1I3	√	√	√
AP1_4															ADCA1I4	√	√	√
AP1_5															ADCA1I5	√	√	√
AP1_6															ADCA1I6	√	√	√
AP1_7															ADCA1I7	√	√	√
AP1_8															ADCA1I8	—	√	√
AP1_9															ADCA1I9	—	√	√
AP1_10															ADCA1I10	—	√	√
AP1_11															ADCA1I11	—	√	√
AP1_12															ADCA1I12	—	√	√
AP1_13															ADCA1I13	—	√	√
AP1_14															ADCA1I14	—	√	√
AP1_15															ADCA1I15	—	√	√

CAUTION

The special function can be used by setting the pin to input mode.

2.10.16.2 Control Registers

Table 2.68 Control Registers (AP1)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
AP1	Analog port register 1	16	7-0	R/W	$00CC_H$	0000_H	√	—	—
			15-0				—	√	√
APSR1	Analog port set/reset register 1	32	23-16, 7-0	R/W	$01CC_H$	$0000\ 0000_H$	√	—	—
			31-16, 15-0				—	√	√
APPR1	Analog port pin read register 1	16	7-0	R	$02CC_H$	0000_H	√	—	—
			15-0				—	√	√
APM1	Analog port mode register 1	16	7-0	R/W	$03CC_H$	$FFFF_H$	√	—	—
			15-0				—	√	√
APNOT1	Analog port NOT register 1	16	7-0	W	$07CC_H$	0000_H	√	—	—
			15-0				—	√	√
APMSR1	Analog port mode set/reset register 1	32	23-16, 7-0	R/W	$08CC_H$	$0000\ FFFF_H$	√	—	—
			31-16, 15-0				—	√	√
APIBC1	Analog port input buffer control register 1	16	7-0	R/W	$40CC_H$	0000_H	√	—	—
			15-0				—	√	√
APBDC1	Analog port bidirection control register 1	16	7-0	R/W	$41CC_H$	0000_H	√	—	—
			15-0				—	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

2.10.17 Input Port 0 (IP0)

2.10.17.1 Alternative Function

Table 2.69 Input Port 0 (IP0)

Port Mode	Alternative Mode														Special Function	Device		
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative		6th Alternative		7th Alternative			144 pins	176 pins	233 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output				
IP0_0															XT2	√	√	√

2.10.17.2 Control Registers

Table 2.70 Control Registers (IP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device		
			Position	R/W*1			144 pins	176 pins	233 pins
IPPR0	Input port pin read register 0	16	0	R	$02F0_H$	0000_H	√	√	√
IPIBC0	Port input buffer control register 0	16	0	R/W	$40F0_H$	0000_H	√	√	√

Note 1. The unused bits are read-only (R). When read, the value after reset is returned.
When writing to unused bits, write the value after reset.

CAUTION

When the IP0_0/XT2 pin is used as an input port, set the IPIBC0_0 bit to 1 while stopping the SOSC operation. For details on the settings for SOSC operations, see **Section 11.4.2.6, SOSCE — SubOSC Enable Register**. When the IP0_0/XT2 pin is used for the SubOSC (SOSC) not as an input port, set the IPIBC0_0 bit to 0.

2.11 Port (Special I/O) Function Overview

This section describes the port (special I/O) functions.

2.11.1 Special I/O after Reset

The special port function after reset release is shown below.

2.11.1.1 P0_0: $\overline{\text{RESETOUT}}$

The P0_0 pin outputs low level while a reset is asserted and continues to output low level after the reset is released due to the initial value of the registers as follows. This pin can be used as $\overline{\text{RESETOUT}}$ function.

- P0.P0_0 = 0: low level
- PM0.PM0_0 = 0: output port
- PODC0.PODC0_0 = 1: open-drain output

When the P0_0 pin setting is updated with another value, the pin operates by new setting.

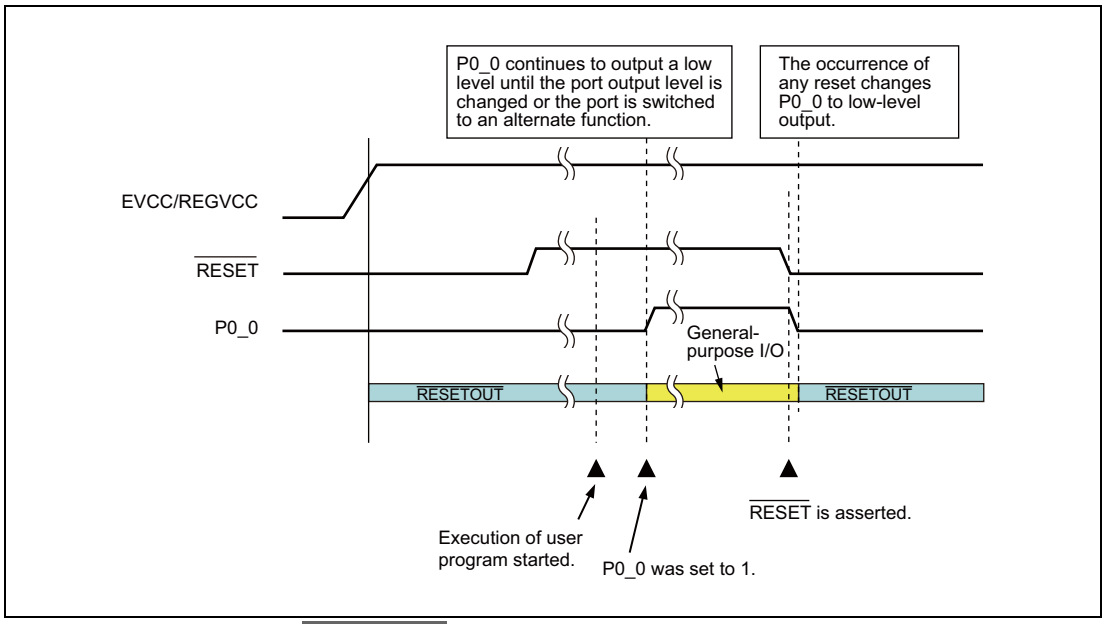


Figure 2.9 P0_0 Pin ($\overline{\text{RESETOUT}}$ Signal) Operation while a Reset is asserted and released

2.11.1.2 JP0_0 to JP0_6: Debug Interface

If the OPJTAG[1:0] setting is the combination below, the pins of the JTAG port group can be used as a debug interface after reset release.

Table 2.71 Debug Intterface

OPJTAG1	OPJTAG0	Mode	JP0_0	JP0_1	JP0_2	JP0_3	JP0_4	JP0_5	JP0_6
1	1	Nexus I/F	DCUTDI input	DCUTDO output	DCUTCK input	DCUTMS input	$\overline{\text{DCUTRST}}$ input	$\overline{\text{DCURDY}}$ output	$\overline{\text{EVTO}}$*1 output
0	1	LPD (4-pin)	LPDI input	LPDO output	LPDCLK input	Port/ alternative function	Port/ alternative function	LPDCLK OUT output	Port/ alternative function
1	0	LPD (1-pin)	LPDIO input/ output	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function

Consequently, the port and alternative functions on these pins cannot be used while the debugger is connected.

Note 1. The $\overline{\text{EVTO}}$ pin is only available in devices with 4-MB code flash memory.

NOTE

For the OPJTAG[1:0] settings, see **Section 36.9.2, OPBT0 — Option Byte 0**.

2.11.1.3 FPDR (JP0_0), FPDT (JP0_1), FPCK (JP0_2): Flash Programmer

These pins are used for connecting a flash programmer. See *Flash Programmer's Manual* for details.

2.11.1.4 Mode Pins

The FLMD0 pin in combination with the P10_8: FLMD1 pin can select serial programming mode. In addition, the P10_1: MODE0 pin in combination with the P10_2: MODE1 pin can select boundary scan mode. For details on the mode selection, see **Section 6, Operating Mode**.

2.11.1.5 IP0_0: XT2

This pin is the SubOSC (SOSC) input pin. When the IPIBC0_0 bit = 1, the IP0_0/XT2 pin is used as an input port. If you make this setting, stop SOSC operation at the same time.

2.11.2 A/D Input Alternative I/O

The following ports are permanently connected to A/D input functions. (However, an analog input to the A/D is controlled by the A/D module.)

Table 2.72 A/D Input Alternative Pins (1/2)

Port	A/D Input	Device		
		144 pins	176 pins	233 pins
P8_0	ADCA0I0S	√	√	√
P8_1	ADCA0I1S	√	√	√
P8_2	ADCA0I4S	√	√	√
P8_3	ADCA0I5S	√	√	√
P8_4	ADCA0I6S	√	√	√
P8_5	ADCA0I7S	√	√	√
P8_6	ADCA0I8S	√	√	√
P8_7	ADCA0I14S	√	√	√
P8_8	ADCA0I15S	√	√	√
P8_9	ADCA0I16S	√	√	√
P8_10	ADCA0I17S	√	√	√
P8_11	ADCA0I18S	√	√	√
P8_12	ADCA0I19S	√	√	√
P9_0	ADCA0I2S	√	√	√
P9_1	ADCA0I3S	√	√	√
P9_2	ADCA0I9S	√	√	√
P9_3	ADCA0I10S	√	√	√
P9_4	ADCA0I11S	√	√	√
P18_0	ADCA1I0S	√	√	√
P18_1	ADCA1I1S	√	√	√
P18_2	ADCA1I2S	√	√	√
P18_3	ADCA1I3S	√	√	√
P18_4	ADCA1I4S	—	√	√
P18_5	ADCA1I5S	—	√	√
P18_6	ADCA1I6S	—	√	√
P18_7	ADCA1I7S	—	√	√
P18_8	ADCA1I8S	—	—	√
P18_9	ADCA1I9S	—	—	√
P18_10	ADCA1I10S	—	—	√
P18_11	ADCA1I11S	—	—	√
P18_12	ADCA1I12S	—	—	√
P18_13	ADCA1I13S	—	—	√
P18_14	ADCA1I14S	—	—	√
P18_15	ADCA1I15S	—	—	√
P19_0	ADCA1I16S	—	—	√
P19_1	ADCA1I17S	—	—	√
P19_2	ADCA1I18S	—	—	√
P19_3	ADCA1I19S	—	—	√
AP0_0	ADCA0I0	√	√	√

Table 2.72 A/D Input Alternative Pins (2/2)

Port	A/D Input	Device		
		144 pins	176 pins	233 pins
AP0_1	ADCA0I1	√	√	√
AP0_2	ADCA0I2	√	√	√
AP0_3	ADCA0I3	√	√	√
AP0_4	ADCA0I4	√	√	√
AP0_5	ADCA0I5	√	√	√
AP0_6	ADCA0I6	√	√	√
AP0_7	ADCA0I7	√	√	√
AP0_8	ADCA0I8	√	√	√
AP0_9	ADCA0I9	√	√	√
AP0_10	ADCA0I10	√	√	√
AP0_11	ADCA0I11	√	√	√
AP0_12	ADCA0I12	√	√	√
AP0_13	ADCA0I13	√	√	√
AP0_14	ADCA0I14	√	√	√
AP0_15	ADCA0I15	√	√	√
AP1_0	ADCA1I0	√	√	√
AP1_1	ADCA1I1	√	√	√
AP1_2	ADCA1I2	√	√	√
AP1_3	ADCA1I3	√	√	√
AP1_4	ADCA1I4	√	√	√
AP1_5	ADCA1I5	√	√	√
AP1_6	ADCA1I6	√	√	√
AP1_7	ADCA1I7	√	√	√
AP1_8	ADCA1I8	—	√	√
AP1_9	ADCA1I9	—	√	√
AP1_10	ADCA1I10	—	√	√
AP1_11	ADCA1I11	—	√	√
AP1_12	ADCA1I12	—	√	√
AP1_13	ADCA1I13	—	√	√
AP1_14	ADCA1I14	—	√	√
AP1_15	ADCA1I15	—	√	√

2.11.3 Special I/O Control

2.11.3.1 Direct I/O Control (PIPC)

Some alternative functions take over the input and output control of the ports.

The following table lists all alternative functions where PIPCn.PIPCn_m must be set to 1. For details, see **Section 2.9.2.3, PIPCn — Port IP Control Register**.

Table 2.73 Alternative Modes that Require Setting PIPCn.PIPCn_m = 1 (1/2)

Function	Alternative Function Name	Port Name	Power Supply Area	Control	Reference Section
MEMC	MEMC0AD0	P10_6	ISO	—	**Section 14**
	MEMC0AD1	P10_7	ISO	—	
	MEMC0AD2	P10_8	ISO	—	
	MEMC0AD3	P10_9	ISO	—	
	MEMC0AD4	P10_10	ISO	—	
	MEMC0AD5	P10_11	ISO	—	
	MEMC0AD6	P10_12	ISO	—	
	MEMC0AD7	P10_13	ISO	—	
	MEMC0AD8	P10_14	ISO	—	
	MEMC0AD9	P11_1	ISO	—	
	MEMC0AD10	P11_2	ISO	—	
	MEMC0AD11	P11_3	ISO	—	
	MEMC0AD12	P11_4	ISO	—	
	MEMC0AD13	P11_5	ISO	—	
	MEMC0AD14	P11_6	ISO	—	
	MEMC0AD15	P11_7	ISO	—	
TAPA	TAPA0UP	P10_0	ISO	U phase Hi-Z control	**Section 29**
	TAPA0UN	P10_1	ISO		
	TAPA0VP	P10_2	ISO	V phase Hi-Z control	
	TAPA0VN	P10_3	ISO		
	TAPA0WP	P10_4	ISO	W phase Hi-Z control	
	TAPA0WN	P10_5	ISO		
CSIG	CSIG0SO	P0_13	AWO	Serial data output control signal	**Section 15**
		P10_6	ISO		
	CSIG0SC	P0_14	AWO	Master (1) / slave (0) mode signal	
		P10_7	ISO		
	CSIG1SO	P11_9	ISO	Serial data output control signal	
	CSIG1SC	P11_10	ISO	Master (1) / slave (0) mode signal	
CSIH	CSIH0SO	P0_3	AWO	Serial data output control signal	**Section 16**
	CSIH0SC	P0_2	AWO	Master (1) / slave (0) mode signal	
	CSIH1SO	P0_5	AWO	Serial data output control signal	
		P10_2	ISO		

Table 2.73 Alternative Modes that Require Setting PIPCn.PIPCn_m = 1 (2/2)

Function	Alternative Function Name	Port Name	Power Supply Area	Control	Reference Section
CSIH	CSIH1SC	P0_6	AWO	Master (1) / slave (0) mode signal	**Section 16**
		P10_1	ISO		
	CSIH2SO	P11_2	ISO	Serial data output control signal	
	CSIH2SC	P11_3	ISO	Master (1) / slave (0) mode signal	
	CSIH3SO	P11_6	ISO	Serial data output control signal	
	CSIH3SC	P11_7	ISO	Master (1) / slave (0) mode signal	

2.11.3.2 Input Buffer Control (PISn, JPISA0)

The port input buffer characteristics (type 1/type 2) of this device can be selected using the PISn register. The applicable pins are shown in the following table.

The JTAG port input buffer characteristics (type 2/type 5) of this device can be selected using the JPISA0 register. The applicable pins are shown in **Table 2.75**.

The pins not listed in the table only support type 2 (SHMT4).

Table 2.74 Input Buffer Characteristics Selection (1/2)

Port Name	Input Buffer Selection		Device		
	Type 1 (PISn_m = 0)	Type 2 (PISn_m = 1)	144 pins	176 pins	233 pins
P0_0	SHMT1	SHMT4	√	√	√
P0_1	SHMT1	SHMT4	√	√	√
P0_2	SHMT1	SHMT4	√	√	√
P0_3	SHMT1	SHMT4	√	√	√
P0_4	SHMT1	SHMT4	√	√	√
P0_5	SHMT1	SHMT4	√	√	√
P0_6	SHMT1	SHMT4	√	√	√
P0_7	SHMT1	SHMT4	√	√	√
P0_9	SHMT1	SHMT4	√	√	√
P0_11	SHMT1	SHMT4	√	√	√
P0_12	SHMT1	SHMT4	√	√	√
P0_13	SHMT1	SHMT4	√	√	√
P1_0	SHMT1	SHMT4	√	√	√
P1_2	SHMT1	SHMT4	√	√	√
P1_4	SHMT1	SHMT4	√	√	√
P1_6	SHMT1	SHMT4	√	√	√
P1_8	SHMT1	SHMT4	√	√	√
P1_10	SHMT1	SHMT4	√	√	√
P1_12	SHMT1	SHMT4	—	√	√
P1_14	SHMT1	SHMT4	—	√	√
P2_0	SHMT1	SHMT4	—	√	√
P2_2	SHMT1	SHMT4	—	√	√
P2_4	SHMT1	SHMT4	—	√	√
P10_0	SHMT1	SHMT4	√	√	√
P10_2	SHMT1	SHMT4	√	√	√
P10_3	SHMT1	SHMT4	√	√	√
P10_4	SHMT1	SHMT4	√	√	√
P10_6	SHMT1	SHMT4	√	√	√
P10_9	SHMT1	SHMT4	√	√	√
P10_11	SHMT1	SHMT4	√	√	√
P10_12	SHMT1	SHMT4	√	√	√
P10_13	SHMT1	SHMT4	√	√	√
P10_14	SHMT1	SHMT4	√	√	√
P10_15	SHMT1	SHMT4	√	√	√

Table 2.74 Input Buffer Characteristics Selection (2/2)

Port Name	Input Buffer Selection		Device		
	Type 1 (PISn_m = 0)	Type 2 (PISn_m = 1)	144 pins	176 pins	233 pins
P11_1	SHMT1	SHMT4	√	√	√
P11_3	SHMT1	SHMT4	√	√	√
P11_5	SHMT1	SHMT4	√	√	√
P11_6	SHMT1	SHMT4	√	√	√
P11_9	SHMT1	SHMT4	√	√	√
P11_12	SHMT1	SHMT4	√	√	√
P11_13	SHMT1	SHMT4	√	√	√
P11_15	SHMT1	SHMT4	√	√	√
P12_1	SHMT1	SHMT4	√	√	√
P12_3	SHMT1	SHMT4	—	√	√
P20_0	SHMT1	SHMT4	—	√	√
P20_2	SHMT1	SHMT4	—	√	√
P20_4	SHMT1	SHMT4	√	√	√

Table 2.75 JTAG port Input Buffer Characteristic Selection

Port Name	Input Buffer Selection			Device		
	Type 1	Type 2 (JPISA0_m = 0)	Type 5 (JPISA0_m = 1)	144 pins	176 pins	233 pins
JP0_0	—	SHMT4	TTL*1,*2,*3,*4	√	√	√
JP0_2	—	SHMT4	TTL*1,*2,*3	√	√	√
JP0_3	—	SHMT4	TTL*1,*2	√	√	√
JP0_4	—	SHMT4	—*1,*2	√	√	√

Note 1. TTL is selected for Boundary scan mode without JPISA0 register setting.
Note 2. TTL is selected for Nexus in normal operating mode without JPISA0 register setting.
Note 3. TTL is selected for LPD (4-pin) in normal operating mode without JPISA0 register setting.
Note 4. TTL is selected for LPD (1-pin) in normal operating mode without JPISA0 register setting.

NOTES

1. For the SHMT1, SHMT4 and TTL pin characteristics, see the data sheet.
2. For the input buffer after reset, type 2 (SHMT4) is selected.

2.11.3.3 Output Buffer Control (PDSC)

The port output driver strength (slow mode/fast mode) can be selected using the PDSCn register. The applicable pins are shown in the following table. Only slow mode is supported for ports other than those listed below.

Table 2.76 Pins with Selectable Output Driving Ability (1/2)

Port Name	Device		
	144 pins	176 pins	233 pins
P0_2*1	√	√	√
P0_3*1	√	√	√
P0_5*1	√	√	√
P0_6*1	√	√	√
P0_7	√	√	√
P0_13	√	√	√
P0_14	√	√	√
P10_0	√	√	√
P10_1*2	√	√	√
P10_2*2	√	√	√
P10_3	√	√	√
P10_4	√	√	√
P10_5	√	√	√
P10_6	√	√	√
P10_7	√	√	√
P10_8	√	√	√
P10_9	—	√	√
P10_10	—	√	√
P10_11	√	√	√
P10_12	—	√	√
P10_13	√	√	√
P10_14	—	√	√
P10_15	—	√	√
P11_0	—	√	√
P11_1	√	√	√
P11_2*2	√	√	√
P11_3*2	√	√	√
P11_4	—	√	√
P11_5	—	√	√
P11_6*2	√	√	√
P11_7*2	√	√	√
P11_8*2	—	√	√
P11_9	√	√	√
P11_10	√	√	√
P11_11	—	√	√
P11_13	—	√	√
P11_14	—	√	√
P11_15	—	√	√

Table 2.76 Pins with Selectable Output Driving Ability (2/2)

Port Name	Device		
	144 pins	176 pins	233 pins
P12_0	—	√	√
P12_1	—	√	√
P12_2	—	√	√
P13_0	—	—	√

Note 1. Set fast mode if the load capacitance of CSIH is 100 pF.

Note 2. Set fast mode if the load capacitance of CSIH is 50 pF.

2.12 Noise Filter & Edge/Level Detector

The input signals at some pins are passed through a filter to remove noise and glitches. The RH850/F1M supports both analog and digital filters.

It also supports the function for edge and level detection after the signals have passed through a filter.

The first part of this section provides an overview of port input pins that are equipped with a filter and the filter type, noise filter & edge/level detection control registers and bits, and register addresses.

For details on the digital/analog filter function and noise filter & edge/level detection control registers, see **Section 2.13, Description of Port Noise Filter & Edge/Level Detection**.

NOTE

In this section, <name> in the noise filter control register represents the peripheral function connected to a filter.

2.12.1 Port Filter Assignment

A list of the input pins that incorporate an analog or digital filter is provided below.

2.12.1.1 Input Pins that Incorporate Analog Filter Type A

The input pins of analog filter type A incorporate an analog filter and edge/level detection function. Edge/level detection is controlled by the following registers.

- Filter control register FCLA0CTLm_<name> (m = 0 to 7)
  A dedicated FCLA0CTLm_<name> register is provided for each pin in a port that incorporates an analog filter.

Table 2.77 Input Pins that Incorporate Analog Filter Type A

	FCLA0CTL Register Configuration		Device		
Input Signal	**Register**	**Address**	**144 pins**	**176 pins**	**233 pins**
NMI	FCLA0CTL0_NMI	FFC3 4000$_H$	√	√	√
INTP0	FCLA0CTL0_INTPL	FFC3 4020$_H$	√	√	√
INTP1	FCLA0CTL1_INTPL	FFC3 4024$_H$	√	√	√
INTP2	FCLA0CTL2_INTPL	FFC3 4028$_H$	√	√	√
INTP3	FCLA0CTL3_INTPL	FFC3 402C$_H$	√	√	√
INTP4	FCLA0CTL4_INTPL	FFC3 4030$_H$	√	√	√
INTP5	FCLA0CTL5_INTPL	FFC3 4034$_H$	√	√	√
INTP6	FCLA0CTL6_INTPL	FFC3 4038$_H$	√	√	√
INTP7	FCLA0CTL7_INTPL	FFC3 403C$_H$	√	√	√
INTP8	FCLA0CTL0_INTPH	FFC3 4040$_H$	√	√	√
INTP9	FCLA0CTL1_INTPH	FFC3 4044$_H$	√	√	√
INTP10	FCLA0CTL2_INTPH	FFC3 4048$_H$	√	√	√
INTP11	FCLA0CTL3_INTPH	FFC3 404C$_H$	√	√	√
INTP12	FCLA0CTL4_INTPH	FFC3 4050$_H$	√	√	√
INTP13	FCLA0CTL5_INTPH	FFC3 4054$_H$	√	√	√
INTP14	FCLA0CTL6_INTPH	FFC3 4058$_H$	√	√	√
INTP15	FCLA0CTL7_INTPH	FFC3 405C$_H$	√	√	√

2.12.1.2 Input Pins that Incorporate Analog Filter Type B

The input pins of analog filter type B incorporate an analog filter. Edge/level detection is controlled by the registers for individual peripheral functions.

Table 2.78 Input Pins that Incorporate Analog Filter Type B

Input Signal	Edge/Level Detection	Device		
		144 pins	176 pins	233 pins
TAUJ0I0	Edge detection*1	√	√	√
TAUJ0I1	Edge detection*1	√	√	√
TAUJ0I2	Edge detection*1	√	√	√
TAUJ0I3	Edge detection*1	√	√	√
TAUJ1I0	Edge detection*1	√	√	√
TAUJ1I1	Edge detection*1	√	√	√
TAUJ1I2	Edge detection*1	√	√	√
TAUJ1I3	Edge detection*1	√	√	√
TAPA0ESO	Edge detection*2	√	√	√
KR0I0	Low level detection	√	√	√
KR0I1	Low level detection	√	√	√
KR0I2	Low level detection	√	√	√
KR0I3	Low level detection	√	√	√
KR0I4	Low level detection	√	√	√
KR0I5	Low level detection	√	√	√
KR0I6	Low level detection	√	√	√
KR0I7	Low level detection	√	√	√

Note 1. For details on edge detection for TAUJ, see **Section 26.3.3.4, TAUJnCMURm — TAUJn Channel Mode User Register**.

Note 2. For details on edge detection for TAPA, see **Section 29.3.2, TAPAnCTL0 — TAPA Control Register 0**.

2.12.1.3 Input Pins that Incorporate Analog Filter Type C

The input pins of analog filter type C only incorporate an analog filter function.

Table 2.79 Input Pins that Incorporate Analog Filter Type C

Input Signal
FLMD0
FLMD1
MODE0
MODE1
$\overline{\text{RESET}}$
DCUTDI
$\overline{\text{DCUTRST}}$

2.12.1.4 Input Pins that Incorporate Digital Filter Type D

The input pins of digital filter type D incorporate a digital filter and edge detection function. The digital filter and edge detection are controlled by the following registers.

- Filter control register FCLA0CTLm_<name> (m = 0)
  Each port with a digital filter has a special FCLA0CTLm_<name> register.
- Digital noise elimination control register DNFA<name>CTL
  Each DNFA<name>CTL control register controls digital filter processing for three input signals per group.
- Digital noise elimination enable register DNFA<name>EN
  The setting of the DNFA<name>ENL[2:0] bits in DNFA<name>EN enables or disables digital noise elimination for three input signals per group.

Table 2.80 Input Pins that Incorporate Digital Filter Type D

Input Pin	Device			Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Filter Control Register	
	144 pins	176 pins	233 pins	Control Register	Address	Control Register	Control Bit	Address	Control Register	Address
ADCA0TRG0	√	√	√	DNFA ADCTL0CTL	FFC3 00A0$_H$	DNFA ADCTL0EN (DNFAA DCTL0ENL)	DNFAADCTL0 ENL0	FFC3 00A4$_H$ (FFC3 00AC$_H$)	FCLA0CTL0 _ADC0	FFC3 4060$_H$
ADCA0TRG1	√	√	√				DNFAADCTL0 ENL1		FCLA0CTL1 _ADC0	FFC3 4064$_H$
ADCA0TRG2	√	√	√				DNFAADCTL0 ENL2		FCLA0CTL2 _ADC0	FFC3 4068$_H$
ADCA1TRG0	√	√	√	DNFA ADCTL1CTL	FFC3 00C0$_H$	DNFA ADCTL1EN (DNFAA DCTL1ENL)	DNFAADCTL1 ENL0	FFC3 00C4$_H$ (FFC3 00CC$_H$)	FCLA0CTL0 _ADC1	FFC3 4080$_H$
ADCA1TRG1	√	√	√				DNFAADCTL1 ENL1		FCLA0CTL1 _ADC1	FFC3 4084$_H$
ADCA1TRG2	√	√	√				DNFAADCTL1 ENL2		FCLA0CTL2 _ADC1	FFC3 4088$_H$

2.12.1.5 Input Pins that Incorporate Digital Filter Type E

The input pins of digital filter type E incorporate a digital filter. The digital filter is controlled by the following registers. Edge detection is controlled by the registers for individual peripheral functions.

- Digital noise elimination control register DNFA<name>CTL
  Each DNFA<name>CTL control register controls digital filter processing for up to 16 input signals per group.
- Digital noise elimination enable register DNFA<name>EN
  The setting of the DNFA<name>ENL[7:0] and DNFA<name>ENH[7:0] bits in DNFA<name>EN enables or disables digital noise elimination for up to 16 input signals per group.

Table 2.81 Input Pins that Incorporate Digital Filter Type E (1/2)

Input Pin	Device			Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Edge Detection
	144 pins	176 pins	233 pins	Control Register	Address	Control Register	Control Bit	Address	Register Name
TAUD0I0	√	√	√	DNFA TAUD0ICTL	FFC3 0000_H	DNFA TAUD0IEN (DNFA TAUD0IENH/ DNFA TAUD0IENL)	DNFATAUD0IENL0	FFC3 0004_H (FFC3 0008_H/ FFC3 $000C_H$)	*1
TAUD0I1	√	√	√				DNFATAUD0IENL1		
TAUD0I2	√	√	√				DNFATAUD0IENL2		
TAUD0I3	√	√	√				DNFATAUD0IENL3		
TAUD0I4	√	√	√				DNFATAUD0IENL4		
TAUD0I5	√	√	√				DNFATAUD0IENL5		
TAUD0I6	√	√	√				DNFATAUD0IENL6		
TAUD0I7	√	√	√				DNFATAUD0IENL7		
TAUD0I8	√	√	√				DNFATAUD0IENH0		
TAUD0I9	√	√	√				DNFATAUD0IENH1		
TAUD0I10	√	√	√				DNFATAUD0IENH2		
TAUD0I11	√	√	√				DNFATAUD0IENH3		
TAUD0I12	√	√	√				DNFATAUD0IENH4		
TAUD0I13	√	√	√				DNFATAUD0IENH5		
TAUD0I14	√	√	√				DNFATAUD0IENH6		
TAUD0I15	√	√	√				DNFATAUD0IENH7		
TAUB0I0	√	√	√	DNFA TAUB0ICTL	FFC3 0020_H	DNFA TAUB0IEN (DNFA TAUB0IENH/ DNFA TAUB0IENL)	DNFATAUB0IENL0	FFC3 0024_H (FFC3 0028_H/ FFC3 $002C_H$)	*2
TAUB0I1	√	√	√				DNFATAUB0IENL1		
TAUB0I2	√	√	√				DNFATAUB0IENL2		
TAUB0I3	√	√	√				DNFATAUB0IENL3		
TAUB0I4	√	√	√				DNFATAUB0IENL4		
TAUB0I5	√	√	√				DNFATAUB0IENL5		
TAUB0I6	√	√	√				DNFATAUB0IENL6		
TAUB0I7	√	√	√				DNFATAUB0IENL7		
TAUB0I8	√	√	√				DNFATAUB0IENH0		
TAUB0I9	√	√	√				DNFATAUB0IENH1		
TAUB0I10	√	√	√				DNFATAUB0IENH2		
TAUB0I11	√	√	√				DNFATAUB0IENH3		
TAUB0I12	√	√	√				DNFATAUB0IENH4		
TAUB0I13	√	√	√				DNFATAUB0IENH5		
TAUB0I14	√	√	√				DNFATAUB0IENH6		
TAUB0I15	√	√	√				DNFATAUB0IENH7		

Table 2.81 Input Pins that Incorporate Digital Filter Type E (2/2)

Input Pin	Device			Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Edge Detection
	144 pins	176 pins	233 pins	Control Register	Address	Control Register	Control Bit	Address	Register Name
TAUB1I0	—	√	√	DNFA TAUB1ICTL	FFC3 0040$_H$	DNFA TAUB1IEN (DNFA TAUB1IENH/ DNFA TAUB1IENL)	DNFATAUB1IENL0	FFC3 0044$_H$ (FFC3 0048$_H$/ FFC3 004C$_H$)	*2
TAUB1I1	—	√	√				DNFATAUB1IENL1		
TAUB1I2	—	√	√				DNFATAUB1IENL2		
TAUB1I3	—	√	√				DNFATAUB1IENL3		
TAUB1I4	—	√	√				DNFATAUB1IENL4		
TAUB1I5	—	√	√				DNFATAUB1IENL5		
TAUB1I6	—	√	√				DNFATAUB1IENL6		
TAUB1I7	—	√	√				DNFATAUB1IENL7		
TAUB1I8	—	√	√				DNFATAUB1IENH0		
TAUB1I9	—	√	√				DNFATAUB1IENH1		
TAUB1I10	—	√	√				DNFATAUB1IENH2		
TAUB1I11	—	√	√				DNFATAUB1IENH3		
TAUB1I12	—	√	√				DNFATAUB1IENH4		
TAUB1I13	—	√	√				DNFATAUB1IENH5		
TAUB1I14	—	√	√				DNFATAUB1IENH6		
TAUB1I15	—	√	√				DNFATAUB1IENH7		
ENCA0TIN0	√	√	√	DNFA ENCA0ICTL	FFC3 0060$_H$	DNFA ENCA0IEN (DNFA ENCA0IENL)	DNFAENCA0IENL0	FFC3 0064$_H$ (FFC3 006C$_H$)	*3
ENCA0TIN1	√	√	√				DNFAENCA0IENL1		
ENCA0E0	√	√	√				DNFAENCA0IENL2		
ENCA0E1	√	√	√				DNFAENCA0IENL3		
ENCA0EC	√	√	√				DNFAENCA0IENL4		

Note 1. For the setting for TAUD edge detection, see **Section 25.3.3.4, TAUDnCMURm — TAUDn Channel Mode User Register**.

Note 2. For the setting for TAUB edge detection, see **Section 24.3.3.4, TAUBnCMURm — TAUBn Channel Mode User Register**

Note 3. For the setting for ENCA edge detection, see **Section 28.3.3, ENCAnIOC0 — ENCAn I/O Control Register 0**.

2.12.2 Clock Supply for Port Filters

The following table shows the clock supply for each filter type in each port domain.

Table 2.82 Clock Supply for Port Filters

Peripheral Function	Port Domain*1	Filter Type	Filter Clock	Setting Register	
				Source Clock Selection	Clock Selection
ADCA0	Always-On area	Digital filter type D	DNFATCKI	CKSC_AADCAS_CTL	CKSC_AADCAD_CTL
ADCA1	Isolated area	Digital filter type D	DNFATCKI	CKSC_IADCAS_CTL	CKSC_IADCAD_CTL
TAUD0	Isolated area	Digital filter type E	DNFATCKI	CKSC_IPERI1S_CTL	—
TAUB0	Isolated area	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—
TAUB1	Isolated area	Digital filter type E	DNFATCKI	CKSC_IPERI2S_CTL	—
ENCA0	Isolated area	Digital filter type E	DNFATCKI	CKSC_IPERI1S_CTL	—

Note 1. Power domain

NOTE

For the Setting Register, see **Section 11.4.3, Clock Selector Control Register**.

2.13 Description of Port Noise Filter & Edge/Level Detection

External signals pass through different types of filters according to the use of each external input signal.

NOTE

In this section, <name> in the noise filter control register represents the peripheral function connected to a filter.

2.13.1 Overview

2.13.1.1 Analog Filter Types

Analog filters have fixed characteristics.

- Type A: An analog filter with edge detection or level detection.
  Used for external interrupt signals.
- Type B: An analog filter
  Edge detection is performed by each peripheral function. Used for the timer input signals, asynchronous Hi-Z control input signals, and key return input signals.
- Type C: An analog filter only
  Used for the external $\overline{\text{RESET}}$ input and mode signals.

2.13.1.2 Digital Filter Types

The digital filter characteristics can be adjusted to suit the application.

- Type D: A digital filter with edge detection.
  Used for the A/D converter external trigger pin.
- Type E: A digital filter. Edge detection is performed by each peripheral function.
  Used for the timer input signals and encoder input signals.

2.13.2 Analog Filters

2.13.2.1 Analog Filter Characteristic

See the data sheet for the input conditions for signals input to pins that incorporate an analog filter.

2.13.2.2 Analog Filter Control Registers

A dedicated FCLA0CTLm_<name> register or control register in the peripheral macro is provided for input pins that incorporate an analog filter.

The assignment of the input signals to the control registers and their addresses are given in **Table 2.77, Input Pins that Incorporate Analog Filter Type A**, in **Section 2.12.1, Port Filter Assignment**.

2.13.2.3 Analog Filter in Standby Mode

Analog filters for the wake-up capability from DeepSTOP mode are located in the Always-On-Area (AWO). Analog filters in the AWO area always operate.

The behavior of an analog filter in standby mode and its wake-up capability depend on the filter type. Refer to the description below for the type of analog filters.

(1) Analog Filter Type A

A block diagram of analog filter type A is shown below.

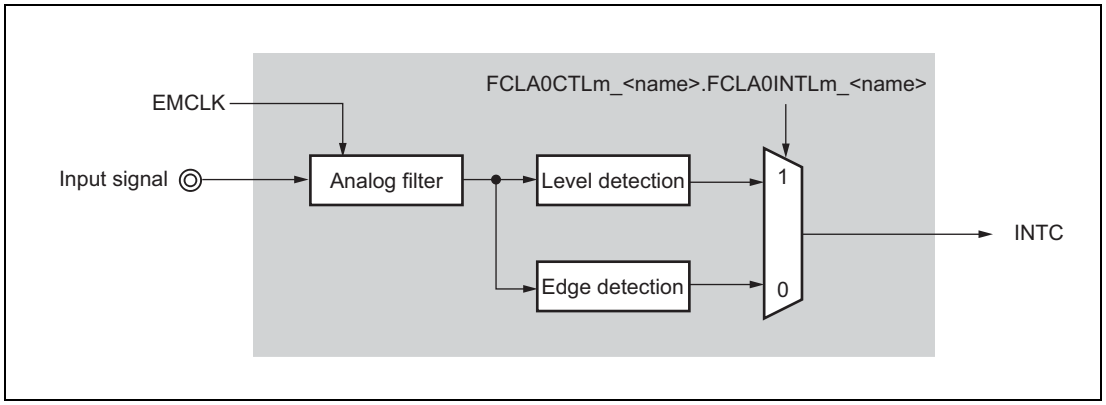


Figure 2.10 Block Diagram of Analog Filter Type A

After passing an external signal through the filter to eliminate noise and glitch, an output signal is generated according to whether an event is detected; that is whether a specified level is detected or whether a change in the level (an edge) occurs.

Whether a level or an edge is detected is selected by the control bit FCLA0CTLm_<name>.FCLA0INTLm_<name>.

- FCLA0INTLm_<name> bit = 0: Edge detection

  Whether a rising or falling edge is detected can be specified by setting the FCLA0CTLm_<name>.FCLA0INTRm_<name> and FCLA0CTLm_<name>.FCLA0INTFm_<name> bits.

- FCLA0INTLm_<name> bit = 1: Level detection

  The detection of a high level or low level can be specified by setting FCLA0CTLm_<name>.FCLA0INTRm_<name> bit.

The table below summarizes the detection conditions of the analog filter.

Table 2.83 Analog Filter Event Detection Conditions

FCLA0INTLm_<name>	FCLA0INTFm_<name>	FCLA0INTRm_<name>	Edge Detection	Level Detection
0	0	0	No edge detected	Disabled
	0	1	Rising edge	
	1	0	Falling edge	
	1	1	Rising and falling edges	
1	X	0	Disabled	Low level
	X	1		High level

Analog filter type A in standby mode

The output signal of an analog filter type A can always be used as a standby mode wake-up signal.

(2) Analog filter type B

A block diagram of analog filter type B is shown below.

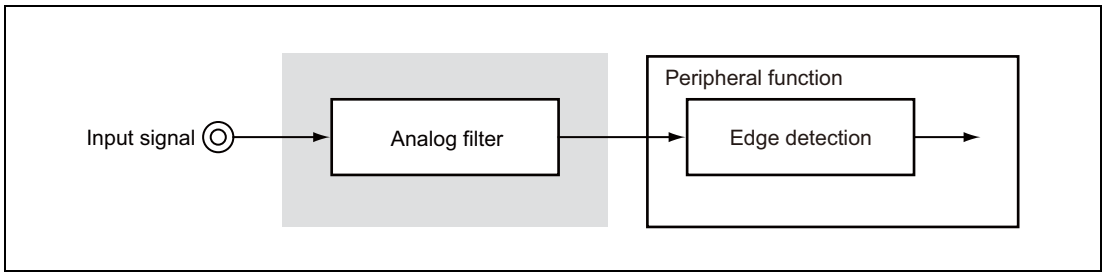


Figure 2.11 Block Diagram of Analog Filter Type B

Analog filter type B in standby mode

The output signal of an analog filter type B can always be used as a standby mode wake-up signal.

(3) Analog filter type C

A block diagram of analog filter type C is shown below.

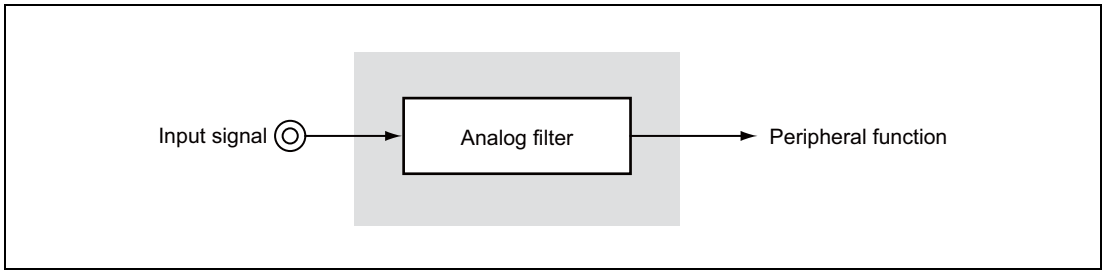


Figure 2.12 Block Diagram of Analog Filter Type C

The generated signals are always input signals that have passed through an analog filter.

Analog filter type C in standby mode

Pins equipped with type C analog filters in this product do not support the input of event signals to trigger wake-up from standby.

2.13.3 Digital Filters

2.13.3.1 Digital Filter Characteristic

The digital filters allow the filter characteristics to be adjusted according to the needs of the application.

The input signal is sampled with the sampling frequency f_s.

If a specified number of successive samples yield the same (high or low) level, the signal level is judged as valid and the filter output signal is set accordingly.

If an external signal level change is detected within the specified number of samples (same level samples), the signal level is judged as noise and the filter output signal does not change.

The length of an external signal pulse to be judged as noise depends on the sampling frequency and the specified number of same level samples.

Both parameters can be specified:

- DNFA<name>CTL.DNFA<name>PRS[2:0] select the sampling frequency based on
  $f_s = f_{DNFATCKI} / 2^{DNFA<name>PRS[2:0]}$
  where $f_{DNFATCKI}$ is the frequency of the DNFATCKI clock.
- DNFA<name>CTL.DNFA<name>NFSTS[1:0] determines the number of same level samples (2 to 5):

  s = DNFA<name>NFSTS[1:0] + 2

External signal pulse shorter than the following are suppressed at all times.

s × 1/fs

External signal pulses longer than the following are always judged as valid and are passed on to the filter output.

(s + 1) × 1/fs

External signal pulses in the following range may be suppressed or judged as valid.

s × 1/fs to (s + 1) × 1/fs

The filter operation is illustrated in the figure below with DNFA<name>NFSTS[1:0] = 01_B, i.e. s = 3 same level samples.

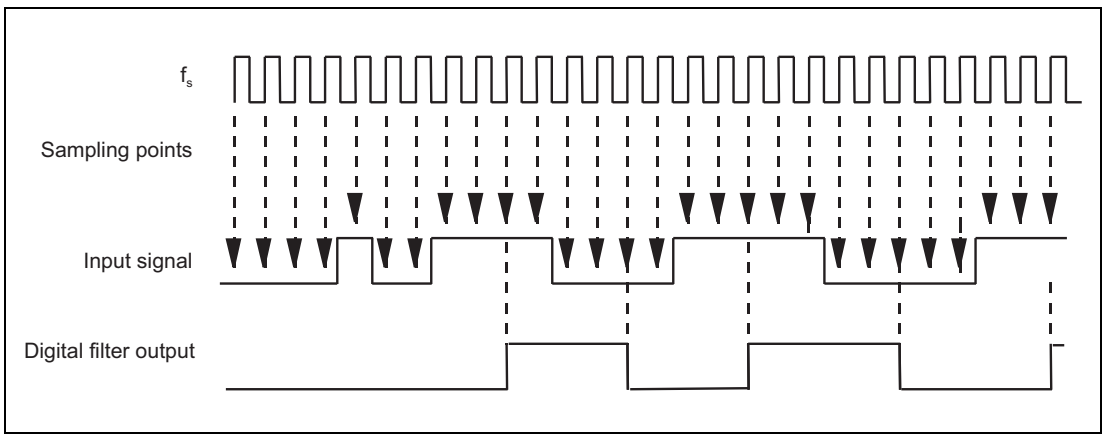


Figure 2.13 Digital Filter Function

2.13.3.2 Digital Filter Groups

The input signals equipped with digital filters are ordered in groups of up to 16 signals.

The digital filter characteristics, specified by DNFA<name>CTL.DNFA<name>PRS[2:0] and DNFA<name>NFSTS[1:0] apply to the signals group.

However, the digital filter for each signal can be enabled or disabled separately by DNFA<name>EN.DNFA<name>ENLm (m = 0 to 7) and DNFA<name>EN.DNFA<name>ENHm (m = 0 to 7).

CAUTIONS

1. **When the output signal from the digital filter is input to an alternative function, allow at least the following interval to elapse after the digital filter is enabled (DNFA<name>EN.DNFA<name>ENLm (m = 0 to 7) = 1 and DNFA<name>EN.DNFA<name>ENHm (m = 0 to 7) = 1) for the port pin to switch to the alternative function.**
   s = DNFA<name>NFSTS[1:0] + 2
   s × 1/fs + 2 × 1/fDNFATCKI
2. **When a digital filter's output signal is used as an interrupt signal, only enable the digital filter (DNFA<name>EN.DNFA<name>ENLm (m = 0 to 7) = 1 and DNFA<name>EN.DNFA<name>ENHm (m = 0 to 7) = 1) while interrupts are disabled. Furthermore, only enable interrupts after enabling the digital filter, waiting for the time below to elapse, and then clearing the interrupt request flag.**
   s × 1/fs + 3 × 1/fDNFATCKI

2.13.3.3 Digital Filter in Standby Mode

Digital filters for the wake-up capability from DeepSTOP mode are located in the Always-On-Area (AWO). Digital filters on the Always-On-Area (AWO) are always operating.

Digital noise elimination requires the clock supply DNFATCKI to operate. Pins equipped with digital filters in this product do not support the input of event signals to trigger wake-up from standby.

2.13.3.4 Digital Filter Control Registers

For each group consisting of up to 16 digital filters, the digital noise elimination control register DNFA<name>CTL and digital noise elimination enable register DNFA<name>EN are used to set all the filters in the same group (<name> = peripheral function group).

The DNFA<name>CTL register specifies the characteristics of the digital noise elimination filter for the digital filter of <name>.

The DNFA<name>EN register enables/disables each filter by setting the corresponding bit in DNFA<name>EN.DNFA<name>ENLm (m = 0 to 7) and DNFA<name>EN.DNFA<name>ENHm (m = 0 to 7).

The edge detection setup is done via the filter dedicated control register and the registers for individual peripheral functions.

The FCLA0CTL0_ADCn registers are ordered in groups of 3 registers with the same index n. The register index n is 0 or 1.

The assignment of the input signals to the control registers and their addresses are given in **Table 2.80, Input Pins that Incorporate Digital Filter Type D** and **Table 2.81, Input Pins that Incorporate Digital Filter Type E** in **Section 2.12.1, Port Filter Assignment**.

CAUTION

Do not change any control register settings while the corresponding digital filter is enabled by DNFA<name>EN.DNFA<name>ENLm (m = 0 to 7) = 1 and DNFA<name>EN.DNFA<name>ENHm (m = 0 to 7) = 1. Otherwise an unintended filter output may be generated.

(1) Digital filter type D

A block diagram of digital filter type D is shown below.

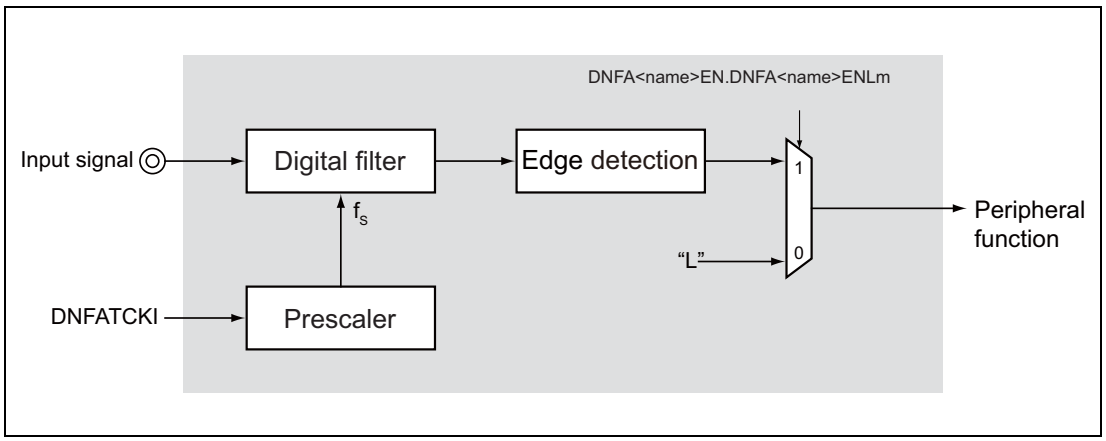


Figure 2.14 Block Diagram of Digital Filter Type D

The generated signal depends on the register setting, as shown in the following table.

Table 2.84 Output Options for Digital Filter Type D

DNFA<name>EN.DNFA<name>ENLm	Signals Output to Peripheral Functions
0	Fixed to low level
1	Input signal passed through filter

(2) Digital filter type E

A block diagram of digital filter type E is shown below.

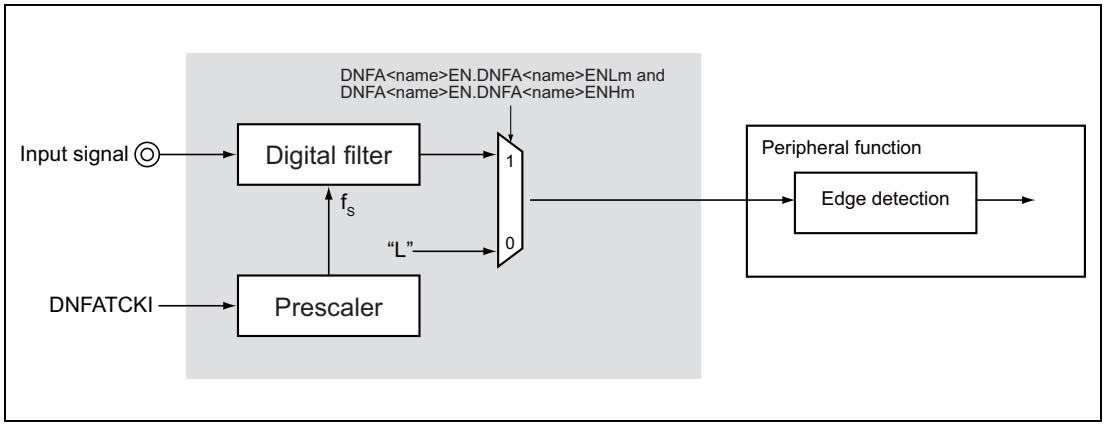


Figure 2.15 Block Diagram of Digital Filter Type E

The generated signal depends on the register setting, as shown in the following table.

Table 2.85 Output Options for Digital Filter Type E

DNFA<name>EN.DNFA<name>ENLm and DNFA<name>EN.DNFA<name>ENHm	Signals Output to Peripheral Functions
0	Fixed to low level
1	Input signal passed through filter

2.13.4 Filter Control Registers

The analog and digital filters are controlled and operated by the following registers:

Table 2.86 List of Filter Registers

Register Name	Symbol	Address
Filter control register m	FCLA0CTLm_<name>	The addresses are shown in the tables in **Section 2.12.1, Port Filter Assignment**.
Digital noise elimination control register	DNFA<name>CTL	
Digital noise elimination enable register	DNFA<name>EN	

2.13.4.1 FCLA0CTLm_<name> — Filter Control Register

Access: This register can be read or written in 8-bit units.

Address: The allocation of input signals to FCLA0CTLm_<name> registers and the address of each register are shown in the tables in **Section 2.12.1, Port Filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	FCLA0INTLm _<name>	FCLA0INTFm _<name>	FCLA0INTRm _<name>
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 2.87 FCLA0CTLm_<name> Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	FCLA0INTLm _<name>	Detection Mode Selection 0: Edge detection 1: Level detection **Note:** This bit is only valid for analog filter type A.
1	FCLA0INTFm _<name>	• In level detection mode (FCLA0INTLm_<name> = 1): This bit has no effect. • In edge detection mode (FCLA0INTLm_<name> = 0): Falling edge detection control 0: Falling edge detection disabled 1: Falling edge detection enabled **Note:** This bit is only valid for analog filter type A and digital filter type D. However, digital filter type D is placed in edge detection mode.
0	FCLA0INTRm _<name>	• In level detection mode (FCLA0INTLm_<name> = 1): Detected level selection 0: Low level detection 1: High level detection • In edge detection mode (FCLA0INTLm_<name> = 0): Rising edge detection control 0: Rising edge detection disabled 1: Rising edge detection enabled **Note:** This bit is only valid for analog filter type A and digital filter type D. However, digital filter type D is placed in edge detection mode.

CAUTION

Digital filter type D: When writing, always set bit2 to "0".

2.13.4.2 DNFA<name>CTL — Digital Noise Elimination Control Register

This register is used to specify the filter characteristics of the digital noise elimination filter.

NOTE

This register is only valid for digital filter type D and digital filter type E.

Access: This register can be read or written in 8-bit units.

Address: For the correspondence between the DNFA<name>CTL register and input signals, and the addresses of individual registers, see **Table 2.80, Input Pins that Incorporate Digital Filter Type D** and **Table 2.81, Input Pins that Incorporate Digital Filter Type E** in **Section 2.12.1, Port Filter Assignment**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	DNFA<name>NFSTS[1:0]		—	—	DNFA<name>PRS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

Table 2.88 DNFA<name>CTL Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6, 5	DNFA<name> NFSTS[1:0]	The DNFA<name>NFSTS[1:0] bits specify the number of samples used to judge whether an external signal pulse is valid.

DNFA<name>NFSTS[1:0]	Number of Samples
00_B	2
01_B	3
10_B	4
11_B	5

4, 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	DNFA<name> PRS[2:0]	Digital filter sampling clock selection

DNFA<name>PRS[2:0]	Sampling Clock Frequency
000_B	DNFATCKI/1
001_B	DNFATCKI/2
010_B	DNFATCKI/4
011_B	DNFATCKI/8
100_B	DNFATCKI/16
101_B	DNFATCKI/32
110_B	DNFATCKI/64
111_B	DNFATCKI/128

2.13.4.3 DNFA<name>EN — Digital Noise Elimination Enable Register

This register enables and disables digital noise elimination for a specified input signal.

NOTE

This register is only valid for digital filter type D and digital filter type E.

Access: This register can be read or written in 16-bit units.
The higher- and lower-order bytes (DNFA<name>ENH[7:0] and DNFA<name>ENL[7:0]) are accessible in 8- or 1-bit units respectively by setting DNFA<name>ENH and DNFA<name>ENL.

Address: For the correspondence between the DNFA<name>EN register and input signals, and the addresses of individual registers, see **Section 2.12.1, Port Filter Assignment**: **Table 2.81, Input Pins that Incorporate Digital Filter Type E** and **Table 2.80, Input Pins that Incorporate Digital Filter Type D**.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DNFA<name>ENH7	DNFA<name>ENH6	DNFA<name>ENH5	DNFA<name>ENH4	DNFA<name>ENH3	DNFA<name>ENH2	DNFA<name>ENH1	DNFA<name>ENH0	DNFA<name>ENL7	DNFA<name>ENL6	DNFA<name>ENL5	DNFA<name>ENL4	DNFA<name>ENL3	DNFA<name>ENL2	DNFA<name>ENL1	DNFA<name>ENL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.89 DNFA<name>EN Register Contents

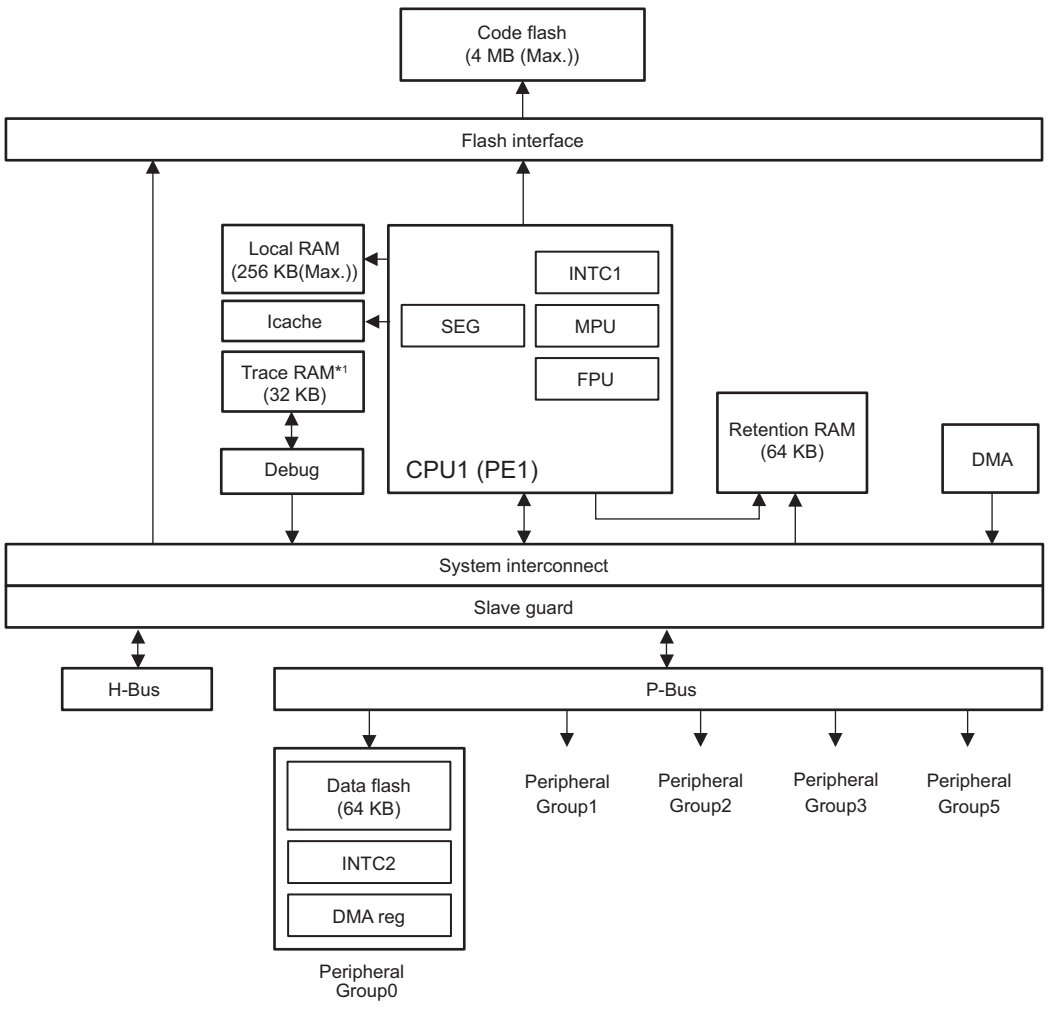
Bit Position	Bit Name	Function
15 to 0	DNFA<name>ENH[7:0] DNFA<name>ENL[7:0]	Digital Noise Elimination Enable/Disable Control 0: Digital noise elimination disabled 1: Digital noise elimination enabled

Section 3 CPU System

3.1 Overview

3.1.1 Block Configuration

Figure 3.1 shows the block configuration diagram of RH850/F1M.



Note 1. Available only in devices with 4-MB code flash memory..

Figure 3.1 Block Configuration Diagram of RH850/F1M

CPU1 (PE1)

The RH850 G3M Core is used as the main CPU.

Local RAM

This is a high-speed accessible RAM.

Retention RAM

The retention RAM is used to retain values in DeepSTOP mode. Since the retention RAM uses the global RAM area, the retention RAM can also serve as a global RAM for sharing data with the DMA.

Code Flash

The code flash memory is included for program storage. It is connected with CPU1 via the flash interface.

Data Flash

The data flash memory can be rewritten by the CPUs. It has a greater write endurance than the code flash memory.

P-Bus and H-Bus

These buses connect the peripheral IPs. The P-Bus is divided into five peripheral groups, 0 to 3, and 5.

INTC1, INTC2

There are two interrupt controllers, INTC1 and INTC2.

DMA

The DMA transfer module (DMAC) is included.

Slave Guard

The slave guard is a function to prevent unauthorized access from a specific bus master, and consists of the following guard structures:

(1) PE Guard (PEG)

The PE guard is a function to prevent unauthorized access to the resources in the PE from an external master. After reset is released, access from other than the own PE is prohibited.

(2) Global RAM Guard (GRG)

The global RAM guard is a function to prevent unauthorized access to the retention RAM from the external master. The retention RAM is in the unprotected state (accessible from all bus masters) after reset is released.

See **Section 33, Functional Safety**.

3.2 CPU

3.2.1 Core Functions

3.2.1.1 Features

Table 3.1 lists features of the RH850 G3M core.

Table 3.1 Features of the RH850 G3M Core

Item	Feature
CPU	• Advanced 32-bit architecture for embedded control • 32-bit internal data bus • Thirty-two 32-bit general-purpose registers • RISC-type instruction set – Long-/short-format load/store instructions – Three-operand instructions – Instruction set based on C language • CPU operating modes – User mode and supervisor mode • Address space: 4-Gbyte linear address space for both data and instructions • Instructions: A snooze instruction (SNOOZE) is included for temporary suspension by switching the CPU clock signal off for 32 clock cycles.
Coprocessor	• Floating-point operation coprocessor (FPU) – Supports single precision (32 bits) and double precision (64 bits). – Supports data types and exceptions conforming to IEEE754. – Rounding mode: Neighborhood, 0 direction, +∞ direction, and –∞ direction – Handling of denormalized numbers: Rounding down to 0 or exception notification to conform to IEEE754
Exception/ Interrupt	• 16 interrupt priority levels settable for each channel • Vector selection method selectable according to performance request or memory usage – Direct branching exception vectors – Indirect branching exception vectors referring to the address table • Supports the high-speed save/return processing of the context by the dedicated instructions (PUSHSP and POPSP) at the generation of an interrupt
Memory Management	• Memory protection function (MPU): 16 areas settable
Cache	• Instruction cache

3.2.1.2 Register Set

This subsection explains the program registers and system registers incorporated in this CPU.

(1) Program Registers

Program registers include the general-purpose registers (r0 to r31) and program counter (PC). The general-purpose register r0 always retains 0, but the values of the registers r1 to r31 after reset are undefined.

Table 3.2 List of Program Registers

Program Register	Name	Function	Description
General-purpose registers	r0	Zero register	Always retains “0”
	r1	Assembler reserved register	Used as working register for generating addresses
	r2	Register for address and data variables (used when the real-time OS used does not use this register)	
	r3	Stack pointer (SP)	Used for generating a stack frame when a function is called
	r4	Global pointer (GP)	Used for accessing a global variable in the data area
	r5	Text pointer (TP)	Used as a register that indicates the start of the text area (area where program code is placed)
	r6 to r29	Register for addresses and data variables	
	r30	Element pointer (EP)	Used as a base pointer for generating addresses when accessing memory
	r31	Link pointer (LP)	Used when the compiler calls a function
Program counter	PC	Retains instruction addresses during execution of programs	

NOTE

For further descriptions of r1, r3 to r5, and r31 used by the assembler and/or C compiler, see the specification of each software development environment.

(a) General-purpose registers

A total of 32 general-purpose registers (r0 to r31) are provided. All of these registers can be used for either data variables or address variables. Of the general-purpose registers, r0 to r5, r30, and r31 are assumed to be used for special purposes in software development environments, so it is necessary to note the following when using them.

1. r0, r3, r30

   These registers are implicitly used by instructions.

   r0 is a register that always retains “0”. It is used for operations that use 0 and addressing with base address being 0.

   r3 is implicitly used by the PREPARE, DISPOSE, PUSHSP, and POPSP instructions.

   r30 is used as a base pointer when the SLD or SST instruction accesses memory.

2. r1, r4, r5, r31

   These registers are implicitly used by the assembler and C compiler.

   When using these registers, register contents must first be saved so they are not lost and must be restored after the registers are used.

3. r2

   This register might be used by a real-time OS in some cases. If the real-time OS that is being used is not using r2, r2 can be used as a register for address variables or data variables.

(b) PC – Program counter

The PC retains the address of the instruction being executed. Bit 0 is fixed to 0, and branching to an odd number address is disabled.

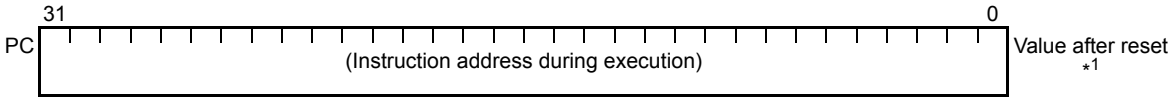


Table 3.3 PC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	PC31 to 1	These bits indicate the address of the instruction being executed.	R/W	*1
0	PC0	This bit is fixed to 0, and branching to an odd-number address is disabled.	R/W	0

Note 1. The value after reset differs depending on the setting value of the reset vector. For details, see **(q) RBASE — Reset vector base address register**.

(2) Basic System Registers

The basic system registers are used to control CPU status and to retain exception information.

System registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and a selection ID.

Table 3.4 Basic System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR0, 0	EIPC	Status save registers when acknowledging EI level exception	SV
SR1, 0	EIPSW	Status save registers when acknowledging EI level exception	SV
SR2, 0	FEPC	Status save registers when acknowledging FE level exception	SV
SR3, 0	FEPSW	Status save registers when acknowledging FE level exception	SV
SR5, 0	PSW	Program status word	*1
SR6, 0	FPSR	(Refer to FPU function registers.)	CU0 and SV
SR7, 0	FPEPC	(Refer to FPU function registers.)	CU0 and SV
SR8, 0	FPST	(Refer to FPU function registers.)	CU0
SR9, 0	FPCC	(Refer to FPU function registers.)	CU0
SR10, 0	FPCFG	(Refer to FPU function registers.)	CU0
SR11, 0	FPEC	(Refer to FPU function registers.)	CU0 and SV
SR13, 0	EIIC	EI level exception cause	SV
SR14, 0	FEIC	FE level exception cause	SV
SR16, 0	CTPC	CALLT execution status save register	UM
SR17, 0	CTPSW	CALLT execution status save register	UM
SR20, 0	CTBP	CALLT base pointer	UM
SR28, 0	EIWR	EI level exception working register	SV
SR29, 0	FEWR	FE level exception working register	SV
SR31, 0	(BSEL)	(Reserved for backwards compatibility with the V850E2 series)*2	SV
SR0, 1	MCFG0	Machine configuration	SV
SR2, 1	RBASE	Reset vector base address	SV
SR3, 1	EBASE	Exception handler vector address	SV
SR4, 1	INTBP	Base address of the interrupt handler table	SV
SR5, 1	MCTL	CPU control	SV
SR6, 1	PID	Processor ID	SV
SR11, 1	SCCFG	SYSCALL operation setting	SV
SR12, 1	SCBP	SYSCALL base pointer	SV
SR0, 2	HTCFG0	Thread configuration	SV
SR6, 2	MEA	Memory error address	SV
SR7, 2	ASID	Address space ID	SV
SR8, 2	MEI	Memory error information	SV

Note 1. The access permission differs depending on the bit. See the description of **(e) PSW — Program status word** in (2) Basic System Register.

Note 2. Reserved for backwards compatibility with the V850E2 Series. These registers are always read as 0. Writing to these registers are ignored.

(a) EIPC — Status save register when acknowledging EI level exception

When an EI level exception is acknowledged, the address of the instruction that was being executed when the EI level exception occurred, or of the next instruction, is saved to the EIPC register (see *Section 4.1.3, Types of exceptions*, in the *RH850G3M User's Manual: Software*).

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the EIPC register. An odd-numbered address cannot be specified.

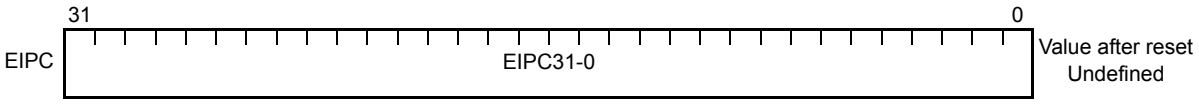


Table 3.5 EIPC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	EIPC31-1	These bits indicate the return PC when an EI level exception is acknowledged.	R/W	Undefined
0	EIPC0	This bit indicates the return PC when an EI level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the EIRET instruction is executed is 0.	R/W	Undefined

(b) EIPSW — Status save register when acknowledging EI level exception

When an EI level exception is acknowledged, the current PSW setting is saved to the EIPSW register.

Because there is only one pair of EI level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

CAUTION

Bits 11 to 9 cannot usually be changed since these bits are associated with debugging.

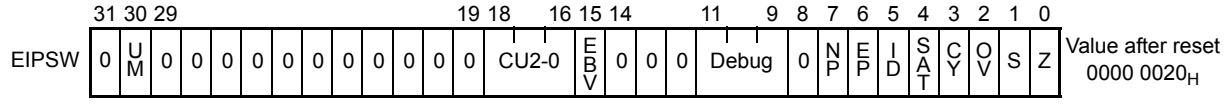


Table 3.6 EIPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
30	UM	This bit saves the PSW.UM bit setting when an EI level exception is acknowledged.	R/W	0
29 to 19	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
18, 17	CU2, 1	(Reserved for future expansion. Be sure to clear to 0.)	R	0
16	CU0	This bit saves the PSW.CU0 field setting when an EI level exception is acknowledged.	R/W	0
15	EBV	This bit saves the PSW.EBV bit setting when an EI level exception is acknowledged.	R/W	0
14 to 12	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
11 to 9	Debug	These bits store the PSW.Debug field when an EI level exception is acknowledged.	R/W	0
8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7	NP	This bit saves the PSW.NP bit setting when an EI level exception is acknowledged.	R/W	0
6	EP	This bit saves the PSW.EP bit setting when an EI level exception is acknowledged.	R/W	0
5	ID	This bit saves the PSW.ID bit setting when an EI level exception is acknowledged.	R/W	1
4	SAT	This bit saves the PSW.SAT bit setting when an EI level exception is acknowledged.	R/W	0
3	CY	This bit saves the PSW.CY bit setting when an EI level exception is acknowledged.	R/W	0
2	OV	This bit saves the PSW.OV bit setting when an EI level exception is acknowledged.	R/W	0
1	S	This bit saves the PSW.S bit setting when an EI level exception is acknowledged.	R/W	0
0	Z	This bit saves the PSW.Z bit setting when an EI level exception is acknowledged.	R/W	0

(c) FEPC — Status save register when acknowledging FE level exception

When an FE level exception is acknowledged, the address of the instruction that was being executed when the FE level exception occurred, or of the next instruction, is saved to the FEPC register (see *Section 4.1.3, Types of exceptions*, in *the RH850G3M User's Manual: Software*).

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

Be sure to set an even-numbered address to the FEPC register. An odd-numbered address cannot be specified.

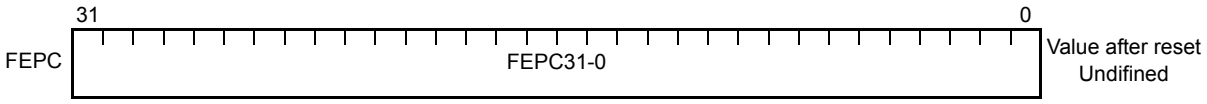


Table 3.7 FEPC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	FEPC31-1	These bits indicate the return PC when an FE level exception is acknowledged.	R/W	Undefined
0	FEPC0	This bit indicates the return PC when an FE level exception is acknowledged. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the FERET instruction is executed is 0.	R/W	Undefined

(d) FEPSW — Status save register when acknowledging FE level exception

When an FE level exception is acknowledged, the current PSW setting is saved to the FEPSW register.

Because there is only one pair of FE level exception status save registers, when processing multiple exceptions, the contents of these registers must be saved by a program.

CAUTION

Bits 11 to 9 cannot usually be changed since these bits are associated with debugging.

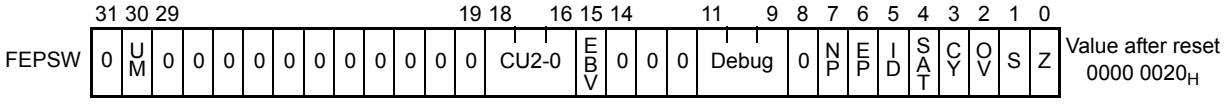


Table 3.8 FEPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
30	UM	This bit saves the PSW.UM bit setting when an FE level exception is acknowledged.	R/W	0
29 to 19	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
18, 17	CU2, 1	(Reserved for future expansion. Be sure to clear to 0.)	R	0
16	CU0	This bit saves the PSW.CU0 field setting when an FE level exception is acknowledged.	R/W	0
15	EBV	This bit saves the PSW.EBV bit setting when an FE level exception is acknowledged.	R/W	0
14 to 12	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
11 to 9	Debug	These bits are associated with debugging.	R/W	0
8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7	NP	This bit saves the PSW.NP bit setting when an FE level exception is acknowledged.	R/W	0
6	EP	This bit saves the PSW.EP bit setting when an FE level exception is acknowledged.	R/W	0
5	ID	This bit saves the PSW.ID bit setting when an FE level exception is acknowledged.	R/W	1
4	SAT	This bit saves the PSW.SAT bit setting when an FE level exception is acknowledged.	R/W	0
3	CY	This bit saves the PSW.CY bit setting when an FE level exception is acknowledged.	R/W	0
2	OV	This bit saves the PSW.OV bit setting when an FE level exception is acknowledged.	R/W	0
1	S	This bit saves the PSW.S bit setting when an FE level exception is acknowledged.	R/W	0
0	Z	This bit saves the PSW.Z bit setting when an FE level exception is acknowledged.	R/W	0

(e) PSW — Program status word

PSW (program status word) is a set of flags that indicate the program status (instruction execution result) and bits that indicate the operation status of the CPU (flags are bits in the PSW that are referenced by conditional instructions (Bcond, CMOV, etc.)).

CAUTIONS

1. **When the LDSR instruction is used to change the contents of bits 7 to 0 in this register, the changed contents become valid immediately after the completion of execution of the LDSR instruction.**
2. **The access permission for the PSW register differs with each bit. All bits can be read, but some bits can only be written under certain conditions. See Table 3.9, Access Permission for PSW Register for the access permission for each bit.**

Table 3.9 Access Permission for PSW Register

Bit		Access Permission When Reading	Access Permission When Writing
30	UM	UM	SV*1
16	CU0	UM	SV*1
15	EBV	UM	SV*1
11 to 9	Debug	UM	Special*1
7	NP	UM	SV*1
6	EP	UM	SV*1
5	ID	UM	SV*1
4	SAT	UM	UM
3	CY	UM	UM
2	OV	UM	UM
1	S	UM	UM
0	Z	UM	UM

Note 1. The access permission for the whole PSW register is UM, so the PIE exception does not occur even if the register is written by using an LDSR instruction when PSW.UM is 1. In this case, writing is ignored.

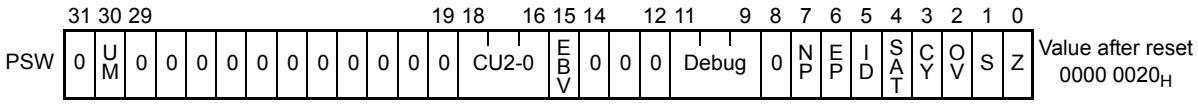


Table 3.10 PSW Register Contents (1/2)

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
30	UM	This bit indicates that the CPU is in user mode (UM mode). 0: Supervisor mode 1: User mode	R/W	0
29 to 19	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
18, 17	CU2, 1	(Reserved for future expansion. Be sure to clear to 0.)	R	0
16	CU0	This bit indicates the coprocessor use permissions. When the bit corresponding to the coprocessor is 0, a coprocessor use prohibition exception is generated in response to execution of a coprocessor instruction or access to coprocessor resources (system registers). CU0 bit 16: FPU	R/W	0

Table 3.10 PSW Register Contents (2/2)

Bit Position	Bit Name	Function	R/W	Value after Reset
15	EBV	This bit indicates the reset vector and exception vector operation. See the descriptions of the RBASE register in **(q) RBASE — Reset vector base address register** and the EBASE register in **(r) EBASE — Exception handler vector address register**.	R/W	0
14 to 12	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
11 to 9	Debug	These bits are used in debugging for development tools. In normal cases, set these bits to 0.	R/W	0
8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7	NP	This bit indicates that an FE-level exception is being processed. When an FE level exception is acknowledged, this bit is set to 1, which prohibits occurrence of multiple exceptions*2. For the exceptions disabled by the NP bit, see *Table 4.1, Exception Cause List*, in *the RH850G3M User's Manual: Software*. 0: FE level exception handling is not in progress. 1: FE level exception handling is in progress.	R/W	0
6	EP	This bit indicates that an exception other than an interrupt is being serviced. It is set to 1 when the corresponding exception occurs. This bit does not affect acknowledging an exception request even when it is set to 1. 0: An exception other than an interrupt is not being serviced. 1: An exception other than an interrupt is being serviced.	R/W	0
5	ID	This bit indicates that an EI-level exception is being processed. It is set to 1 when an EI level exception is acknowledged, disabling generation of multiple exceptions*2. For the exceptions disabled by the ID bit, see *Table 4.1, Exception Cause List*, in *the RH850G3M User's Manual: Software*. This bit is also used to disable EI level exceptions from being acknowledged in a critical section during ordinary program execution or interrupt servicing. It is set to 1 when the DI instruction is executed, and cleared to 0 when the EI instruction is executed. Changes to the value of the ID bit by the EI or DI instruction will be effective from the next instruction. 0: EI level exception is not being processed or the section is not a critical section (after execution of EI instruction). 1: EI level exception is being processed or the section is a critical section (after execution of DI instruction).	R/W	1
4	SAT*1	This bit indicates that the operation result is saturated because the result of a saturated operation instruction operation has overflowed. This is a cumulative flag, so when the operation result of the saturated operation instruction becomes saturated, this bit is set to 1, but it is not cleared to 0 when the operation result for a subsequent instruction is not saturated. This bit is cleared to 0 by the LDSR instruction. This bit is neither set to 1 nor cleared to 0 when an arithmetic operation instruction is executed. 0: Not saturated 1: Saturated	R/W	0
3	CY	This bit indicates whether a carry or borrow has occurred in the operation result. 0: Carry and borrow have not occurred. 1: Carry or borrow has occurred.	R/W	0
2	OV*1	This bit indicates whether or not an overflow has occurred during an operation. 0: Overflow has not occurred. 1: Overflow has occurred.	R/W	0
1	S*1	This bit indicates whether or not the result of an operation is negative. 0: Result of operation is positive or 0. 1: Result of operation is negative	R/W	0
0	Z	This bit indicates whether or not the result of an operation is 0. 0: Result of operation is not 0. 1: Result of operation is 0.	R/W	0

Note 1. The operation result of the saturation processing is determined in accordance with the contents of the OV flag and S flag during a saturated operation. The SAT flag is set to 1 only when the OV flag is set to 1 in a saturated operation.

Note 2. Exceptions from certain sources are accepted regardless of the states of the ID and NP bits. For details, see **Section 3.3.4, Overwriting Context upon Acceptance of Multiple Exceptions**.

Operation Result Status	Flag Status			Operation Result after Saturation Processing
	SAT	OV	S	
Exceeded positive maximum value	1	1	0	7FFF FFFF$_H$
Exceeded negative maximum value	1	1	1	8000 0000$_H$
Positive (maximum value not exceeded)	Value prior to operation is retained.	0	0	Operation result itself
Negative (maximum value not exceeded)			1	

(f) EIIC — EI level exception source register

The EIIC register retains the source of any EI level exception that occurs. The value retained in this register is an exception source code corresponding to a specific exception source.

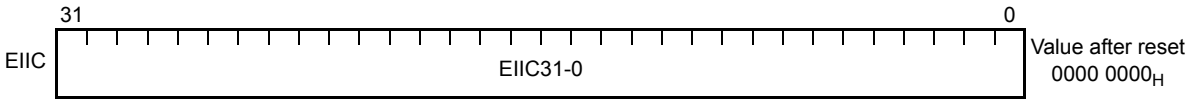


Table 3.11 EIIC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	EIIC31-0	These bits store the exception source code when an EI level exception is acknowledged. Regarding the stored exception source code, see **Table 7.4, EI Level Maskable Interrupt Sources**, and *Table 4.1, Exception Cause List*, in *the RH850G3M User's Manual: Software*. The EIIC31-16 field stores detailed exception source codes defined individually for each exception. If there is no particular definition of a function related to the exception these bits are set to 0.	R/W	0

(g) FEIC — FE level exception source register

The FEIC register retains the source of any FE level exception that occurs. The value retained in this register is an exception code corresponding to a specific exception source.

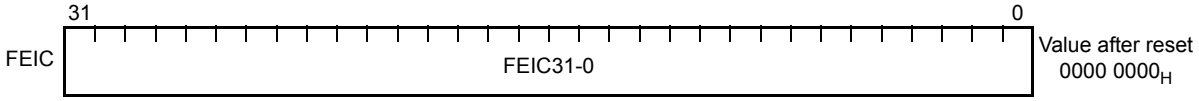


Table 3.12 FEIC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	FEIC31-0	These bits store the exception source code when an FE level exception is acknowledged. Regarding the stored exception source code, see **Table 7.2, FE Level Non-Maskable Interrupt Requests**, **Table 7.3, FE Level Maskable Interrupt Requests**, and *Table 4.1, Exception Cause List*, in *the RH850G3M User's Manual: Software*. The FEIC31-16 field stores detailed exception source codes defined individually for each exception. If there is no particular definition of a function related to the exception these bits are set to 0.	R/W	0

(h) CTPC — Status save register when executing CALLT instruction

When a CALLT instruction is executed, the address of the next instruction after the CALLT instruction is saved to CTPC. Be sure to set an even-numbered address to the CTPC register. An odd-numbered address cannot be specified.

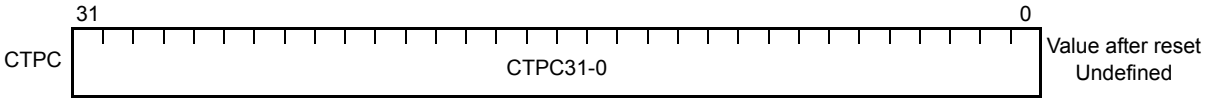


Table 3.13 CTPC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	CTPC31-1	These bits indicate the PC of the instruction after the CALLT instruction.	R/W	Undefined
0	CTPC0	This bit indicates the PC of the instruction after the CALLT instruction. Always set this bit to 0. Even if it is set to 1, the value transferred to the PC when the CTRET instruction is executed is 0.	R/W	Undefined

(i) CTPSW — Status save register when executing CALLT instruction

When a CALLT instruction is executed, some of the PSW (program status word) settings are saved to CTPSW.

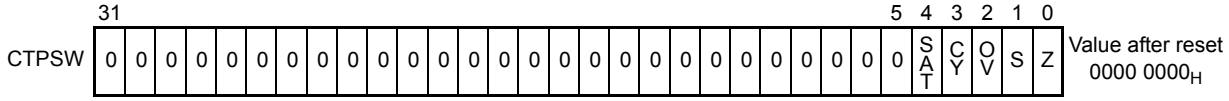


Table 3.14 CTPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
4	SAT	This bit saves the PSW.SAT bit setting when the CALLT instruction is executed.	R/W	0
3	CY	This bit saves the PSW.CY bit setting when the CALLT instruction is executed.	R/W	0
2	OV	This bit saves the PSW.OV bit setting when the CALLT instruction is executed.	R/W	0
1	S	This bit saves the PSW.S bit setting when the CALLT instruction is executed.	R/W	0
0	Z	This bit saves the PSW.Z bit setting when the CALLT instruction is executed.	R/W	0

(j) CTBP — CALLT base pointer register

The CTBP register is used to specify table addresses of the CALLT instruction and generate target addresses.

Be sure to set the CTBP register to a halfword address.

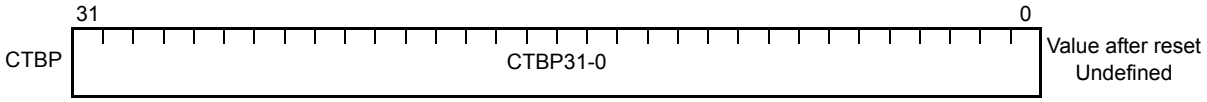


Table 3.15 CTBP Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	CTBP31-1	These bits indicate the base pointer address of the CALLT instruction. These bits indicate the first address in the table used by the CALLT instruction.	R/W	Undefined
0	CTBP0	This bit indicates the base pointer address of the CALLT instruction. This bit indicates the first address in the table used by the CALLT instruction. Always set this bit to 0.	R	0

(k) ASID — Address space ID register

This is the address space ID. This is used to identify the address space provided by the memory management function.

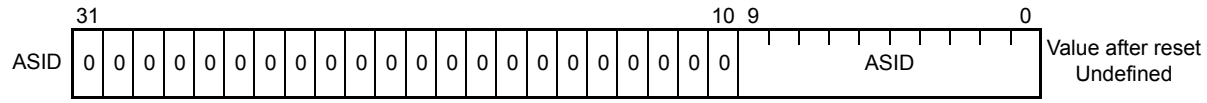


Table 3.16 ASID Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 10	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
9 to 0	ASID	This field is the address space ID.	R/W	Undefined

(l) EIWR — EI level exception working register

The EIWR register is used as a working register when an EI level exception has occurred.

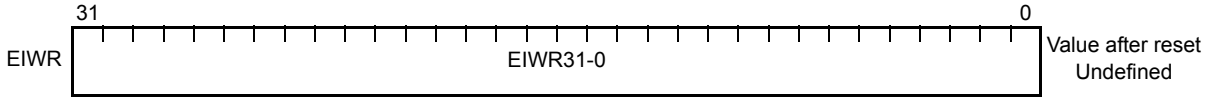


Table 3.17 EIWR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	EIWR31-0	These bits constitute a working register that can be used for any purpose during the processing of an EI level exception. This register can be used to temporarily save the values of general-purpose registers, etc.	R/W	Undefined

(m) FEWR — FE level exception working register

The FEWR register is used as a working register when an FE level exception has occurred.

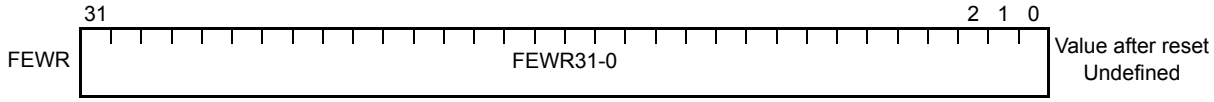


Table 3.18 FEWR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	FEWR31-0	These bits constitute a working register that can be used for any purpose during the processing of an FE level exception. This register can be used to temporarily save the values of general-purpose registers, etc.	R/W	Undefined

(n) HTCFG0 — Thread configuration register

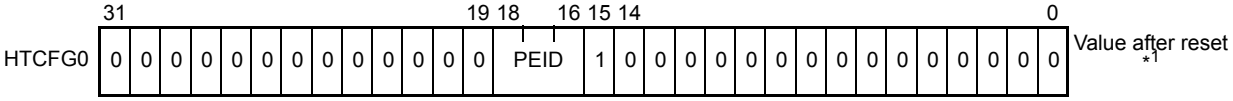


Table 3.19 HTCFG0 Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 19	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
18 to 16	PEID	These bits indicate the processor element number.	R	*2
15	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
14 to 0	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

Note 1. The value after reset is 0001 8000_H for CPU1 (PE1) of this product.

Note 2. The value after reset is 001_B for CPU1 (PE1) of this product.

(o) MEA — Memory error address register

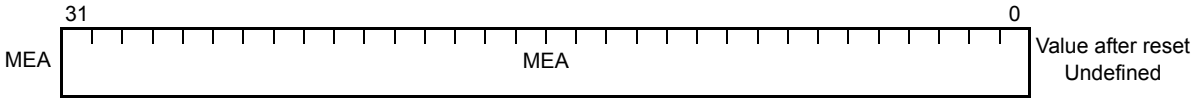


Table 3.20 MEA Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MEA	These bits store an address when a MAE (misaligned) or MPU violation occurs.	R/W	Undefined

(p) MEI — Memory error information register

This register is used to store information about the instruction that caused the exception when a misaligned (MAE) or memory protection (MDP) exception had occurred. This information is used during emulation.

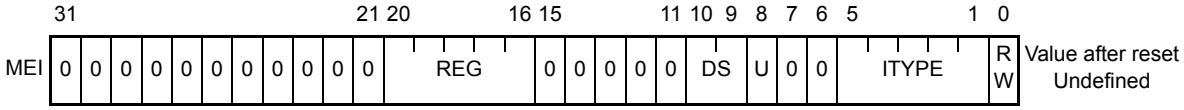


Table 3.21 MEI Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 21	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
20 to 16	REG4-0	These bits indicate the number of the source or destination register accessed by the instruction that caused the exception. For details, see **Table 3.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
15 to 11	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
10, 9	DS	These bits indicate the data type of the instruction that caused the exception.*1 0: Byte (8 bits) 1: Halfword (16 bits) 2: Word (32 bits) 3: Double-word (64 bits) For details, see **Table 3.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
8	U	These bits indicate the sign extension method of the instruction that caused the exception. 0: Signed 1: Unsigned For details, see **Table 3.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
5 to 1	ITYPE4-0	These bits indicate the instruction that caused the exception. For details, see **Table 3.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined
0	RW	This bit indicates whether the operation of the instruction that caused the exception was read (Load-memory) or write (Store-memory). 0: Read (Load-memory) 1: Write (Store-memory) For details, see **Table 3.22, Instructions Causing Exceptions and Values of MEI Register**.	R/W	Undefined

Note 1. Even if the data is divided and access is made several times due to the specifications of the hardware, the original data type indicated by the instruction is stored.

Table 3.22 Instructions Causing Exceptions and Values of MEI Register (1/2)

Instruction	REG	DS	U	RW	ITYPE
SLD.B	dst	0 (byte)	0 (signed)	0 (read)	00000_B
SLD.BU	dst	0 (byte)	1 (unsigned)	0 (read)	00000_B
SLD.H	dst	1 (halfword)	0 (signed)	0 (read)	00000_B
SLD.HU	dst	1 (halfword)	1 (unsigned)	0 (read)	00000_B
SLD.W	dst	2 (word)	0 (signed)	0 (read)	00000_B
SST.B	src	0 (byte)	0 (signed)	1 (write)	00000_B
SST.H	src	1 (halfword)	0 (signed)	1 (write)	00000_B
SST.W	src	2 (word)	0 (signed)	1 (write)	00000_B
LD.B (disp16)	dst	0 (byte)	0 (signed)	0 (read)	00001_B

Table 3.22 Instructions Causing Exceptions and Values of MEI Register (2/2)

Instruction	REG	DS	U	RW	ITYPE
LD.BU (disp16)	dst	0 (byte)	1 (unsigned)	0 (read)	00001_B
LD.H (disp16)	dst	1 (halfword)	0 (signed)	0 (read)	00001_B
LD.HU (disp16)	dst	1 (halfword)	1 (unsigned)	0 (read)	00001_B
LD.W (disp16)	dst	2 (word)	0 (signed)	0 (read)	00001_B
ST.B (disp16)	src	0 (byte)	0 (signed)	1 (write)	00001_B
ST.H (disp16)	src	1 (halfword)	0 (signed)	1 (write)	00001_B
ST.W (disp16)	src	2 (word)	0 (signed)	1 (write)	00001_B
LD.B (disp23)	dst	0 (byte)	0 (signed)	0 (read)	00010_B
LD.BU (disp23)	dst	0 (byte)	1 (unsigned)	0 (read)	00010_B
LD.H (disp23)	dst	1 (halfword)	0 (signed)	0 (read)	00010_B
LD.HU (disp23)	dst	1 (halfword)	1 (unsigned)	0 (read)	00010_B
LD.W (disp23)	dst	2 (word)	0 (signed)	0 (read)	00010_B
ST.B (disp23)	src	0 (byte)	0 (signed)	1 (write)	00010_B
ST.H (disp23)	src	1 (halfword)	0 (signed)	1 (write)	00010_B
ST.W (disp23)	src	2 (word)	0 (signed)	1 (write)	00010_B
LD.DW (disp23)	dst	3 (double-word)	0 (signed)	0 (read)	00010_B
ST.DW (disp23)	src	3 (double-word)	0 (signed)	1 (write)	00010_B
LDL.W	dst	2 (word)	0 (signed)	0 (read)	00111_B
STC.W	src	2 (word)	0 (signed)	1 (write)	00111_B
CAXI	dst	2 (word)	1 (unsigned)	0 (read)*1	01000_B
SET1	—	0 (byte)	1 (unsigned)	0 (read)*1	01001_B
CLR1	—	0 (byte)	1 (unsigned)	0 (read)*1	01001_B
NOT1	—	0 (byte)	1 (unsigned)	0 (read)*1	01001_B
TST1	—	0 (byte)	1 (unsigned)	0 (read)	01001_B
PREPARE	—	2 (word)	1 (unsigned)	1 (write)	01100_B
DISPOSE	—	2 (word)	1 (unsigned)	0 (read)	01100_B
PUSHSP	—	2 (word)	1 (unsigned)	1 (write)	01101_B
POPSP	—	2 (word)	1 (unsigned)	0 (read)	01101_B
SWITCH	—	1 (halfword)	0 (signed)	0 (read)	10000_B
CALLT	—	1 (halfword)	1 (unsigned)	0 (read)	10001_B
SYSCALL	—	2 (word)	1 (unsigned)	0 (read)	10010_B
CACHE	—	—	—	0/1*2	10100_B
Interrupt (table reference)*3	—	2 (word)	1 (unsigned)	0 (read)	10101_B

Note 1. This exception occurs when the instruction executes a read access.

Note 2. The value depends on actual operation.

Note 3. When reading the interrupt vector by using the table reference method.

NOTE

dst: destination register number, src: source register number

(q) RBASE — Reset vector base address register

This register indicates the reset vector address when there is a reset. If the PSW.EBV bit is 0, this vector address is also used as the exception vector address.

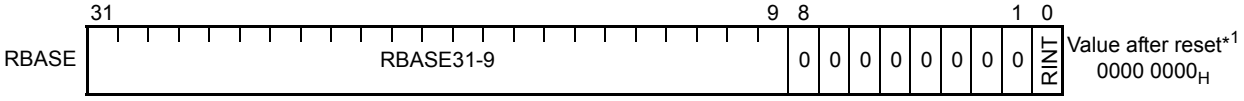


Table 3.23 RBASE Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	RBASE31-9	These bits indicate the reset vector when there is a reset. When PSW.EBV = 0, this address is also used as the exception vector. The RBASE8-0 bits are implicitly 0.	R	CPU1: 0000 0000 0000 0000 0000 000B*1
8 to 1	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
0	RINT	When the RINT bit is set, the exception handler address for interrupt processing is reduced. For details, see *Section 4.5.1 (1) Direct vector method* in *the RH850G3M User's Manual: Software*. This bit is valid when PSW.EBV = 0.	R	0

Note 1. The value depends on the reset vector. The values set at shipment are shown in the table. When the reset vector is modified, the address will be changed.

(r) EBASE — Exception handler vector address register

This register indicates the exception handler vector address. This register is valid when the PSW.EBV bit is 1.

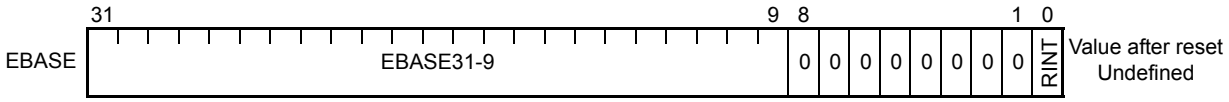


Table 3.24 EBASE Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	EBASE31-9	The exception handler routine address is changed to the address resulting from adding the offset address of each exception to the base address specified for this register. The EBASE8-0 bits are implicitly 0.	R/W	Undefined
8 to 1	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
0	RINT	When the RINT bit is set, the offset address of the exception handler used during interrupt servicing is decremented. For details, see *Section 4.5.1 (1) Direct vector method* in *the RH850G3M User's Manual: Software*.	R/W	Undefined

(s) INTBP — Base address register of the interrupt handler table

This register indicates the base address of the address table when the table reference method is selected as the interrupt handler address selection method.

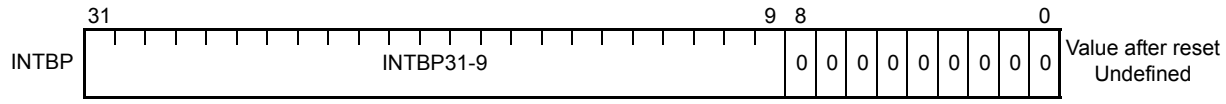


Table 3.25 INTBP Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	INTBP31-9	These bits indicate the base pointer address for an interrupt prescribed in the expanded specifications. The value indicated by these bits is the first address in the table used to determine the exception handler when the interrupts (EIINT0 to EIINT350) prescribed by the expanded specifications is acknowledged. The INTBP8-0 bits are implicitly 0.	R/W	Undefined
8 to 0	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

(t) PID — Processor ID register

The PID register retains a processor identifier that is unique to the CPU. The PID register is a read-only register.

CAUTION

The PID register indicates information used to identify the incorporated CPU core and CPU core configuration. Usage such that the software behavior varies dynamically according to the PID register information is not assumed.

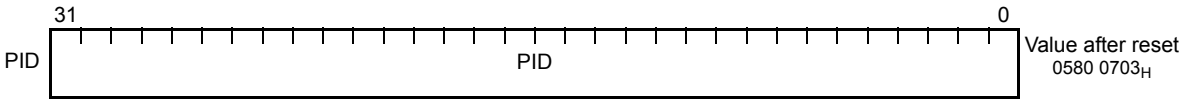


Table 3.26 PID Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 24	PID	Architecture Identifier This identifier indicates the architecture of the processor.	R	05_H
23 to 8		Function Identifier This identifier indicates the functions of the processor. These bits indicate whether or not functions defined per bit are implemented (1: implemented, 0: not implemented). Bits 23 to 11: Reserved Bit 10: Double-precision floating-point operation function Bit 9: Single-precision floating-point operation function Bit 8: Memory protection function (MPU)	R	8007_H
7 to 0		Version Identifier This identifier indicates the version of the processor.	R	03_H

(u) SCCFG — SYSCALL operation setting register

This register is used to specify operations related to the SYSCALL instruction. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

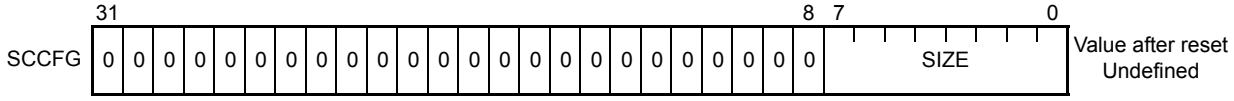


Table 3.27 SCCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7 to 0	SIZE	These bits specify the maximum number of entries of a table that the SYSCALL instruction references. The maximum number of entries the SYSCALL instruction references is 1 if SIZE is 0, and 256 if SIZE is 255. By setting the maximum number of entries appropriately in accordance with the number of functions branched by the SYSCALL instruction, the memory area can be effectively used. If vectors exceeding the maximum number of entries are specified by the SYSCALL instruction, the first entry is selected. Place an error processing routine at the first entry.	R/W	Undefined

(v) SCBP — SYSCALL base pointer register

The SCBP register is used to specify a table address of the SYSCALL instruction and generate a target address. Be sure to set an appropriate value to this register before using the SYSCALL instruction.

Be sure to set a word address to the SCBP register.

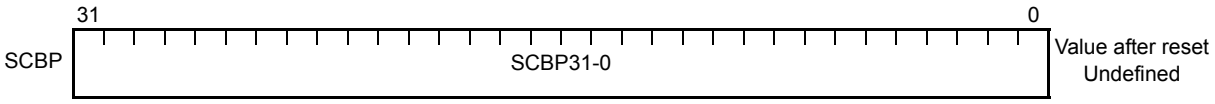


Table 3.28 SCBP Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	SCBP31-2	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the first address in the table used by the SYSCALL instruction.	R/W	Undefined
1, 0	SCBP1-0	These bits indicate the base pointer address of the SYSCALL instruction. These bits indicate the first address in the table used by the SYSCALL instruction. Always set these bits to 0.	R	0

(w) MCFG0 — Machine configuration register

This register indicates the CPU configuration.

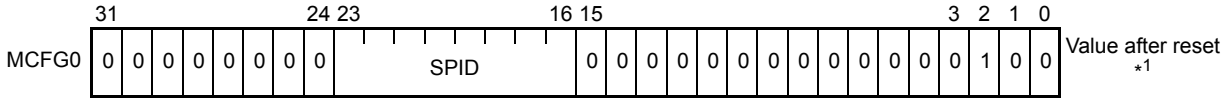


Table 3.29 MCFG0 Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 24	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
23 to 16	SPID	Bit 23 to 18: The RH850/F1M does not support these bits. (Reserved for future expansion. Be sure to set to 0.) Bit 17, 16: These bits indicate the system protection number.	R/W	*2
15 to 3	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
2	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
1, 0	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

Note 1. The value after reset is 0001 0004_H for CPU1 (PE1) of this product.

Note 2. The value after reset is 01_H for CPU1 (PE1) of this product.

(x) MCTL — Machine control register

This register is used to control the CPU.

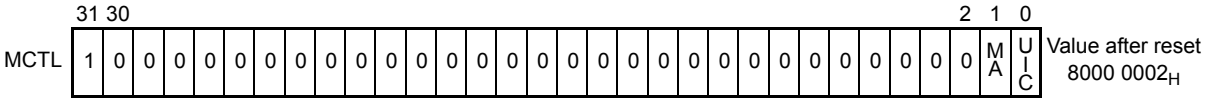


Table 3.30 MCTL Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
30 to 2	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
1	MA	This bit specifies the operation when a misaligned access occurs. 0: An exception occurs when a misaligned access has occurred. *1 1: Hardware operates normally.*2	R/W	1
0	UIC	This bit is used to control the interrupt enable/disable operation in user mode. When this bit is set to 1, executing the EI/DI instruction becomes possible in user mode.	R/W	0

Note 1. For the double-word format only, a misaligned access exception does not occur when data is placed at the word boundary rather than the double-word boundary(where the lower 3 bits of the address = 0).

Note 2. For the double-word format only, a misaligned access exception occur when data is not placed at the word boundary.

(3) Interrupt Function Registers

Table 3.31 Interrupt Function System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR7, 1	FPIPR	FPI exception interrupt priority setting	SV
SR10, 2	ISPR	Priority of interrupt being serviced	SV
SR11, 2	PMR	Interrupt priority masking	SV
SR12, 2	ICSR	Interrupt control status	SV
SR13, 2	INTCFG	Interrupt function setting	SV

(a) FPIPR — FPI exception interrupt priority setting register

This register is used to set the interrupt priority of FPI exception.

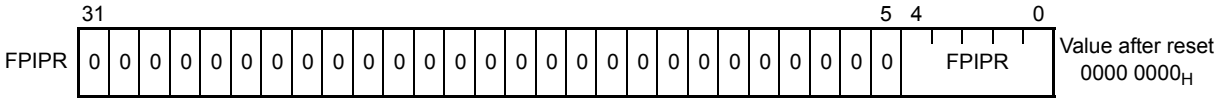


Table 3.32 FPIPR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
4 to 0	FPIPR	These bits specify the priority of the floating-point operation exception interrupt (FPI, indicating imprecision). Values from 0 to 16 should be used; setting a value of 17 or higher are prohibited. FPI exceptions are handled according to this interrupt priority. When generated at the same time as another interrupt with the same priority level, the FPI exception takes priority. **CAUTION** A set value of more than 16 is treated as 16.	R/W	0

(b) ISPR — Priority of interrupt being serviced register

This register retains the priority of the EIINTn interrupt being serviced. This priority value is then used to perform priority ceiling processing when multiple interrupts are generated.

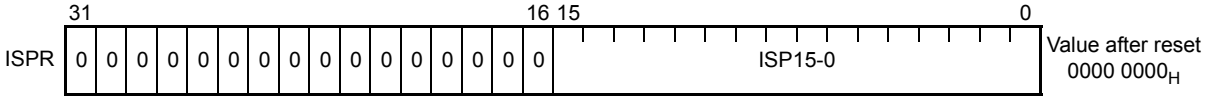


Table 3.33 ISPR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 16	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
15 to 0	ISP15-0	These bits indicate the acknowledgment status of an EIINTn interrupt with a priority*1 that corresponds to the relevant bit position. 0: An interrupt request for an interrupt whose priority corresponds to the relevant bit position has not been acknowledged. 1: An interrupt request for an interrupt whose priority corresponds to the relevant position is being serviced by the CPU core. The bit positions correspond to the following priority levels: **Bit** / **Priority** 0 / Priority 0 (highest) 1 / Priority 1 : / : 14 / Priority 14 15 / Priority 15 When an interrupt request (EIINTn) is acknowledged, the bit corresponding to the acknowledged interrupt request is automatically set to 1. If PSW.EP is 0 when the EIRET instruction is executed, the bit with the highest priority among the ISP15-0 bits that are set (0 is the highest priority) is cleared to 0.*2 While a bit in this register is set to 1, the same or lower priority interrupts (EIINTn) and FPI exceptions*3 are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged. When performing software-based priority control using the PMR register, be sure to clear this register by using the INTCFG.ISPC bit.	R*4	0

Note 1. For details, see *Section 4.1.5, Interrupt Exception Priority and Priority Masking*, in *the RH850G3M User's Manual: Software*.

Note 2. Interrupt acknowledgment and auto-updating of values when the EIRET instruction is executed are disabled by setting (1) to the INTCFG.ISPC bit. It is recommended to enable auto-updating of values, so in normal cases, the INTCFG.ISPC bit should be cleared to 0.

Note 3. Since FPI exceptions have the same level of priority as EIINTn interrupts, they are affected by the ISPR setting in the same way as interrupts. The priority of FPI exceptions is set by the FPIPR register.

Note 4. This is R or R/W, depending on the setting of the INTCFG.ISPC bit. It is recommended to use this register as a read-only (R) register.

(c) PMR — Interrupt priority masking register

This register is used to mask the specified interrupt priority.

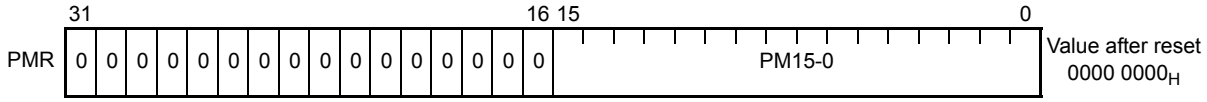


Table 3.34 PMR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 16	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
15 to 0	PM15-0	These bits mask an interrupt request with a priority level that corresponds to the relevant bit position. 0: Servicing of an interrupt request with a priority that corresponds to the relevant bit position is enabled. 1: Servicing of an interrupt request with a priority that corresponds to the relevant bit position is disabled. The bit positions correspond to the following priority levels: **Bit** / **Priority** 0 / Priority 0 (highest) 1 / Priority 1 : / : 14 / Priority 14 15 / Priority 15 and priority 16 (lowest) While a bit in this register is set to 1, interrupts (EIINTn) and FPI exceptions*1 with the priority corresponding to that bit are masked. Priority level judgment is therefore not performed when the system is determining whether to acknowledge an exception, meaning that exceptions will not be acknowledged*2.	R/W	0

Note 1. Since FPI exceptions are specified as the same level of priority as that of interrupts (EIINTn), it is affected by the PMR like interrupts. The priority of FPI exceptions is set by the FPIPR register.

Note 2. Specify the masks by setting the bits to 1 in order from the lowest-priority bit. For example, $FF00_H$ can be set, but $F0F0_H$ or $00FF_H$ cannot.

(d) ICSR — Interrupt control status register

This register indicates the interrupt control status in the CPU.

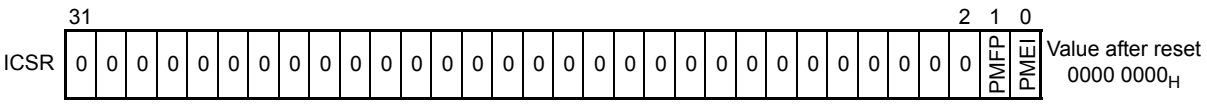


Table 3.35 ICSR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
1	PMFP	This bit indicates that an FPI exception with the priority level masked by the PMR register exists.	R	0
0	PMEI	This bit indicates that an interrupt (EIINTn) with the priority level masked by the PMR register exists.	R	0

(e) INTCFG — Interrupt function setting register

This register is used to specify settings related to the CPU's internal interrupt function.

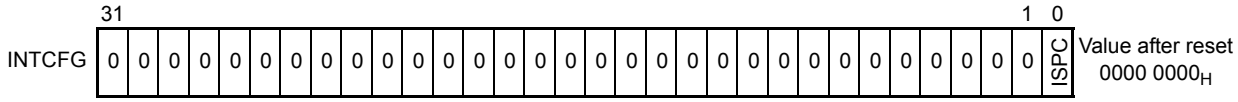


Table 3.36 INTCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 1	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
0	ISPC	This bit specifies how the ISPR register is updated. 0: The ISPR register is automatically updated. Updates triggered by the program (via execution of an LDSR instruction) are ignored. 1: The ISPR register is not automatically updated. Updates triggered by the program (via execution of an LDSR instruction) are performed. If this bit is cleared to 0, the bits of the ISPR register are automatically set to 1 when an interrupt (EIINTn) is acknowledged, and cleared to 0 when the EIRET instruction is executed. In this case, updating by the program (via execution of an LDSR instruction) is ignored. If this bit is set to 1, the bits of the ISPR register are not updated by the acknowledgment of an interrupt (EIINTn) or by execution of the EIRET instruction. In this case, the bits can be updated by an LDSR instruction executed by the program. In normal cases, the ISPC bit should be cleared. When performing software-based priority control, however, set this bit to 1 and perform priority control by using the PMR register.	R/W	0

(4) FPU Function Registers

(a) Floating-point operation registers

The FPU uses the CPU general-purpose registers (r0 to r31). There are no register files used only for floating-point operations.

- Single-precision floating-point instruction
  Thirty-two 32-bit registers can be specified. These registers correspond to the general-purpose registers r0 to r31.
- Double-precision floating-point instruction
  Sixteen 64-bit registers can be specified. These registers correspond to register pairs in the form of a pair of general-purpose registers ({r1, r0}, {r3, r2} … {r31, r30}). Each register pair is specified in the instruction format with an even numbered register. Because r0 is a zero register (always holds “0”), in principle {r1, r0} should not be used by a double-precision floating-point instruction.

(b) Floating-point system registers

The FPU can use the following system registers to control floating-point operations. Reading from/writing to a floating-point system register is performed by specifying the system register number, which consists of the register number and selection identifier, by using the LDSR and STSR instructions. For details of the registers, see *Section 3.4.2, Floating-Point Function System Registers* in *the RH850G3M User’s Manual: Software*.

Table 3.37 FPU System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR6, 0	FPSR	Floating-point operation setting/status	CU0 and SV
SR7, 0	FPEPC	Floating-point operation exception program counter	CU0 and SV
SR8, 0	FPST	Floating-point status	CU0
SR9, 0	FPCC	Floating-point operation comparison result	CU0
SR10, 0	FPCFG	Floating-point function setting	CU0
SR11, 0	FPEC	Floating-point operation exception control	CU0 and SV

(5) MPU function registers

The MPU function system registers are read from or written to by using the LDSR and STSR instructions and specifying the system register number, which is made up of a register number and selection ID.

Table 3.38 MPU Function System Registers (1/3)

Register No. (regID, selID)	Symbol	Function	Access Permission
SR0, 5	MPM	Memory protection operation mode setting	SV
SR1, 5	MPRC	MPU region control	SV
SR4, 5	MPBRGN	MPU base region number	SV
SR5, 5	MPTRGN	MPU end region number	SV
SR8, 5	MCA	Memory protection setting check address	SV
SR9, 5	MCS	Memory protection setting check size	SV
SR10, 5	MCC	Memory protection setting check command	SV

Table 3.38 MPU Function System Registers (2/3)

Register No. (regID, selID)	Symbol	Function	Access Permission
SR11, 5	MCR	Memory protection setting check result	SV
SR0, 6	MPLA0	Protection area lower limit address	SV
SR1, 6	MPUA0	Protection area upper limit address	SV
SR2, 6	MPAT0	Protection area attribute	SV
SR4, 6	MPLA1	Protection area lower limit address	SV
SR5, 6	MPUA1	Protection area upper limit address	SV
SR6, 6	MPAT1	Protection area attribute	SV
SR8, 6	MPLA2	Protection area lower limit address	SV
SR9, 6	MPUA2	Protection area upper limit address	SV
SR10, 6	MPAT2	Protection area attribute	SV
SR12, 6	MPLA3	Protection area lower limit address	SV
SR13, 6	MPUA3	Protection area upper limit address	SV
SR14, 6	MPAT3	Protection area attribute	SV
SR16, 6	MPLA4	Protection area lower limit address	SV
SR17, 6	MPUA4	Protection area upper limit address	SV
SR18, 6	MPAT4	Protection area attribute	SV
SR20, 6	MPLA5	Protection area lower limit address	SV
SR21, 6	MPUA5	Protection area upper limit address	SV
SR22, 6	MPAT5	Protection area attribute	SV
SR24, 6	MPLA6	Protection area lower limit address	SV
SR25, 6	MPUA6	Protection area upper limit address	SV
SR26, 6	MPAT6	Protection area attribute	SV
SR28, 6	MPLA7	Protection area lower limit address	SV
SR29, 6	MPUA7	Protection area upper limit address	SV
SR30, 6	MPAT7	Protection area attribute	SV
SR0, 7	MPLA8	Protection area lower limit address	SV
SR1, 7	MPUA8	Protection area upper limit address	SV
SR2, 7	MPAT8	Protection area attribute	SV
SR4, 7	MPLA9	Protection area lower limit address	SV
SR5, 7	MPUA9	Protection area upper limit address	SV
SR6, 7	MPAT9	Protection area attribute	SV
SR8, 7	MPLA10	Protection area lower limit address	SV
SR9, 7	MPUA10	Protection area upper limit address	SV
SR10, 7	MPAT10	Protection area attribute	SV
SR12, 7	MPLA11	Protection area lower limit address	SV
SR13, 7	MPUA11	Protection area upper limit address	SV
SR14, 7	MPAT11	Protection area attribute	SV
SR16, 7	MPLA12	Protection area lower limit address	SV
SR17, 7	MPUA12	Protection area upper limit address	SV
SR18, 7	MPAT12	Protection area attribute	SV
SR20, 7	MPLA13	Protection area lower limit address	SV
SR21, 7	MPUA13	Protection area upper limit address	SV
SR22, 7	MPAT13	Protection area attribute	SV

Table 3.38 MPU Function System Registers (3/3)

Register No. (regID, selID)	Symbol	Function	Access Permission
SR24, 7	MPLA14	Protection area lower limit address	SV
SR25, 7	MPUA14	Protection area upper limit address	SV
SR26, 7	MPAT14	Protection area attribute	SV
SR28, 7	MPLA15	Protection area lower limit address	SV
SR29, 7	MPUA15	Protection area upper limit address	SV
SR30, 7	MPAT15	Protection area attribute	SV

(a) MPM — Memory protection operation mode register

The memory protection mode register is used to define the basic operating state of the memory protection function.

	31																												2	1	0	
MPM	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	SVP	MPE	Value after reset 0000 0000$_H$

Table 3.39 MPM Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
1	SVP	In SV mode (when PSW.UM = 0), this bit is used to specify whether to restrict access according to the SX, SW, and SR bits of the MPAT register for each protection area.*1 0: As usual, implicitly enable all access in SV mode. 1: Restrict access according to the SX, SW, and SR bits even in SV mode.*2	R/W	0
0	MPE	This bit is used to specify whether to enable or disable the MPU function. 0: Disable 1: Enable	R/W	0

Note 1. If the SVP bit is set to 1, access will be restricted in accordance with the setting for each protection area, even in SV mode. Therefore, specify the protection area beforehand so that the access from the program which set the SVP bit is not restricted.

Note 2. If access is restricted in SV mode, execution of MDP exceptions or the MIP exception handling itself might not be possible depending on the settings. Be careful to specify settings so that access by the exception handler and to the memory area necessary for exception handling is permitted.

(b) MPRC — MPU region control register

Bits used to perform special memory protection function operations are located in this register.

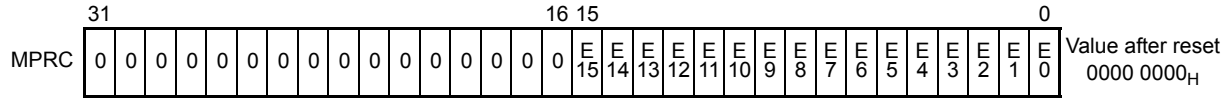


Table 3.40 MPRC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 16	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
15 to 0	E15-0	These are the enable bits for each protection area. Bit En is a copy of bit MPATn.E (where n = 15 to 0). The number of protection area for this product is 16.	R/W	0

(c) MPBRGN — MPU base region register

This register indicates the minimum usable MPU area number.

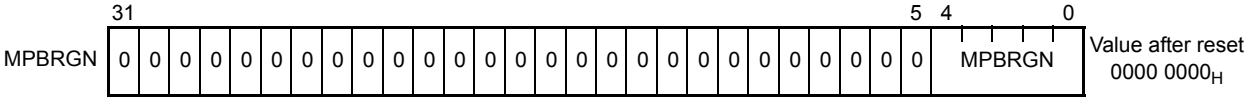


Table 3.41 MPBRGN Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
4 to 0	MPBRGN	These bits indicate the smallest number of an MPU area.	R	00000_B

(d) MPTRGN — MPU end region register

This register indicates the maximum usable MPU area number + 1.

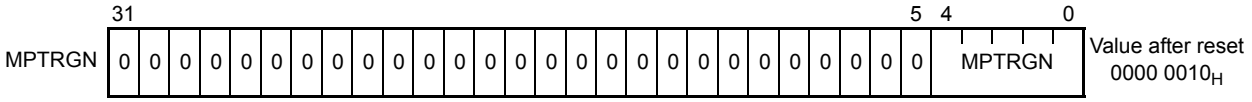


Table 3.42 MPTRGN Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 5	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
4 to 0	MPTRGN	These bits indicate the largest number of an MPU area plus one. These bits always indicate the maximum number of MPU areas that the hardware provides.	R	10_H

(e) MCA — Memory protection setting check address register

This register is used to specify the base address of the area for which a memory protection setting check is to be performed.

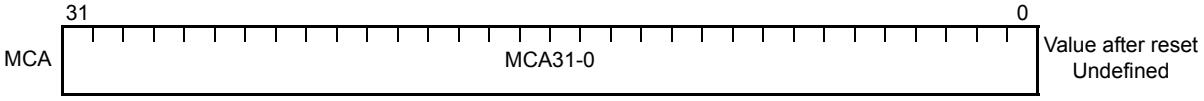


Table 3.43 MCA Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MCA31-0	These bits are used to specify the start address of the memory area subject to a memory protection setting check in bytes.	R/W	Undefined

(f) MCS — Memory protection setting check size register

This register is used to specify the size of the area for which a memory protection setting check is to be performed.

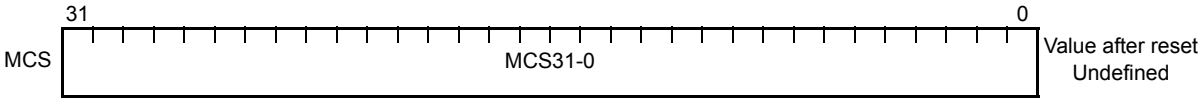


Table 3.44 MCS Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MCS31-0	These bits specify the size of the memory area subject to a memory protection setting check in bytes. Checking areas whose address values are lower than the address value in the MCA register is not possible because the specified size is handled as an unsigned integer. Do not set 0000 0000_H in the MCS register.	R/W	Undefined

(g) MCC — Memory protection setting check command register

This register is used to specify the base address of the area where memory protection settings are checked.

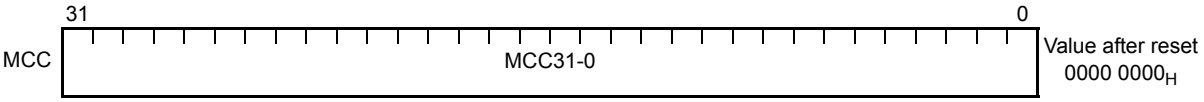


Table 3.45 MCC Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	MCC31-0	When any value is written to the MCC register, a memory protection setting check starts. Setting the MCA and MCS registers and then writing to this register leads to storage of the check results in the MCR register. Since writing any value to this register starts the check, no extra registers are required if r0 is used as the source register. The result of checking is reflected in MCR according to each area setting regardless of the setting of the PSW.UM bit. The value read from the MCC register is always 0000 0000_H.	R/W	0

(h) MCR — Memory protection setting check result register

This register is used to store the results of a memory protection setting check.

Be sure to clear bits 31 to 9, 7 and 6 to 0.

CAUTION

If the area for which checking is specified crosses 0000 0000$_H$ or 7FFF FFFF$_H$, the area is judged to have been specified incorrectly, and the MCR.OV bit is set to 1. Therefore, when referring to the results of checking, check the MCR.OV bit and confirm that the result is valid (OV = 0) before using any results of checking.

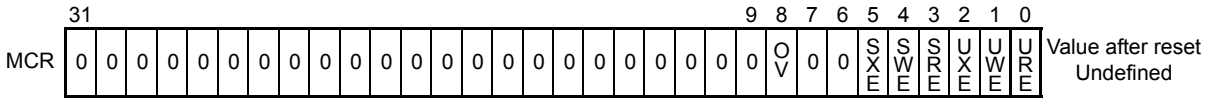


Table 3.46 MCR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 9	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
8	OV	If the specified area includes 0000 0000$_H$ or 7FFF FFFF$_H$, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
7, 6	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
5	SXE	If the specified area is contained within one of the protection areas and execution is permitted for that area in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
4	SWE	If the specified area is contained within one of the protection areas and writing to that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
3	SRE	If the specified area is contained within one of the protection areas and reading from that area is permitted in supervisor mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
2	UXE	If the specified area is contained within one of the protection areas and execution is permitted for that area in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
1	UWE	If the specified area is contained within one of the protection areas and writing to that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined
0	URE	If the specified area is contained within one of the protection areas and reading from that area is permitted in user mode, 1 is stored in this bit. In other cases, 0 is stored in this bit.	R/W	Undefined

(i) MPLAn — Protection area lower limit address register

These registers indicate the lower limit address of area n (where n = 0 to 15). The number of the protection areas in this product is 16.

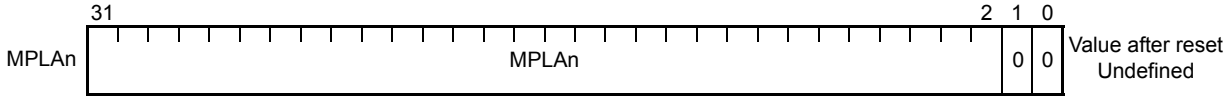


Table 3.47 MPLAn Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	MPLA31-2	These bits indicate the minimum address of area n. The MPLA1-0 bits are implicitly set to 0.	R/W	Undefined
1, 0	—	Reserved for future expansion. Be sure to clear to 0.	R	0

(j) MPUAn — Protection area upper limit address register

These registers indicate the upper limit address of area n (where n = 0 to 15).

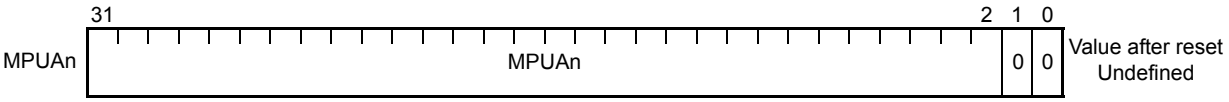


Table 3.48 MPUAn Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	MPUA31-2	These bits indicate the upper limit address of area n. The MPUA1-0 bits are implicitly set to 1.	R/W	Undefined
1, 0	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

(k) MPATn — Protection area attribute register

These registers indicate the attributes of area n (where n = 0 to 15).

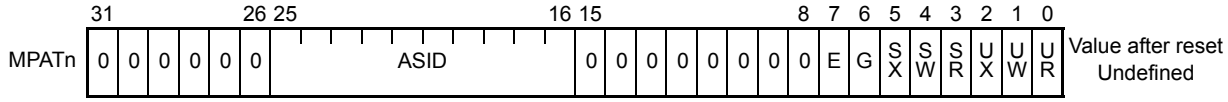


Table 3.49 MPATn Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 26	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
25 to 16	ASID	These bits indicate the ASID value to be used as the area match condition.	R/W	Undefined
15 to 8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7	E	This bit indicates whether area n is enabled or disabled. 0: Area n is disabled. 1: Area n is enabled.	R/W	0
6	G	0: Areas match only if ASIDs are equal. 1: Areas match even if ASIDs are not equal. When this bit is 0, MPATn.ASID = ASID.ASID is the condition for an area match. When this bit is 1, areas may match even if the values of MPATn.ASID and ASID.ASID are not equal.	R/W	Undefined
5	SX	This bit indicates the execution privilege for supervisor mode.*1 0: Execution is disabled. 1: Execution is enabled.	R/W	Undefined
4	SW	This bit indicates whether writing is enabled in supervisor mode.*1 0: Writing is disabled. 1: Writing is enabled.	R/W	Undefined
3	SR	This bit indicates whether reading is enabled in supervisor mode.*1 0: Reading is disabled. 1: Reading is enabled.	R/W	Undefined
2	UX	This bit indicates the execution privilege for user mode. 0: Execution is disabled. 1: Execution is enabled.	R/W	Undefined
1	UW	This bit indicates whether writing is enabled in user mode. 0: Writing is disabled. 1: Writing is enabled.	R/W	Undefined
0	UR	This bit indicates whether reading is enabled in user mode. 0: Reading is disabled. 1: Reading is enabled.	R/W	Undefined

Note 1. If access is restricted in SV mode, execution of MDP exceptions or the MIP exception handling itself might not be possible depending on the settings. Be careful to specify settings so that access by the exception handler and to the memory area necessary for exception handling is permitted.

(6) Cache Operation Function Registers

Reading and writing of the cache control system registers is performed by issuing an LDSR and STSR instructions specifying the system register number, which consists of the register number and selection identifier.

Table 3.50 Cache Operation Function Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR16, 4	ICTAGL	Instruction cache tag Lo access	SV
SR17, 4	ICTAGH	Instruction cache tag Hi access	SV
SR18, 4	ICDATL	Instruction cache data Lo access	SV
SR19, 4	ICDATH	Instruction cache data Hi access	SV
SR24, 4	ICCTRL	Instruction cache control	SV
SR26, 4	ICCFG	Instruction cache configuration	SV
SR28, 4	ICERR	Instruction cache error	SV

(a) ICTAGL — Instruction cache tag Lo access register

This register is used for CISTI and CILDI instructions for the instruction cache. This register stores the value to be stored in the tag RAM of the instruction cache by the execution of a CISTI instruction and the value read from the tag RAM of the instruction cache by the execution of a CILDI instruction.

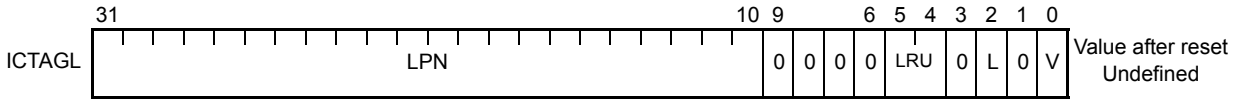
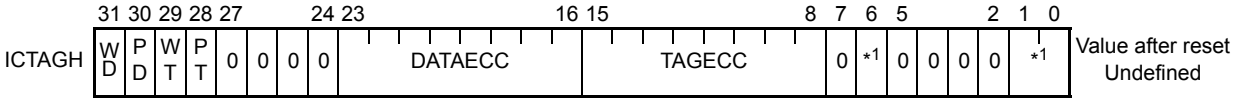


Table 3.51 ICTAGL Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 10	LPN	These bits retain the values of bits 24 to 11, i.e. the physical page numbers. When writing, always write 0 to bits 31 to 25 and 10.	R/W	Undefined
9 to 6	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
5, 4	LRU	These bits indicate the LRU information of the specified cache line. The CISTI instruction cannot alter the value of the LRU information.	R/W	Undefined
3	—	(Reserved for future expansion. This bit should be set to 0.)	R	0
2	L	This bit retains the lock information.	R/W	Undefined
1	—	(Reserved for future expansion. This bit should be set to 0.)	R	0
0	V	This bit retains whether the specified cache line is enabled or disabled.	R/W	Undefined

(b) ICTAGH — Instruction cache tag Hi access register

This register is used for a CISTI or CILDI instruction for the instruction cache. This register stores the value to be stored in the instruction cache tag RAM by the execution of a CISTI instruction and the value read from the instruction cache tag RAM by the execution of a CILDI instruction.



Note 1. Undefined

Table 3.52 ICTAGH Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	WD	When this bit is set to 1, executing a CISTI instruction updates the cache data RAM.	R/W	Undefined
30	PD	When this bit is set to 1, the value in the DATAECC field of this register is written to the ECC field of the data RAM on execution of a CISTI instruction. When this bit is cleared to 0, the ECC is automatically generated from the written data.	R/W	Undefined
29	WT	When this bit is set to 1, executing a CISTI instruction updates the cache data RAM.	R/W	Undefined
28	PT	When this bit is set to 1, the value in the TAGECC field is written to the ECC of the tag RAM on execution of a CISTI execution. When this bit is cleared to 0, the ECC is automatically generated from the written data.	R/W	Undefined
27 to 24	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
23 to 16	DATAECC	These bits retain the ECC of the data RAM.	R/W	Undefined
15 to 8	TAGECC	These bits retain the ECC of the tag RAM. Be sure to clear bit 15 to 0.	R/W	Undefined
7	—	(Reserved for future expansion. This bit should be set to 0.)	R	0
6	—	(Reserved for future expansion. This bit should be set to 0.)	R	Undefined
5 to 2	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
1, 0	—	(Reserved for future expansion. Be sure to clear to 0.)	R	Undefined

(c) ICDATL — Instruction cache data Lo access register

This register is used for a CISTI or CILDI instruction for the instruction cache. This register stores the value to be stored in the instruction cache data RAM by the execution of a CISTI instruction and the value read from the instruction cache data RAM by the execution of a CILDI instruction.

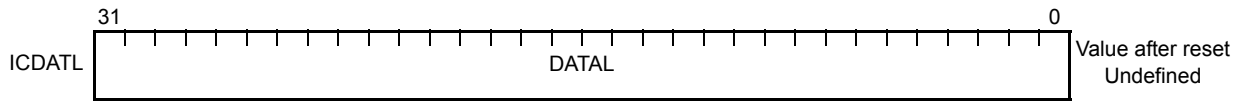


Table 3.53 ICDATL Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	DATAL	These bits retain the values of bits 31 to 0 or of bits 95 to 64 from the instruction data of the block within the specified cache lines. The offset of the index specifies the bit numbers to be retained. Index offset = 0000: bits 31 to 0 Index offset = 1000: bits 95 to 64	R/W	Undefined

(d) ICDATH — Instruction cache data Hi access register

This register is used for a CISTI or CILDI instruction for the instruction cache. This register stores the value to be stored in the instruction cache data RAM by the execution of a CISTI instruction and the value read from the instruction cache data RAM by the execution of a CILDI instruction.

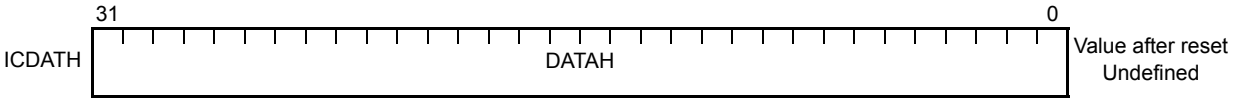


Table 3.54 ICDATH Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 0	DATAH	These bits retain the values of bit 63 to 32 or those of bit 127 to 96 from the instruction data of the block within the specified cache lines. The index offset specifies the bit numbers to be retained. Index offset = 0000: bits 63 to 32 Index offset = 1000: bits 127 to 96	R/W	Undefined

(e) ICCTRL — Instruction cache control register

This register controls the instruction cache.

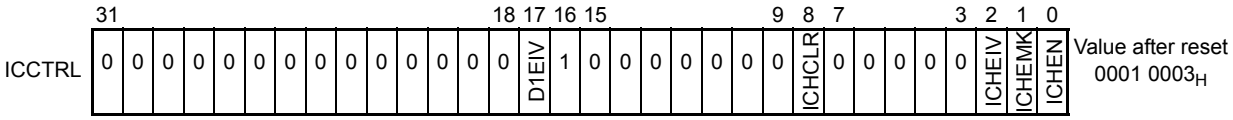


Table 3.55 ICCTRL Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 18	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
17	D1EIV	This bit selects the operation in response to 1-bit errors in the data RAM. 0: The error is corrected and then processing continues, but the entry that had an error is retained. 1: The error is not corrected, the entry is cleared, and fetching is repeated. Even if this bit is set, this bit retains the previous read value until the setting is reflected in the instruction cache.	R/W	0
16	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
15 to 9	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
8	ICHCLR	Setting this bit to 1 clears the whole instruction cache in a single operation. After this bit has been set to 1, it will be read as 1 until clearing is completed. The bit is cleared to 0 once clearing of the cache is completed.	R/W	0
7 to 3	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
2	ICHEIV	Setting this bit to 1 specifies that the instruction cache is automatically disabled (ICHEN bit is cleared to 0) when a cache error occurs.	R/W	0
1	ICHEMK	Setting this bit to 1 masks cache error exception notifications for the CPU when a cache error occurs.	R/W	1
0	ICHEN	This bit disables or enables the instruction cache. 0: Instruction cache is disabled. 1: Instruction cache is enabled.	R	1

(f) ICCFG — Instruction cache configuration register

This register shows the configuration of the instruction cache.

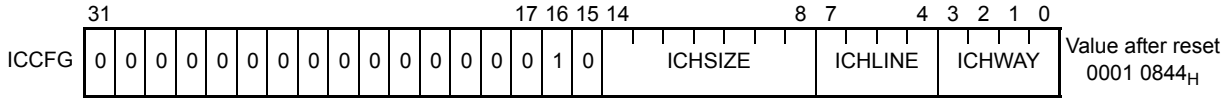


Table 3.56 ICCFG Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 17	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
16	—	(Reserved for future expansion. Be sure to set to 1.)	R	1
15	—	(Reserved for future expansion. This bit should be set to 0.)	R	0
14 to 8	ICHSIZE	These bits indicate the capacity (in Kbytes) of the instruction cache. 000 1000: 8 Kbytes	R	08_H
7 to 4	ICHLINE	These bits indicate the number of lines per 1 way of the instruction cache. 0100: 128 lines	R	4_H
3 to 0	ICHWAY	These bits indicate the number of ways of the instruction cache. 0100: 4 ways	R	4_H

(g) ICERR — Instruction cache error register

This register stores cache error data of the instruction cache.

Once the ICHERR bit is set to 1, subsequent cache error data is not stored in this register until the ICHERR bit is cleared to 0. During execution of CILDI, cache error is not stored.

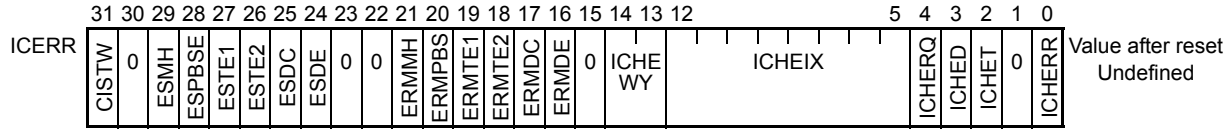


Table 3.57 ICERR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31	CISTW	This bit indicates that there was an error in the specification of the way subject to writing by a CISTI instruction. The write operation will complete and the entry information will be updated, but the V bit will be cleared when the corresponding bit is next read (because it will be judged that there is no corresponding data in the cache; i.e., that a cache miss occurred). The setting of this bit does not cause an exception notification to be sent to the CPU.	R/W	0
30	—	(Reserved for future expansion. This bit should be set to 0.)	R	0
29	ESMH	Error status: Multi-hit	R/W	Undefined
28	ESPBSE	Error status: WAY error	R/W	Undefined
27	ESTE1	Error status: 1-bit error in the tag RAM	R/W	Undefined
26	ESTE2	Error status: 2-bit error in the tag RAM	R/W	Undefined
25	ESDC	Error status: 1-bit correction in the data RAM	R/W	Undefined
24	ESDE	Error status: 2-bit error in the data RAM	R/W	Undefined
23, 22	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
21	ERMMH	Error exception notification mask: Multi-hit	R/W	0
20	ERMPBSE	Error exception notification mask: WAY error	R/W	0
19	ERMTE1	Error exception notification mask: 1-bit error in the tag RAM	R/W	0
18	ERMTE2	Error exception notification mask: 2-bit error in the tag RAM	R/W	0
17	ERMDC	Error exception notification mask: 1-bit correction in the data RAM	R/W	0
16	ERMDE	Error exception notification mask: 2-bit error in the data RAM	R/W	0
15	—	(Reserved for future expansion. This bit should be set to 0.)	R	0
14, 13	ICHEWY	These bits retain the number of the way where a cache error occurred.	R/W	Undefined
12 to 5	ICHEIX	These bits retain the index of the way where a cache error occurred.	R/W	Undefined
4	ICHERQ	Setting of this bit to 1 indicates that the CPU is being notified of a cache error exception. If cache error exceptions are masked, however, the CPU is not notified of an exception even when this bit is set to 1.	R/W	0
3	ICHED	This bit indicates that an error occurred in the data RAM.	R/W	0
2	ICHET	This bit indicates that an error occurred in the tag RAM.	R/W	0
1	—	(Reserved for future expansion. This bit should be set to 0.)	R	0
0	ICHERR	This bit is set to 1 when a cache error occurs.	R/W	0

(7) Data Buffer Operation Function Registers

The data buffer control system registers are provided for each physical CPU operating.

Table 3.58 Data Buffer Operation Function Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR24,13	CDBCR	Data buffer control register	SV

(a) CDBCR — Data buffer control register

This register controls the data buffer.

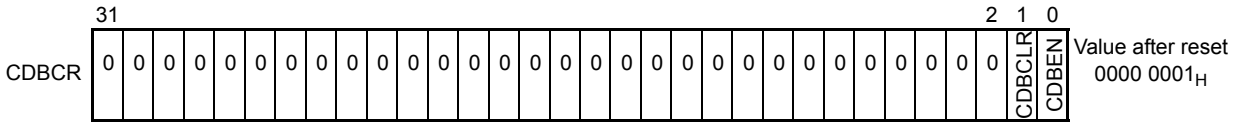


Table 3.59 CDBCR Register Contents

Bit Position	Bit Name	Function	R/W	Value after Reset
31 to 2	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
1	CDBCLR	When this bit is set to 1, all of the data buffer are cleared. This bit is always read as 0.	W	0
0	CDBEN	This bit disables or enables the data buffer. 0: Data buffer is disabled. 1: Data buffer is enabled.	R/W	1

3.2.2 Instruction Cache and Data Buffer

3.2.2.1 Features

A 8-Kbyte 4-way set-associative instruction cache is provided between CPUn (n = 1) and the code flash. The instruction cache and the code flash are connected by a 128-bit dedicated bus to minimize penalties caused by a cache miss. Also a data buffer is mounted between CPUn and the code flash to achieve high-speed data access. The 32-MB area from 0000 0000_H to 01FF $FFFF_H$ in the address space is intended for the instruction cache and data buffer.

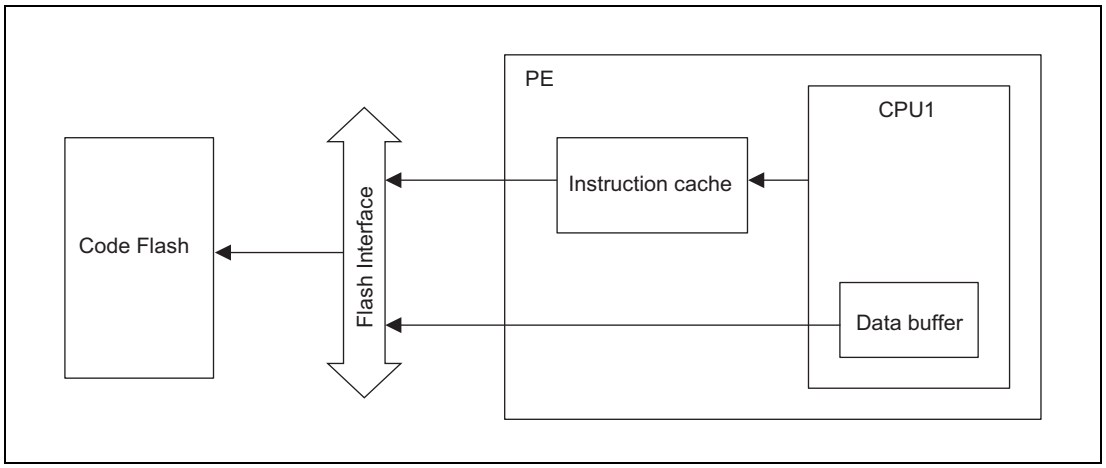


Figure 3.2 Instruction Cache and Data Buffer

3.2.2.2 Instruction Cache Function

The 8-Kbyte 4-way set-associative cache includes four ways consisting of 128-entry blocks of four words per line, amounting to 8-Kbyte capacity in total. The ways are divided into two groups, Way Group 0 consisting of Way0 and Way1 and Way Group 1 consisting of Way2 and Way3. The Way Group is selected and used by decoding the address information to be accessed. If a cache miss occurs, each line is refilled by a replace algorithm using LRU. When instructions are fetched from the code flash area, instruction codes are stored in the instruction cache.

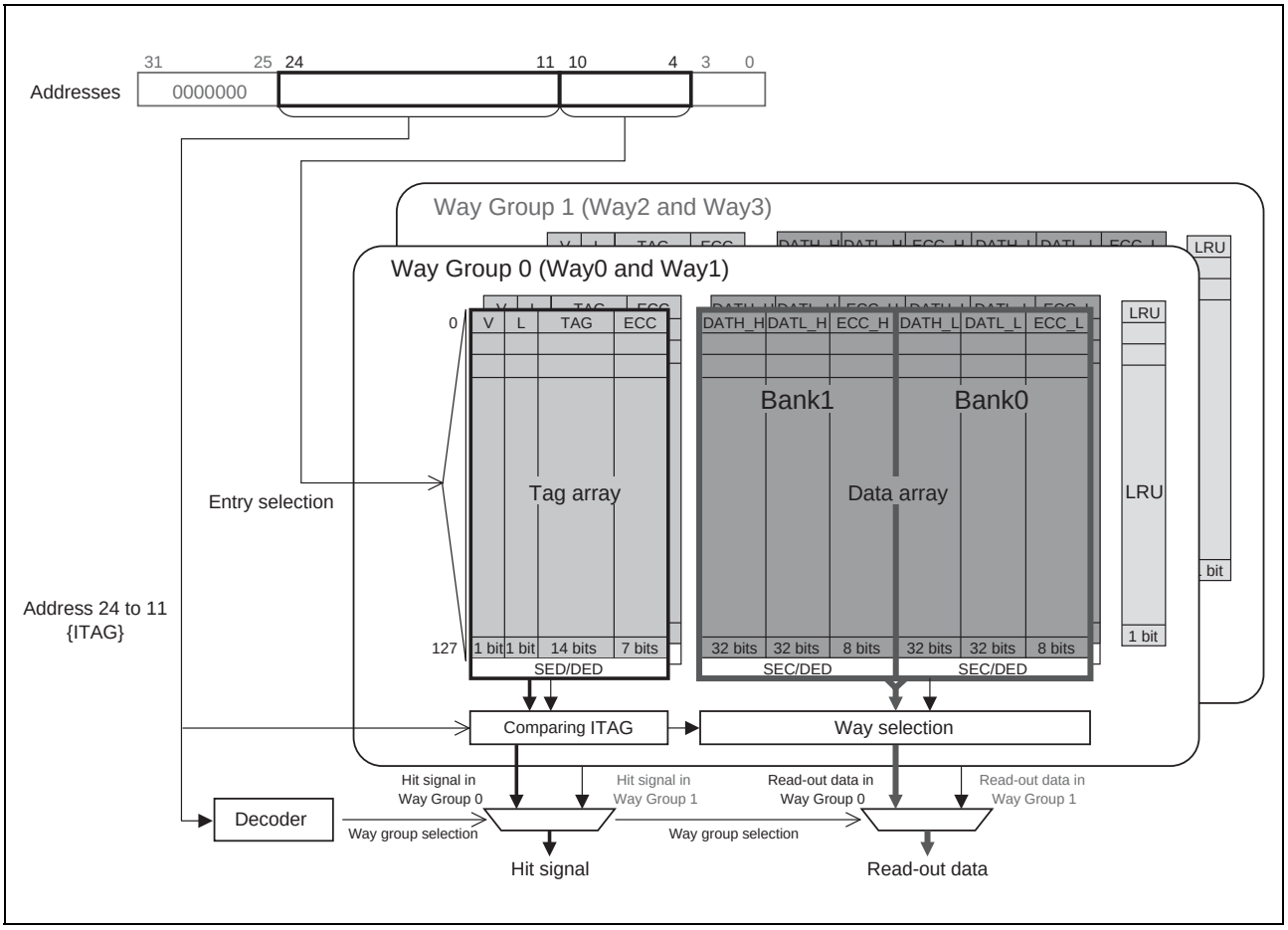


Figure 3.3 Instruction Cache Configuration

Tag Array

V bit	This bit indicates whether valid data is stored in the cache line. Setting of this bit to 1 makes the cache line data valid. The V bit is initialized to 0 by reset.
L bit	This bit indicates whether a cache line is locked or not. Setting of this bit to 1 locks the cache line and it cannot be replaced with new data. The L bit is valid only when the V bit is 1, and it is not initialized by reset.
TAG	Among 32 bits in the operable addresses of the data line to be cached, bits 24 to 11 are stored in the TAG bits. The TAG bits are not initialized by reset.
ECC	The ECC of the tag array is stored in this bit. The ECC bit is not initialized by reset.

Data Array

DATH_H, DATH_L, DATL_H, DATL_L	The 128-bit cache line data is stored in 32-bit units as follows: bits [127:96], [95:64], [63:32], and [31:0] are stored in DATH_H, DATL_H, DATH_L, and DATL_L, respectively. In the CISTI or CILDI operation executed by a CACHE instruction, the ICDATH register is used for DATH_H and DATH_L and the ICDATL register is used for DATL_H and DATL_L.
ECC_H, ECC_L	The ECC of the data in bits [127:64] and [63:0] is stored in ECC_H and ECC_L, respectively.

LRU

LRU	The LRU information in the same Way Group is stored in this data array. The LRU is initialized by reset.

CAUTION

When an instruction is fetched from an applicable line after issuing a CISTI instruction for writing test data to the tag array of the instruction cache, the tag information must be written Way Group units. For example, when writing tag information to a line on the Way0 side of Way Group 0, also write tag information for the same line to Way1, and then execute the instruction fetch.

- **When writing to Way Group 0 (Way0 and Way1), write a value such that the exclusive OR of the ICTAGL.LPN bit is 0.**
- **When writing to Way Group 1 (Way2 and Way3), write a value such that the exclusive OR of the ICTAGL.LPN bit is 1.**

Fetching an instruction after a value that does not follow the above rule has been written to the tag array causes a way error, which sets the ICERR.ESPBSE bit to 1. Fetching an instruction after the same tag information has been written to the same lines of both ways in a way group causes a multi-hit error, which sets the ICERR.ESMH bit to 1.

3.2.2.3 Data Buffer Function

A four-line buffer with 128 bits per line is mounted as a data buffer. Data of 128 bits per line read from the code flash is stored in the data buffer. When the same address is subsequently accessed, the data is read out from the data buffer, so the code flash does not need to be accessed again.

3.2.3 Reliability Functions

3.2.3.1 PE Guard Function (PEG)

(1) Overview of the PEG Function

The PEG is a constituent of the Slave Guard function and prevents unauthorized access to the resources in the CPU (PE) from an external master. This function protects access to the local RAM in the PE. In the initial state after a reset, access by masters other than own PE is disabled. Setting the registers listed in (3), List of PEG Protection Setting Registers, enables access by masters other than own PE.

(1) Detecting PE guard violation

If an external master makes an unauthorized access to the resource area in a PE for which PE guard is set, the access is detected as a PE guard violation.

(2) Blocking unauthorized access

When a PE guard violation is detected, unauthorized access to the internal resources of the PE are blocked to prevent unauthorized modification of the contents of PE resources.

(3) Notifying occurrence of violation

When a PE guard violation is detected, INTGUARD is generated.

(2) Protection by SPID

- Setting PEG Protection
  - Up to four areas can be set depending on the local RAM address of the own PE.
  - The area range is specified by the base address and the mask bit (4 Kbytes to 192 Kbytes).
  - “Read enable” and “write enable” can be set for each area.
  - “Enable” or “disable” can be selected based on the system protection identifier (SPID) for each area.
- Procedure for permitting access by using the system protection identifier (SPID) (see **Figure 3.4**)

1. Is the area subject to access is the local RAM area? If so, go to step 2.
2. Is the area subject to access is within the range of valid areas 0, 1, 2, or 3? If so, go to step 3. Otherwise, return an error response.
3. Are all the conditions below for the relevant area satisfied? If so, permit access.
   – The system protection identifier (SPID) is enabled.
   – Required operations (read/write) are enabled.
   Otherwise, return an error response.

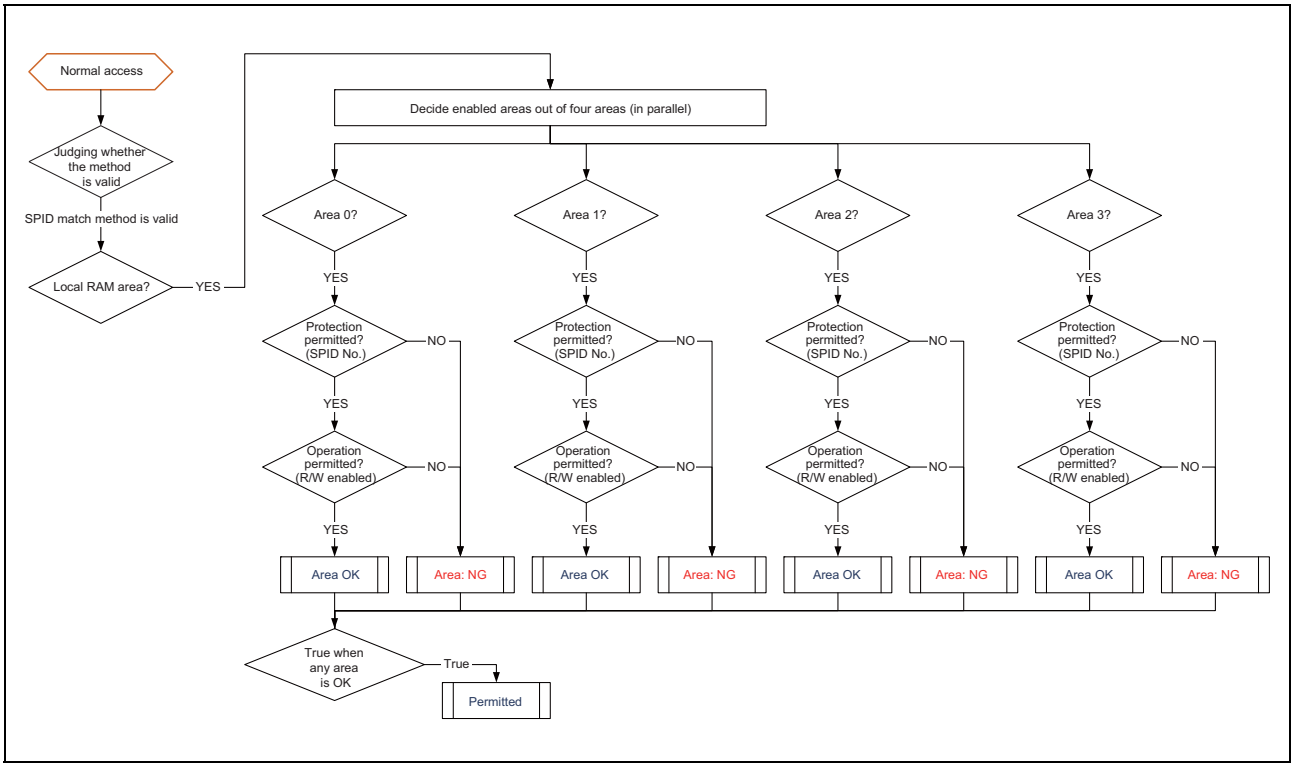


Figure 3.4 Access Permission by the System Protection Identifier (SPID)

(3) List of PEG Protection Setting Registers

Specify the necessary settings in the registers below to protect PE resources from unauthorized access by an external master.

- Whether to permit access to the local RAM in the PE can be specified.
- For accesses to the register set, no access restriction is provided independently for the PEG function.
- PEG protection is enabled by the following procedure:
  1. Set the PE guard protection area n mask setting register (PEGGnMK) (n = 0 to 3).
  2. Set the PE guard protection area n base setting register (PEGGnBA) (n = 0 to 3).

Table 3.60 List of Registers

Address Offset	Size (Byte)	Register Name	Abbreviation	Right	R/W	Operable Bit				Value after reset
						1	8	16	32	
$+00C_H$	2	PE guard SPID master decision control register	PEGSP	—	R/W	—	√	√	—	0000_H
$+080_H$	4	PE guard area 0 mask setting register	PEGG0MK	—	R/W	—	√	√	√	$0000\ 0000_H$
$+084_H$	4	PE guard area 0 base setting register	PEGG0BA	—	R/W	—	√	√	√	$0000\ 0000_H$
$+090_H$	4	PE guard area 1 mask setting register	PEGG1MK	—	R/W	—	√	√	√	$0000\ 0000_H$
$+094_H$	4	PE guard area 1 base setting register	PEGG1BA	—	R/W	—	√	√	√	$0000\ 0000_H$
$+0A0_H$	4	PE guard area 2 mask setting register	PEGG2MK	—	R/W	—	√	√	√	$0000\ 0000_H$
$+0A4_H$	4	PE guard area 2 base setting register	PEGG2BA	—	R/W	—	√	√	√	$0000\ 0000_H$
$+0B0_H$	4	PE guard area 3 mask setting register	PEGG3MK	—	R/W	—	√	√	√	$0000\ 0000_H$
$+0B4_H$	4	PE guard area 3 base setting register	PEGG3BA	—	R/W	—	√	√	√	$0000\ 0000_H$

Note: Base address: $FFFE\ E600_H$

(4) Register Set

(a) PEGSP — PE guard SPID master decision control register

This register is used to enable or disable access by an external master to the resources in the PE. The initial value of the SPEN bit is 0, which disables access to PE resources by an external master. Setting the SPEN bit to 1 enables access by an external master under the conditions set by PEGGnMK (n = 0 to 3) and PEGGnBA (n = 0 to 3).

Access: PEGSP can be read or written in 16-bit units.
PEGSPL can be read or written in 8-bit units.

Address: PEGSP: FFFE E60C$_H$,
PEGSPL: FFFE E60C$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 3.61 PEGSP Register Contents

Bit Position	Bit Name	Function
15 to 1	Reserved.	When read, the value after reset is returned. When writing, write the value after reset.
0	SPEN	This bit enables or disables detection of accesses by an external master having SPID. 0: Accesses by the external master having SPID is disabled. 1: Accesses by the external master having SPID is enabled.

(b) PEGGnMK — PE guard protection area n mask setting register (n = 0 to 3)

In combination with the PEGGnBA register (n = 0 to 3), this register specifies a range or ranges within PE guard protection area n. Setting a GnMASK bit to 1 masks the corresponding address bit of the PEGGnBA register and places the corresponding area or areas within the range of PE guard protection area n. Note that the minimum setting unit for PE guard protection area n is 4 KB.

Ex.) With the settings of PEGGnBA[31:12] = $FEBF6_H$ and PEGGnMK[31:12] = 00008_H, the PE guard protection area n is specified within the ranges of FEBF 6000_H to FEBF $6FFF_H$ and FEBF $E000_H$ to FEBF $EFFF_H$.

Access: PEGGnMK can be read or written in 32-bit units.
PEGGnMKL and PEGGnMKH can be read or written in 16-bit units.
PEGGnMKLH, PEGGnMKHL, and PEGGnMKHH can be read or written in 8-bit units.

Address: PEGGnMK: FFFE $E680_H$ + n × 10_H,
PEGGnMKL: FFFE $E680_H$ + n × 10_H,
PEGGnMKH: FFFE $E680_H$ + n × 10_H + 2_H,
PEGGnMKLH: FFFE $E680_H$ + n × 10_H + 1_H,
PEGGnMKHL: FFFE $E680_H$ + n × 10_H + 2_H,
PEGGnMKHH: FFFE $E680_H$ + n × 10_H + 3_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GnMASK[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GnMASK[15:12]				—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.62 PEGGnMK Register Contents

Bit Position	Bit Name	Function
31 to 12	GnMASK[31:12]	These bits determine whether to mask the base address bits PEGGnBA[31:12] that specify the range of PE guard protection area n. 0: Target address bits are compared when determining the PE guard area. 1: Target address bits are not compared when determining the PE guard area.
11 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(c) PEGGnBA — PE guard protection area n base setting register (n = 0 to 3)

In combination with the PEGGnMK register (n = 0 to 3), this register specifies a range or ranges within PE guard protection area n and sets the access enable conditions for the specified area. Setting the GnEN bit to 1 validates the access enable conditions specified by this register and the PEGGnMK register.

Access: PEGGnBA can be read or written in 32-bit units.
PPEGGnBAL and PEGGnBAH can be read or written in 16-bit units.
PEGGnBALL, PEGGnBALH, PEGGnBAHL, and PEGGnBAHH can be read or written in 8-bit units.

Address: PEGGnBA: FFFE $E684_H$ + n × 10_H,
PEGGnBAL: FFFE $E684_H$ + n × 10_H,
PEGGnBAH: FFFE $E684_H$ + n × 10_H + 2_H,
PEGGnBALL: FFFE $E684_H$ + n × 10_H,
PEGGnBALH: FFFE $E684_H$ + n × 10_H + 1_H,
PEGGnBAHL: FFFE $E684_H$ + n × 10_H + 2_H,
PEGGnBAHH: FFFE $E684_H$ + n × 10_H + 3_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GnBASE[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GnBASE[15:12]				—	—	—	—	—	—	GnSP1	GnSP0	—	GnWR	GnRD	GnEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 3.63 PEGGnBA Register Contents

Bit Position	Bit Name	Function
31 to 12	GnBASE [31:12]	These bits set a base address that specifies PE guard protection area n.
11 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	GnSP1	Enables accesses to PE guard protection area n by the external master having SPID = 1 (CPU). 0: Access by the external master having SPID = 1 is disabled. 1: Access by the external master having SPID = 1 is enabled.
4	GnSP0	Enables accesses to PE guard protection area n by the external master having SPID = 0 (peripheral device connected to H-BUS). 0: Access by the external master having SPID = 0 is disabled. 1: Access by the external master having SPID = 0 is enabled.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	GnWR	Enables write access to PE guard protection area n. 0: Write access is disabled. 1: Write access is enabled.
1	GnRD	Enables read access to PE guard protection area n. 0: Read access is disabled. 1: Read access is enabled.
0	GnEN	Enables settings for access enable conditions for PE guard protection area n. 0: Settings for access enable conditions are disabled. 1: Settings for access enable conditions are enabled.

CAUTION

The GnEN bit is cleared by writing to the PEGGnMK register.

3.2.4 Reliability Functions

3.2.4.1 System Error Notification Control Function (SEG)

Errors caused by an instruction fetch or data access can be the sources of system error exceptions. A system error exception is an FE level exception from which return or recovery is not possible.

For source codes (FEIC) of the system error exceptions and error handling, see **Table 3.66, Error Factor Codes and Handling of G3M Core System Error Exceptions**.

SEG (SysErrGen) controls the notification and recording of the error due to data access. Errors due to instruction-fetch access are not notified via SEG, but SEG is notified of errors that have occurred on the instruction cache RAM (data and tag). For details, see **(2) Register Set**, **(a) SEGCONT — Error notification control register**, and **(3) SEG Function**, **(b) Supplementary notes on SYSERR exception**.

Multiple error occurrence inputs are categorized according to error factor, and are processed sequentially from the highest-priority error factor, generating an FE-level asynchronous exception (SYSERR).

The bit position of the SEGFLAG register becomes the error factor priority. Error factors of lower-order bits take precedence over error factors of higher-order bits.

Error information is recorded only once regardless of error frequency.

The error with the highest priority among the error factors is valid when errors occur simultaneously. Recorded error information is not overwritten by subsequent errors.

(1) List of SEG Function Control Registers

Table 3.64 List of Registers

Address Offset*1	Size (Byte)	Register Name	Abbreviation	Right	R/W	Operable Bit				Value after reset
						1	8	16	32	
$+00_H$	2	Error notification control register	SEGCONT	—	R/W *2	—	—	√	—	0000_H
$+02_H$	2	Error occurrence retention register	SEGFLAG	—	R/W *2	—	—	√	—	0000_H

Note 1. Base address: $FFFE\ E980_H$

Note 2. Write access in user mode is ignored.

NOTE

- If an access is made with an address offset or operable bits other than those specified above, SYSERR exception is generated.
- Write access is only possible in supervisor mode (UM = 0). Attempting to write in other modes leads to generate SYSERR exception. In above cases, VCIF flag set.
- No restriction is provided for read accesses.
- Read accesses to ranges permitted by other protection systems are enabled at any time.

(2) Register Set

(a) SEGCONT — Error notification control register

This register is used to enable (= 1) or disable (= 0) notification of SysErr request in response to error flags that store the error occurrence status for each factor.

Access: SEGCONT can be read or written in 16-bit units.

Address: SEGCONT: FFFE E980$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	VPGE	—	—	TCME	ROME	VCIE	—	ICCE	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R	R/W	R	R

Table 3.65 SEGCONT Register Contents (1/2)

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	VPGE	Notification that the following P-Bus error was returned: • P-Bus guard error in write access
8,7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	TCME	Notification that one of the following errors occurred during data access to the local RAM by the own PE: • An uncorrectable ECC error • Detection of an access to the following area PE1 – FEC0 0000$_H$ to FEDB FFFF$_H$ – FEA0 0000$_H$ to FEBB FFFF$_H$
5	ROME	Notification that the following error occurred during access to the code flash while the table was being referenced by a table reference interrupt: • An uncorrectable ECC error • An address parity error

Table 3.65 SEGCONT Register Contents (2/2)

Bit Position	Bit Name	Function
4	VCIE	• Notification that the following P-Bus errors were returned (excluding errors in writing to P-Bus). – Access to an unimplemented area (FFFF 7900$_H$ to FFFF 7EFF$_H$) – P-Bus guard error – Read or write access to a peripheral in the following domain that is not clocked. C_AWO_TAUJ, C_AWO_ADCA, C_ISO_PERI1, C_ISO_PERI2, C_ISO_LIN, C_ISO_CAN, C_ISO_CSI. • Notification that the following code flash error was returned: – An uncorrectable ECC error – Address Parity error • Notification that the following global RAM error was returned: – Access protection violation – An uncorrectable ECC error • Notification that access to an interconnect reserved area was detected. PE1 – FFFF 0000$_H$ to FFFF 4FFF$_H$ – FFFE 0000$_H$ to FFFE BFFF$_H$ – FB00 0000$_H$ to FE9F FFFF$_H$ – F300 0000$_H$ to F8FF FFFF$_H$ • Notification that PE guard error occurred
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ICCE	Instruction Cache Error Notification Enable The following error is handled when the instruction cache system register, ICCTRL.ICHEMK, is set to 0 (the value after reset is 1): An uncorrectable ECC error
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 3.66 Error Factor Codes and Handling of G3M Core System Error Exceptions

Factor Codes	Error Contents
10	Reserved
11	Instruction fetch errors (from code flash memory)
12	Errors whose notification is enabled by the SEGCONT second bit
13	Instruction fetch errors (from other than code flash memory)
14	Errors whose notification is enabled by the SEGCONT fourth bit
15	Errors whose notification is enabled by the SEGCONT fifth bit
16	Errors whose notification is enabled by the SEGCONT sixth bit
17	Reserved
18	Reserved
19	Errors whose notification is enabled by the SEGCONT ninth bit
1A	Reserved
1B	Reserved
1C	Reserved
1D	Reserved
1E	Reserved
1F	Reserved

(b) SEGFLAG — Error occurrence retention register

This register contains error flags that indicate the status of error occurrence for each factor. This flag is set to 1 when an error occurs but is not automatically cleared to 0.

The values of this register can be both set and cleared by writing.

Access: SEGFLAG can be read or written in 16-bit units.

Address: SEGFLAG: FFFE E982$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	VPGF	—	—	TCMF	ROMF	VCIF	—	ICCF	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R	R/W	R	R

Table 3.67 SEGFLAG Register Contents

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	VPGF	Flag corresponding to bit 9 of the SEGCONT register
8, 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	TCMF	Flag corresponding to bit 6 of the SEGCONT register
5	ROMF	Flag corresponding to bit 5 of the SEGCONT register
4	VCIF	Flag corresponding to bit 4 of the SEGCONT register
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ICCF	Flag corresponding to bit 2 of the SEGCONT register
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(3) SEG Function

(a) SEG function: SYSERR request notification by error flag

- Setting an error flag takes precedence over clearing the same flag.
  - A simultaneous clearing operation is ignored.
- Priority of error factors
  - The bit position of the notification-enabled SEGFLAG register becomes the error factor priority. Error factors of lower bits take precedence over error factors of upper bits. Notification is made from the highest-priority error factor.
  - The bit position of error factors is reported as a “SYSERR factor code”.
- Conditions for starting SYSERR request notification
  - In case error notification is disabled by SEGCONT register, even if SEGFLAG register is set to 1, notification is not made.
  - In case error notification is enabled by SEGCONT register, notification is made immediately after SEGFLAG register is set to 1.
  - After clearing of SEGFLAG register, notification is made depending on SEGFLAG register state (re-arbitration).
- Finishing notification at a SYSERR request response
  - Even after notification is finished, SEGFLAG register (flags that indicate the status) is not cleared automatically.
  - Error notification is not made until re-arbitration is performed by setting or clearing the flag again.
  - If an error flag that is prioritized higher than the error factor is set prior to a request response, the error notification information may be replaced with an upper SYSERR factor code.

(b) Supplementary notes on SYSERR exception

- The PSW.EBV bit retains its value and the base address for the exception handler is not changed even when a SYSERR exception is generated.
- Detection of errors in the instruction cache
  Detection of an error in the instruction cache RAM does not lead to a SYSERR exception of the type that requires re-execution by the source of the instruction fetch. The entry that caused the error is automatically invalidated and the instruction is fetched from the code flash again, allowing the CPU to continue with instruction execution. The error that occurred during instruction fetch can be specified to be reported to SEG by setting the ICCTRL.ICHEMK bit to 0. Regarding errors in the instruction cache, see **3.2.1.2 Register Set**, **(g) ICERR — Instruction cache error register**.

3.3 Notes

3.3.1 Synchronization of Store Instruction Completion and Subsequent Instruction Execution

When a control register is updated by a store instruction, there is a time lag after the CPU executes the store instruction and before the control register is actually updated. Therefore, if the updated content of the control register is to be used by the instruction following the store instruction, the appropriate synchronization is required. How to perform synchronization processing is shown below.

For the procedures to synchronize updating system registers by LDSR instruction and the subsequent instruction execution, see APPENDIX A. Hazard Resolution Procedure for System Registers in the RH850G3M User's Manual: Software.

3.3.1.1 When the updated results in the control registers are to be used by the subsequent instruction

Example 1:

This includes the following case: an interrupt is enabled by execution of an EI instruction after an interrupt request is cleared by access from the control register in the INTC2 and the peripheral circuits. Proceed as follows in this case.

1. Execute the store instruction to update a control register (ST.W, etc.)
2. Perform a dummy read of the above control register (LD.W etc.).
3. Execute SYNCP.
4. Execute the subsequent instruction (EI).

Example 2:

When you must wait until a control register (control register A) has been completely updated before accessing another control register (control register B), execute similar processing. This includes the following cases: different peripheral functions are linked, or the interrupt mask for INTC is cleared after the peripheral function is set. Note that this processing is not required if the control registers A and B belong to the same peripheral group (Peripheral Group 0 to 5).

1. Execute the store instruction to update the control register A (ST.W, etc.)
2. Perform a dummy read of the above control register (LD.W etc.).
3. Execute SYNCP.
4. Execute the store instruction to access the control register B (ST.W, LD.W, etc.)

The same processing is also required when starting to access control registers and memory to be protected after waiting for the completion of settings for safety functions such as memory protection, ECC checking, and so on.

For details of the peripheral groups, see **Section 1.1.2, Internal Block Diagram** and **Section 4.3 Peripheral I/O Address Map**.

3.3.1.2 When the updated results of the control register or memory are to be used in the instruction fetch of the subsequent instruction

(a) In case of writing the instructions to the RAM before jumping to the RAM to execute instructions from the RAM, take the following procedure.

1. Execute the store instruction to update a memory (ST.W, etc).
2. Perform a dummy read of the above memory (LD.W, etc).
3. Execute SYNCP.
4. Execute SYNCI.
5. Execute the subsequent instruction (branch instruction, etc).

(b) In case of updating control registers for memory protection and ECC functions before jumping to the memory to be controlled by the registers, take the following procedure.

1. Execute the store instruction to update a control register (ST.W, etc).
2. Perform a dummy read of the control register (LD.W, etc).
3. Execute SYNCP.
4. Execute SYNCI.
5. Execute the subsequent instruction (branch instruction, etc).

When switching the code flash memory area:

In this case, see Section 10, Usage Notes, (7) Updating the BFASELR register in the ***RH850/F1L, RH850/F1M, RH850/F1H Flash Memory User's Manual: Hardware Interface***.

3.3.2 Accessing Registers by Using Bit-Manipulation Instructions

Writing by using bit-manipulation instructions consists of atomic read-modify-write processing in 8-bit units. Thus, access by a bit-manipulation instruction is only possible for registers for which reading and writing in 8-bit units is possible. If a register includes multiple flag bits, the read-modify-write operation may lead to the clearing of flags that were not actually targets for clearing.

Write access to the FlexRay and MEMC registers by using bit-manipulation instructions is not atomic. Access by other masters may interrupt the read-modify-write processing of these instructions.

3.3.3 Ensuring Coherency after Code Flash Update

The CPU has an efficient instruction cache and data buffer for the code flash area. Therefore, after using self-programming to program the code flash memory, clear the instruction cache and data buffer to ensure coherency. The instruction cache and data buffer can be cleared by using the ICCTRL register and the CDBCR register, respectively.

3.3.4 Overwriting Context upon Acceptance of Multiple Exceptions

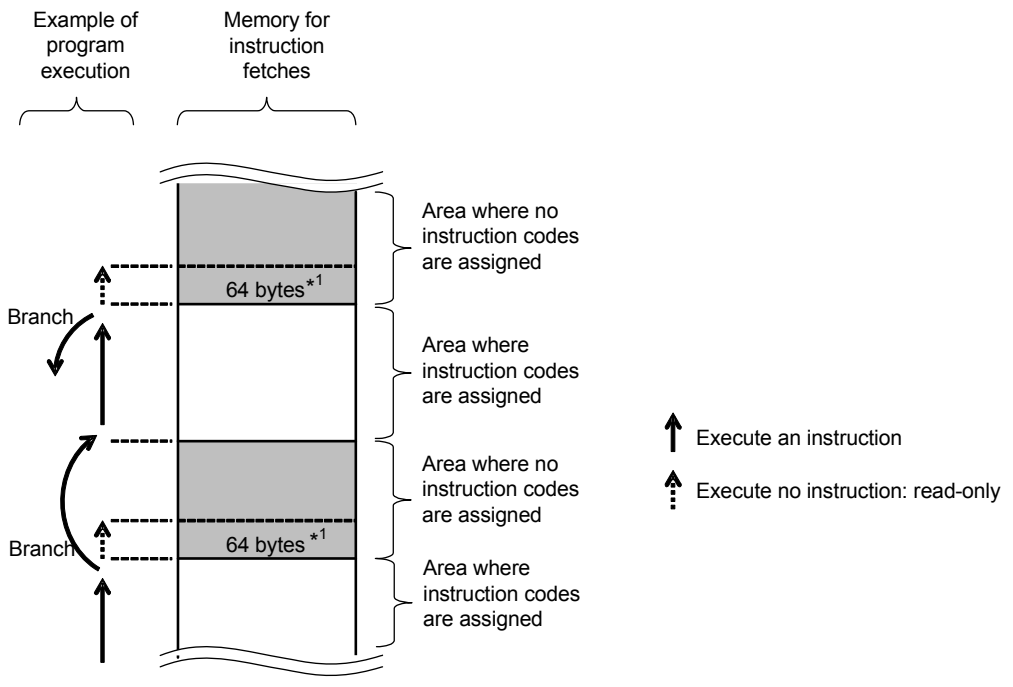
Acceptance of an exception depends on the type of exception source, regardless of the states of the ID and NP bits in the PSW register. When multiple exceptions are generated, the contents of the system register which hold the context information are overwritten. For the conditions for acceptance and whether correct return or recovery is possible for each exception source, see the List of Exception Sources in the *RH850G3M User's Manual: Software*.

3.3.5 Notes on Prefetching

CPU executes speculative instruction fetching from locations beyond the current program counter to maintain the throughput of instruction fetches. This prefetching may result in reading from memory areas to which no instruction codes have been stored (note 1 in **Figure 3.5**). Please note the following. The CPU does not execute values read in such cases.

These notes apply to all locations where instruction fetching is possible.

- Occurrence of ECC errors due to values in memory being undefined
  This prefetching may lead to an ECC error in already erased code flash or local RAM or retention RAM before initialization. When instruction codes are stored to memory, initialize the memory area with some data (note 1 in **Figure 3.5**).
- Detection of illegal access by the GRG
  The GRG function may detect such prefetching as illegal access. To prevent prefetching being detected as an illegal access, do not allow any region of overlap area between applicable areas (note 1 in **Figure 3.5**) and areas to which access is prohibited by the GRG. Reading from an area protected by the MPU does not cause a memory protection exception.
- Access to Access Prohibited Area
  Store instruction codes to memory without allowing any overlap between applicable areas (note 1 in **Figure 3.5**) and an access-prohibited area.



Note 1. Area from final address of instruction code to final address + 64 bytes. Even if instruction codes are not placed in this area, requests for prefetching may lead to reading from memory at the time of branching.

Figure 3.5 Area that Requires Attention Regarding Prefetching

3.3.6 System Register Hazards

To resolve potential hazards when updating the register values of some system registers, implement the following procedures.

- Instruction fetch:
  When fetching instructions after updating the following registers, start to do so only after executing the EIRET instruction, FERET instruction, or SYNCI instruction after executing the register update instruction.
  - PSW.UM, MCFG0.SPID

  When fetching instructions after updating the following registers, start to do so only after executing the SYNCI instruction after executing the register update instruction.
  - ASID, MPU: All related registers (Register number: SR*, 5-7)
- SYSCALL instruction:
  When a SYSCALL instruction is to be executed after updating the register below, execute a SYNCP instruction after the instruction to update the register and before the SYSCALL instruction.
  - SCCFG
- Load/Store:
  When executing instructions that involve load/store operations after updating the following registers, execute the load/store instruction only after executing the SYNCP instruction after executing the register update instruction.
  - ASID, MPU protection area setting registers (Register number: SR*, 6-7)

  Do not execute instructions that involve load/store operations during the instruction preceding and the instruction following update of the following registers.
  - MCTL.MA
- Interrupts:
  Update the following registers in the interrupt disabled state (PSW.ID = 1).
  - PSW.EBV, EBASE, INTBP, ISPR, PMR, ICSR, INTCFG
- Instruction cache clear operation:
  When the confirmation of instruction cache clear operation completed, check the read value of ICCTRL.ICHCLR bit.
- FPU register update:
  When the following register by executing the instruction is updated, execute the SYNCP instruction after executing the instruction.
  - FPU All related registers (Register number: SR6-11, 0)
- FPP/FPI exception mode change:
  When the exception FPP/FPI mode is changed, update the following register after executing the SYNCP and SYNCE instruction. For change the register, apply also the above "FPU register update".
  - FPSR.PEM

3.3.7 Access to an External Memory Area

Writing to an external memory area in 64 bits (ST.DW, PREPARE, and PUSHSP instructions) are not supported.

Section 4 Address Space

4.1 Address Space

Table 4.1 and **Table 4.2** show the address space of the RH850/F1M.

CAUTION

Do not access an address with which no register is mapped in the on-chip I/O register space. In addition, do not access any access prohibited area specified in Table 4.1 and Table 4.2. If such an address is accessed, operation is not guaranteed.

Table 4.1 Address Space (3-MB Product)

Address	Address Space Type	Size
0000 0000_H to 002F $FFFF_H$ (0001 7000_H to 0001 $7FFF_H$)	Code flash (User area read) (FCU firmware area (Map can be swapped by FCUFAREA register))*1	3 MB (4 KB)
0030 0000_H to 00FF $FFFF_H$	Access prohibited area	
0100 0000_H to 0100 $7FFF_H$	Code flash (extended area)	32 KB
0100 8000_H to 1001 $FFFF_H$	Access prohibited area	
1002 0000_H to 1002 $1FFF_H$	On-chip peripheral I/O area	8 KB
1002 2000_H to 1002 $FFFF_H$	Access prohibited area	
1003 0000_H to 1003 $03FF_H$	On-chip peripheral I/O area	1 KB
1003 0400_H to 1FFF $FFFF_H$	Access prohibited area	
2000 0000_H to 200F $FFFF_H$	External memory area (CS0)	1 MB*2
2010 0000_H to 21FF $FFFF_H$	Access prohibited area	
2200 0000_H to 220F $FFFF_H$	External memory area (CS1)	1 MB*2
2210 0000_H to 23FF $FFFF_H$	Access prohibited area	
2400 0000_H to 240F $FFFF_H$	External memory area (CS2)	1 MB*2
2410 0000_H to 27FF $FFFF_H$	Access prohibited area	
2800 0000_H to 280F $FFFF_H$	External memory area (CS3)	1 MB*2
2810 0000_H to FEBC $FFFF_H$	Access prohibited area	
FEBD 0000_H to FEBF $FFFF_H$	Local RAM	192 KB
FEC0 0000_H to FEDC $FFFF_H$	Access prohibited area	
FEDD 0000_H to FEDF $FFFF_H$	Local RAM (mirror)	192 KB
FEE0 0000_H to FEEF $FFFF_H$	Access prohibited area	
FEF0 0000_H to FEF0 $FFFF_H$	Retention RAM	64 KB
FEF1 0000_H to FEFF $FFFF_H$	Access prohibited area	
FF00 0000_H to FFFD $FFFF_H$ (FF20 0000_H to FF20 $FFFF_H$) (FFA1 2000_H to FFA1 $2FFF_H$)	On-chip peripheral I/O area (Data Flash (Read/Write)) (FCU RAM)	16 MB - 128 KB (64 KB) (4 KB)
FFFE 0000_H to FFFE DFFF$_H$	Access prohibited area	
FFFE $E000_H$ to FFFE FFFF$_H$	On-chip peripheral I/O area	8 KB
FFFF 0000_H to FFFF 4FFF$_H$	Access prohibited area	
FFFF 5000_H to FFFF FFFF$_H$	On-chip peripheral I/O area	44 KB

Note 1. For details, see the *RH850/F1L, RH850/F1M, RH850/F1H Flash Memory User's Manual: Hardware Interface*.

Note 2. 144 pin products do not have an external memory area.

Table 4.2 Address Space (4-MB Product)

Address	Address Space Type	Size
0000 0000_H to 003F $FFFF_H$ (0001 7000_H to 0001 $7FFF_H$)	Code flash (User area read) (FCU firmware area (Map can be swapped by FCUFAREA register))*1	4 MB (4 KB)
0040 0000_H to 00FF $FFFF_H$	Access prohibited area	
0100 0000_H to 0100 $7FFF_H$	Code flash (extended area)	32 KB
0100 8000_H to 1001 $FFFF_H$	Access prohibited area	
1002 0000_H to 1002 $1FFF_H$	On-chip peripheral I/O area	8 KB
1002 2000_H to 1002 $FFFF_H$	Access prohibited area	
1003 0000_H to 1003 $03FF_H$	On-chip peripheral I/O area	1 KB
1003 0400_H to 1FFF $FFFF_H$	Access prohibited area	
2000 0000_H to 200F $FFFF_H$	External memory area (CS0)	1 MB*2
2010 0000_H to 21FF $FFFF_H$	Access prohibited area	
2200 0000_H to 220F $FFFF_H$	External memory area (CS1)	1 MB*2
2210 0000_H to 23FF $FFFF_H$	Access prohibited area	
2400 0000_H to 240F $FFFF_H$	External memory area (CS2)	1 MB*2
2410 0000_H to 27FF $FFFF_H$	Access prohibited area	
2800 0000_H to 280F $FFFF_H$	External memory area (CS3)	1 MB*2
2810 0000_H to FEBB $FFFF_H$	Access prohibited area	
FEBC 0000_H to FEBF $FFFF_H$	Local RAM	256 KB
FEC0 0000_H to FEDB $FFFF_H$	Access prohibited area	
FEDC 0000_H to FEDF $FFFF_H$	Local RAM (mirror)	256 KB
FEE0 0000_H to FEEF $FFFF_H$	Access prohibited area	
FEF0 0000_H to FEF0 $FFFF_H$	Retention RAM	64 KB
FEF1 0000_H to FEFF $FFFF_H$	Access prohibited area	
FF00 0000_H to FFFD $FFFF_H$ (FF20 0000_H to FF20 $FFFF_H$) (FFA1 2000_H to FFA1 $2FFF_H$)	On-chip peripheral I/O area (Data Flash (Read/Write)) (FCU RAM)	16 MB - 128 KB (64 KB) (4 KB)
FFFE 0000_H to FFFE $DFFF_H$	Access prohibited area	
FFFE $E000_H$ to FFFE $FFFF_H$	On-chip peripheral I/O area	8 KB
FFFF 0000_H to FFFF $4FFF_H$	Access prohibited area	
FFFF 5000_H to FFFF $FFFF_H$	On-chip peripheral I/O area	44 KB

Note 1. For details, see the *RH850/F1L, RH850/F1M, RH850/F1H Flash Memory User's Manual: Hardware Interface*.

Note 2. 144 pin products do not have an external memory area.

4.2 Address Space Viewed from Each Bus Master

Figure 4.1 shows address spaces viewed from each bus master.

4.2.1 Data Space Accessible by CPU1

See **Figure 4.1** for the accessible spaces from CPU1.

4.2.2 Data Space Accessible by DMA

See **Figure 4.1** for the accessible spaces from the DMA.

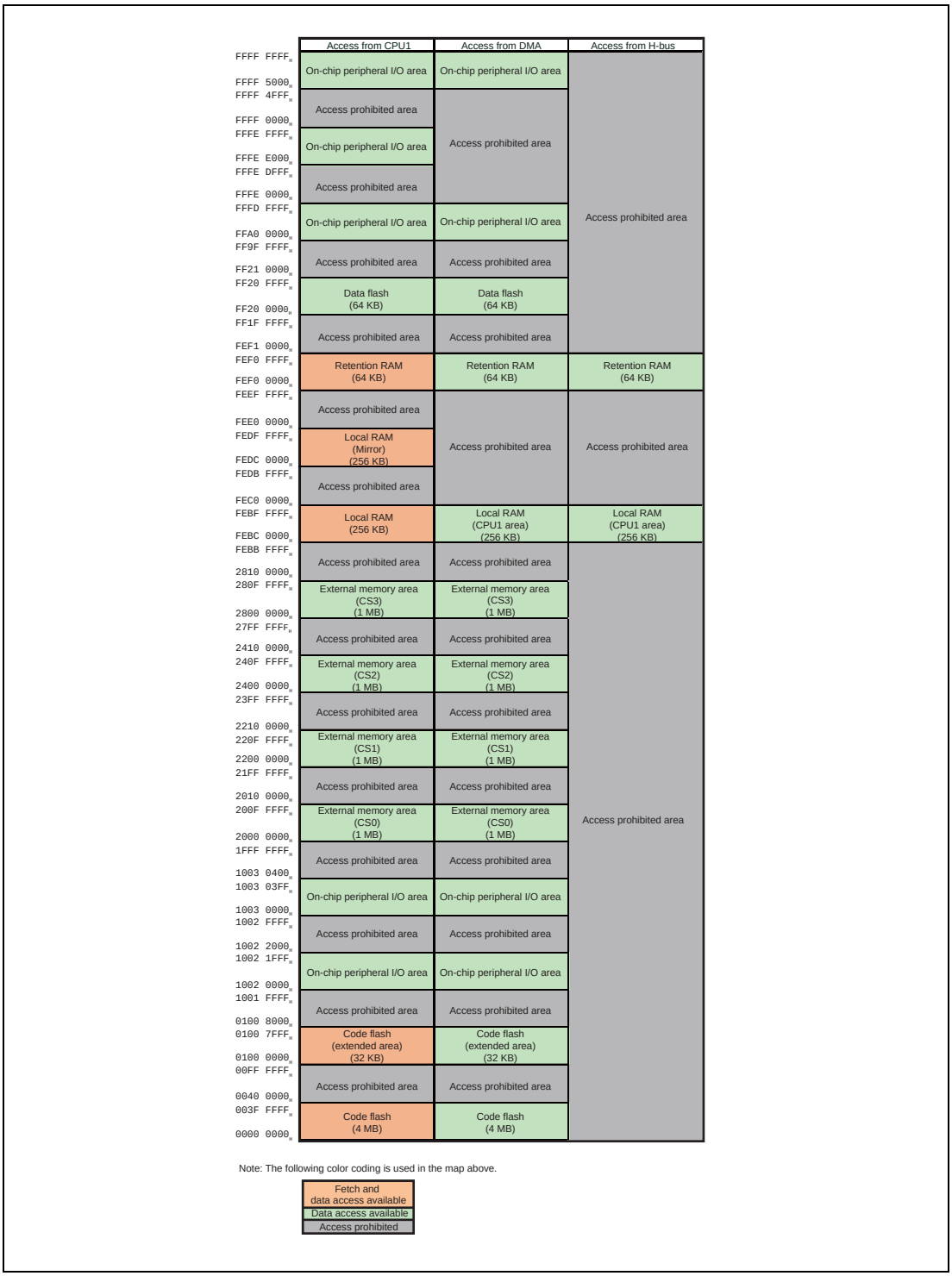


Figure 4.1 Address Space Viewed from Each Bus Master (233 pin 4 MB Product)

4.3 Peripheral I/O Address Map

Table 4.3 Peripheral I/O Address Map (1/5)

Address	Peripheral Group	Peripheral I/O
FF00 0000$_H$ to FF1F FFFF$_H$	—	Access prohibited area
FF20 0000$_H$ to FF20 FFFF$_H$	0	Data flash
FF21 0000$_H$ to FF9F FFFF$_H$	—	Access prohibited area
FFA0 0000$_H$ to FFA0 001F$_H$	3	FLMD
FFA0 0020$_H$ to FFA0 7FFF$_H$	—	Access prohibited area
FFA0 8000$_H$ to FFA0 801F$_H$	3	SELF
FFA0 8020$_H$ to FFA0 FFFF$_H$	—	Access prohibited area
FFA1 0000$_H$ to FFA1 010F$_H$	3	Flash controller (except FCUAREA register)
FFA1 0110$_H$ to FFA1 1FFF$_H$	—	Access prohibited area
FFA1 2000$_H$ to FFA1 2FFF$_H$	3	FCURAM
FFA1 3000$_H$ to FFA1 FFFF$_H$	—	Access prohibited area
FFA2 0000$_H$ to FFA2 FFFF$_H$	3	FACI command-issuing area
FFA3 0000$_H$ to FFBF FFFF$_H$	—	Access prohibited area
FFC0 0000$_H$ to FFC0 000F$_H$	3	FENMI
FFC0 0010$_H$ to FFC0 00FF$_H$	—	Access prohibited area
FFC0 0100$_H$ to FFC0 010F$_H$	3	FEINT
FFC0 0110$_H$ to FFC0 0FFF$_H$	—	Access prohibited area
FFC0 1000$_H$ to FFC0 100F$_H$	3	SELB_INTC
FFC0 1010$_H$ to FFC0 FFFF$_H$	—	Access prohibited area
FFC1 0000$_H$ to FFC1 4C5F$_H$	3	PORT
FFC1 4C60$_H$ to FFC1 FFFF$_H$	—	Access prohibited area
FFC2 0000$_H$ to FFC2 04CF$_H$	3	PORT (JTAG)
FFC2 04D0$_H$ to FFC2 FFFF$_H$	—	Access prohibited area
FFC3 0000$_H$ to FFC3 00CF$_H$	3	PORT (DNF)
FFC3 00D0$_H$ to FFC3 3FFF$_H$	—	Access prohibited area
FFC3 4000$_H$ to FFC3 408F$_H$	3	FCLA
FFC3 4090$_H$ to FFC4 8FFF$_H$	—	Access prohibited area
FFC4 9000$_H$ to FFC4 930F$_H$	0	Global RAM Guard
FFC4 9310$_H$ to FFC5 8FFF$_H$	—	Access prohibited area
FFC5 9000$_H$ to FFC5 900F$_H$	0	FCUFAREA
FFC5 9010$_H$ to FFC5 97FF$_H$	—	Access prohibited area
FFC5 9800$_H$ to FFC5 981F$_H$	0	DCIB
FFC5 9820$_H$ to FFC6 03FF$_H$	—	Access prohibited area
FFC6 0400$_H$ to FFC6 065F$_H$	0	Instruction cache data RAM ECC
FFC6 0660$_H$ to FFC6 13FF$_H$	—	Access prohibited area
FFC6 1400$_H$ to FFC6 165F$_H$	0	Instruction cache tag RAM ECC
FFC6 1660$_H$ to FFC6 1FFF$_H$	—	Access prohibited area
FFC6 2000$_H$ to FFC6 200F$_H$	0	Code flash address parity
FFC6 2010$_H$ to FFC6 21FF$_H$	—	Access prohibited area
FFC6 2200$_H$ to FFC6 235F$_H$	0	Code flash ECC (VCI)
FFC6 2360$_H$ to FFC6 23FF$_H$	—	Access prohibited area
FFC6 2400$_H$ to FFC6 255F$_H$	0	Code flash ECC (CPU1)

Table 4.3 Peripheral I/O Address Map (2/5)

Address	Peripheral Group	Peripheral I/O
FFC6 2560$_H$ to FFC6 29FF$_H$	—	Access prohibited area
FFC6 2A00$_H$ to FFC6 2A1F$_H$	0	Data flash ECC
FFC6 2A20$_H$ to FFC6 3FFF$_H$	—	Access prohibited area
FFC6 4000$_H$ to FFC6 465F$_H$	0	Global RAM ECC
FFC6 4660$_H$ to FFC6 4FFF$_H$	—	Access prohibited area
FFC6 5000$_H$ to FFC6 502F$_H$	0	Local RAM ECC TEST
FFC6 5030$_H$ to FFC6 53FF$_H$	—	Access prohibited area
FFC6 5400$_H$ to FFC6 541F$_H$	0	Local RAM ECC (PE1)
FFC6 5420$_H$ to FFC6 FFFF$_H$	—	Access prohibited area
FFC7 0000$_H$ to FFC7 003F$_H$	2	ECC CSIH0-3
FFC7 0040$_H$ to FFC7 0FFF$_H$	—	Access prohibited area
FFC7 1000$_H$ to FFC7 105F$_H$	2	ECC CAN
FFC7 1060$_H$ to FFC7 1FFF$_H$	—	Access prohibited area
FFC7 2000$_H$ to FFC7 301F$_H$	2	ECC FLXA
FFC7 3020$_H$ to FFC7 7FFF$_H$	—	Access prohibited area
FFC7 8000$_H$ to FFC7 800F$_H$	3	SELB_READTEST
FFC7 8010$_H$ to FFC9 FFFF$_H$	—	Access prohibited area
FFCA 0000$_H$ to FFCA 004F$_H$	3	RIIC
FFCA 0050$_H$ to FFCC FFFF$_H$	—	Access prohibited area
FFCD 0000$_H$ to FFCD 00DF$_H$	3	SCDS
FFCD 00E0$_H$ to FFCD FFFF$_H$	—	Access prohibited area
FFCE 0000$_H$ to FFCE 013F$_H$	3	RLIN2
FFCE 0140$_H$ to FFCE 1FFF$_H$	—	Access prohibited area
FFCE 2000$_H$ to FFCE 216F$_H$	3	RLIN3
FFCE 2170$_H$ to FFCF FFFF$_H$	—	Access prohibited area
FFD0 0000$_H$ to FFD0 19FF$_H$	2	RSCAN0
FFD0 1A00$_H$ to FFD6 CFFF$_H$	—	Access prohibited area
FFD6 D000$_H$ to FFD6 D51F$_H$	2	ADCA1
FFD6 D520$_H$ to FFD6 FFFF$_H$	—	Access prohibited area
FFD7 0000$_H$ to FFD7 002F$_H$	2	OSTM0
FFD7 0030$_H$ to FFD7 00FF$_H$	—	Access prohibited area
FFD7 0100$_H$ to FFD7 012F$_H$	2	OSTM1
FFD7 0130$_H$ to FFD7 01FF$_H$	—	Access prohibited area
FFD7 0200$_H$ to FFD7 022F$_H$	2	OSTM2
FFD7 0230$_H$ to FFD7 02FF$_H$	—	Access prohibited area
FFD7 0300$_H$ to FFD7 032F$_H$	2	OSTM3
FFD7 0330$_H$ to FFD7 03FF$_H$	—	Access prohibited area
FFD7 0400$_H$ to FFD7 042F$_H$	2	OSTM4
FFD7 0430$_H$ to FFD7 FFFF$_H$	—	Access prohibited area
FFD8 0000$_H$ to FFD8 107F$_H$	2	CSIH0
FFD8 1080$_H$ to FFD8 1FFF$_H$	—	Access prohibited area
FFD8 2000$_H$ to FFD8 307F$_H$	2	CSIH1
FFD8 3080$_H$ to FFD8 3FFF$_H$	—	Access prohibited area
FFD8 4000$_H$ to FFD8 507F$_H$	2	CSIH2

Table 4.3 Peripheral I/O Address Map (3/5)

Address	Peripheral Group	Peripheral I/O
FFD8 5080$_H$ to FFD8 5FFF$_H$	—	Access prohibited area
FFD8 6000$_H$ to FFD8 707F$_H$	2	CSIH3
FFD8 7080$_H$ to FFD8 7FFF$_H$	—	Access prohibited area
FFD8 8000$_H$ to FFD8 901F$_H$	2	CSIG0
FFD8 9020$_H$ to FFD8 9FFF$_H$	—	Access prohibited area
FFD8 A000$_H$ to FFD8 B01F$_H$	2	CSIG1
FFD8 B020$_H$ to FFDC FFFF$_H$	—	Access prohibited area
FFDD 0000$_H$ to FFDD 00EF$_H$	1	PIC
FFDD 00F0$_H$ to FFE1 FFFF$_H$	—	Access prohibited area
FFE2 0000$_H$ to FFE2 029F$_H$	1	TAUD
FFE2 02A0$_H$ to FFE2 3FFF$_H$	—	Access prohibited area
FFE2 4000$_H$ to FFE2 400F$_H$	1	SELB_TAUD0
FFE2 4010$_H$ to FFE2 FFFF$_H$	—	Access prohibited area
FFE3 0000$_H$ to FFE3 029F$_H$	1	TAUB0
FFE3 02A0$_H$ to FFE3 0FFF$_H$	—	Access prohibited area
FFE3 1000$_H$ to FFE3 129F$_H$	1	TAUB1
FFE3 12A0$_H$ to FFE4 FFFF$_H$	—	Access prohibited area
FFE5 0000$_H$ to FFE5 00AF$_H$	1	TAUJ0
FFE5 00B0$_H$ to FFE5 0FFF$_H$	—	Access prohibited area
FFE5 1000$_H$ to FFE5 10AF$_H$	1	TAUJ1
FFE5 10B0$_H$ to FFE5 3FFF$_H$	—	Access prohibited area
FFE5 4000$_H$ to FFE5 400F$_H$	1	SELB_TAUJ0
FFE5 4010$_H$ to FFE6 FFFF$_H$	—	Access prohibited area
FFE7 0000$_H$ to FFE7 00FF$_H$	1	PWSA
FFE7 0100$_H$ to FFE7 0FFF$_H$	—	Access prohibited area
FFE7 1000$_H$ to FFE7 27EF$_H$	1	PWGA
FFE7 27F0$_H$ to FFE7 27FF$_H$	—	Access prohibited area
FFE7 2800$_H$ to FFE7 281F$_H$	1	PWBA
FFE7 2820$_H$ to FFE7 2FFF$_H$	—	Access prohibited area
FFE7 3000$_H$ to FFE7 300F$_H$	1	SLPWGA
FFE7 3010$_H$ to FFE7 7FFF$_H$	—	Access prohibited area
FFE7 8000$_H$ to FFE7 807F$_H$	1	RTCA
FFE7 8080$_H$ to FFE7 FFFF$_H$	—	Access prohibited area
FFE8 0000$_H$ to FFE8 004F$_H$	1	ENCA0
FFE8 0050$_H$ to FFE8 FFFF$_H$	—	Access prohibited area
FFE9 0000$_H$ to FFE9 002F$_H$	1	TAPA0
FFE9 0030$_H$ to FFEC FFFF$_H$	—	Access prohibited area
FFED 0000$_H$ to FFED 000F$_H$	1	WDTA0
FFED 0010$_H$ to FFED 0FFF$_H$	—	Access prohibited area
FFED 1000$_H$ to FFED 100F$_H$	1	WDTA1
FFED 1010$_H$ to FFF1 FFFF$_H$	—	Access prohibited area
FFF2 0000$_H$ to FFF2 051F$_H$	3	ADCA0
FFF2 0520$_H$ to FFF6 FFFF$_H$	—	Access prohibited area
FFF7 0000$_H$ to FFF7 002F$_H$	3	DCRA0

Table 4.3 Peripheral I/O Address Map (4/5)

Address	Peripheral Group	Peripheral I/O
FFF7 0030$_H$ to FFF7 0FFF$_H$	—	Access prohibited area
FFF7 1000$_H$ to FFF7 102F$_H$	3	DCRA1
FFF7 1030$_H$ to FFF7 1FFF$_H$	—	Access prohibited area
FFF7 2000$_H$ to FFF7 202F$_H$	3	DCRA2
FFF7 2030$_H$ to FFF7 2FFF$_H$	—	Access prohibited area
FFF7 3000$_H$ to FFF7 302F$_H$	3	DCRA3
FFF7 3030$_H$ to FFF7 7FFF$_H$	—	Access prohibited area
FFF7 8000$_H$ to FFF7 800F$_H$	3	KR
FFF7 8010$_H$ to FFF7 FFFF$_H$	—	Access prohibited area
FFF8 0000$_H$ to FFF8 000F$_H$	5	Write protected register
FFF8 0010$_H$ to FFF8 00FF$_H$	—	Access prohibited area
FFF8 0100$_H$ to FFF8 011F$_H$	5	STBC
FFF8 0120$_H$ to FFF8 03FF$_H$	—	Access prohibited area
FFF8 0400$_H$ to FFF8 040F$_H$	5	STBC (WUF0)
FFF8 0410$_H$ to FFF8 051F$_H$	—	Access prohibited area
FFF8 0520$_H$ to FFF8 052F$_H$	5	STBC (WUF20)
FFF8 0530$_H$ to FFF8 075F$_H$	—	Access prohibited area
FFF8 0760$_H$ to FFF8 0A0F$_H$	5	Reset controller / Supply Voltage Monitor (LVI, VLVI)
FFF8 0A10$_H$ to FFF8 0AFF$_H$	—	Access prohibited area
FFF8 0B00$_H$ to FFF8 0B0F$_H$	5	STBC (IOHOLD)
FFF8 0B10$_H$ to FFF8 0FFF$_H$	—	Access prohibited area
FFF8 1000$_H$ to FFF8 271F$_H$	5	Clock controller
FFF8 2720$_H$ to FFF8 27FF$_H$	—	Access prohibited area
FFF8 2800$_H$ to FFF8 280F$_H$	5	FOUT
FFF8 2810$_H$ to FFF8 2FFF$_H$	—	Access prohibited area
FFF8 3000$_H$ to FFF8 304F$_H$	5	LPS
FFF8 3050$_H$ to FFF8 30FF$_H$	—	Access prohibited area
FFF8 3100$_H$ to FFF8 311F$_H$	5	Supply Voltage Monitor (CVM)
FFF8 3120$_H$ to FFF8 31FF$_H$	—	Access prohibited area
FFF8 3200$_H$ to FFF8 320F$_H$	5	Write protected register
FFF8 3210$_H$ to FFF8 7FFF$_H$	—	Access prohibited area
FFF8 8000$_H$ to FFF8 800F$_H$	5	Write protected register
FFF8 8010$_H$ to FFF8 810F$_H$	—	Access prohibited area
FFF8 8110$_H$ to FFF8 811F$_H$	5	STBC (WUFISO)
FFF8 8120$_H$ to FFF8 8FFF$_H$	—	Access prohibited area
FFF8 9000$_H$ to FFF8 AC0F$_H$	5	Clock controller
FFF8 AC10$_H$ to FFF8 BFFF$_H$	—	Access prohibited area
FFF8 C000$_H$ to FFF8 C01F$_H$	5	CLMA0
FFF8 C020$_H$ to FFF8 C0FF$_H$	—	Access prohibited area
FFF8 C100$_H$ to FFF8 C10F$_H$	5	CLMATEST
FFF8 C110$_H$ to FFF8 C1FF$_H$	—	Access prohibited area
FFF8 C200$_H$ to FFF8 C20F$_H$	5	Write protected register
FFF8 C210$_H$ to FFF8 CFFF$_H$	—	Access prohibited area
FFF8 D000$_H$ to FFF8 D01F$_H$	5	CLMA1

Table 4.3 Peripheral I/O Address Map (5/5)

Address	Peripheral Group	Peripheral I/O
FFF8 $D020_H$ to FFF8 $DFFF_H$	—	Access prohibited area
FFF8 $E000_H$ to FFF8 $E01F_H$	5	CLMA2
FFF8 $E020_H$ to FFFE $DFFF_H$	—	Access prohibited area
FFFE $E000_H$ to FFFE $E5FF_H$	—	Access prohibited area
FFFE $E600_H$ to FFFE $E6BF_H$	self	PEG
FFFE $E6C0_H$ to FFFE $E97F_H$	—	Access prohibited area
FFFE $E980_H$ to FFFE $E98F_H$	self	SEG
FFFE $E990_H$ to FFFE $E9FF_H$	—	Access prohibited area
FFFE $EA00_H$ to FFFE $EB7F_H$	self	INTC1
FFFE $EB80_H$ to FFFF $4FFF_H$	—	Access prohibited area
FFFF 5000_H to FFFF $7FFF_H$	—	Access prohibited area
FFFF 8000_H to FFFF $8BFF_H$	0	DMA
FFFF $8C00_H$ to FFFF $AFFF_H$	—	Access prohibited area
FFFF $B000_H$ to FFFF $BD7F_H$	0	INTC2
FFFF $BD80_H$ to FFFF $FFFF_H$	—	Access prohibited area

Section 5 Write-Protected Registers

This section contains a generic description of the write-protected registers.

The first part in this section describes the features specific to the write-protected registers, and the ensuing sections describe the various registers.

5.1 Overview

5.1.1 Functional Overview

The RH850/F1M products require a special procedure using write-protected registers to set important registers that affect the system, such as clock, reset, and port-related registers. The settings of protected registers are protected against unauthorized writing by programs by requiring this special procedure. For details about the protected registers, see **Section 5.1.5, Write-Protection Target Registers**. Write-protected registers are managed in units of protected registers called register protection clusters.

5.1.2 Write Procedure to Write-Protected Registers

Write access to a write-protected register is enabled by using the following protection unlock sequence:

1. Write the fixed value 0000 $00A5_H$ to the protection command register.
2. Write the desired value to the protected register.
3. Write the bit-wise inversion of the desired value to the protected register.
4. Write the desired value to the protected register.
5. Verify that the desired value has been written to the protected register.
   Verify successful write of the desired value to the protected register by verifying that the error monitor bit in the protection status register is “0”.
   In case the write was not successful, indicated by the error monitor bit set to “1”, the entire sequence has to be restarted at step 1.

If another register (second register) is accessed between step 1 and step 4 of the above sequence for writing to a write-protected register (first register), the protection mechanism operates as follows:

- If the second register belongs to the same cluster, the write to the protected register fails (the error monitor bit is set to 1). The entire sequence has to be restarted at step 1.
- If the second register does not belong to the same cluster, the protection unlock sequence is not disrupted and the write to the first register completes successfully.

5.1.3 Interrupt during Write Protection Unlock

If an interrupt occurs during the protection unlock sequence, the protection mechanism operates as follows:

(1) If an interrupt request is accepted during the protection unlock sequence and write access to a register of the same cluster is performed

The protection unlock sequence is disrupted, so the write operation to the protected register cannot be completed upon returning from the interrupt service routine. **Figure 5.1** shows an execution example.

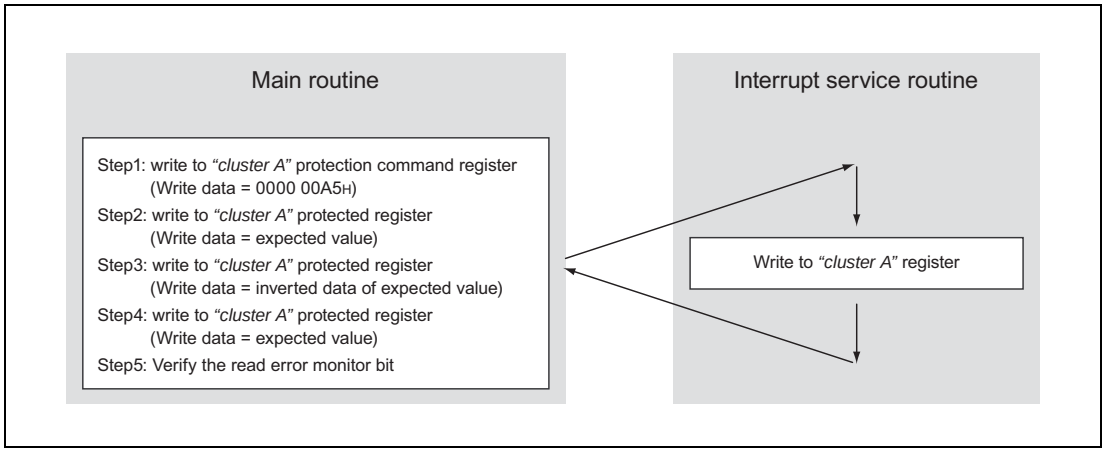


Figure 5.1 Example of Interruption of Register Protection Unlock Sequence

(2) If an interrupt request is accepted during the protection unlock sequence and write access to a register of a different cluster is performed

The protection unlock sequence is not disrupted, so the write operation to the protected register is completed upon return from the interrupt service routine. **Figure 5.2** shows an execution example.

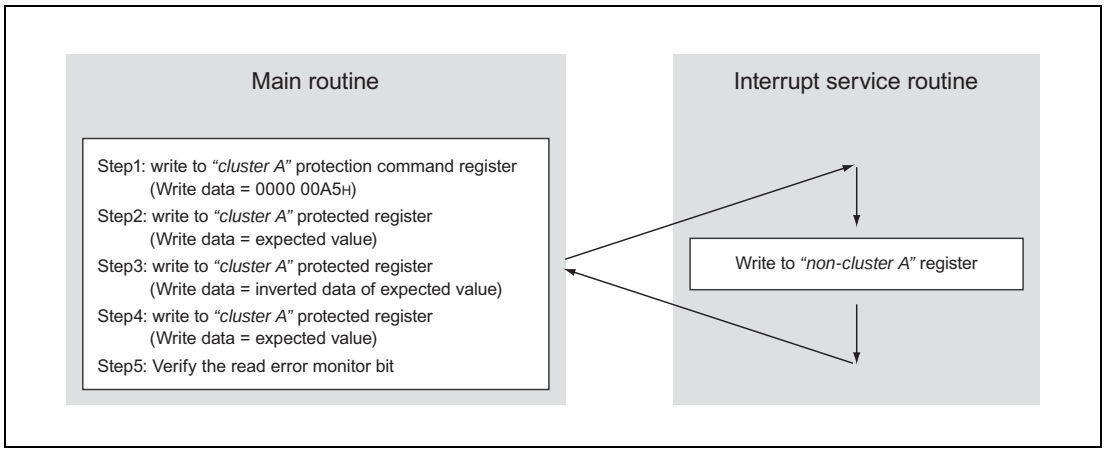


Figure 5.2 Example of Successful Protection Unlock Sequence

For more information on RH850/F1M register protection clusters, see **Section 5.1.5, Write-Protection Target Registers**.

5.1.4 Emulation Break during Write Protection Unlock Sequence

If an emulation break occurs during the protection unlock sequence, e.g. because of a breakpoint hit, the register protection is suspended until normal operation is resumed.
Even if any register of the same cluster is accessed during the break, the protection unlock sequence is not disrupted and the error monitor bit is not set to 1.

5.1.5 Write-Protection Target Registers

The registers that are protected through the write-protection control registers are listed below.

Table 5.1 Write-Protection Target Registers (1/2)

Protection Target	Protection Target Register	Protection Control Register		Protection Cluster
		Command Register	Status Register	
Clock controller	MOSCE	PROTCMD0	PROTS0	Control protection cluster 0
	ROSCE			
	SOSCE			
	CKSC_AWDTAD_CTL			
	CKSC_ATAUJS_CTL			
	CKSC_ATAUJD_CTL			
	CKSC_ARTCAS_CTL			
	CKSC_ARTCAD_CTL			
	CKSC_AADCAS_CTL			
	CKSC_AADCAD_CTL			
	CKSC_AFOUTS_CTL			
Stand-by function	STBC0PSC			
	STBC0STPT			
	IOHOLD			
Reset function	LVICNT			
	SWRESA			
Clock Controller	PLL0E	PROTCMD1	PROTS1	Control protection cluster 1
	PLL1E			
	CKSC_CPUCLKS_CTL			
	CKSC_CPUCLKD_CTL			
	CKSC_IPERI1S_CTL			
	CKSC_IPERI2S_CTL			
	CKSC_ILINS_CTL			
	CKSC_IADCAS_CTL			
	CKSC_IADCAD_CTL			
	CKSC_ILIND_CTL			
	CKSC_ICANS_CTL			
	CKSC_ICANOSCD_CTL			
	CKSC_ICSIS_CTL			
	CKSC_IIICS_CTL			

Table 5.1 Write-Protection Target Registers (2/2)

Protection Target	Protection Target Register	Protection Control Register		Protection Cluster
		Command Register	Status Register	
Clock Monitor	CLMA0CTL0	CLMA0PCMD	CLMA0PS	Clock Monitor control protection cluster 0
	CLMA1CTL0	CLMA1PCMD	CLMA1PS	Clock Monitor control protection cluster 1
	CLMA2CTL0	CLMA2PCMD	CLMA2PS	Clock Monitor control protection cluster 2
	CLMATEST	PROTCMDCLMA	PROTSCLMA	Clock Monitor test protection cluster
Port*1	JPODC0*2	JPPCMD0	JPPROTS0	Port protection cluster 0
	PODC0*2	PPCMD0	PPROTS0	
	PODC1*2	PPCMD1	PPROTS1	
	PODC2*2	PPCMD2	PPROTS2	
	PODC3*2	PPCMD3	PPROTS3	
	PODC8*2	PPCMD8	PPROTS8	
	PDSC0*2	PPCMD0	PPROTS0	
	PODC9	PPCMD9	PPROTS9	Port protection cluster 1
	PODC10	PPCMD10	PPROTS10	
	PODC11	PPCMD11	PPROTS11	
	PODC12	PPCMD12	PPROTS12	
	PODC13	PPCMD13	PPROTS13	
	PODC18	PPCMD18	PPROTS18	
	PODC19	PPCMD19	PPROTS19	
	PODC20	PPCMD20	PPROTS20	
	PDSC10	PPCMD10	PPROTS10	
	PDSC11	PPCMD11	PPROTS11	
	PDSC12	PPCMD12	PPROTS12	
	PDSC13	PPCMD13	PPROTS13	
	PDSC18	PPCMD18	PPROTS18	
	PDSC20	PPCMD20	PPROTS20	
Core Voltage Monitor	CVMF	PROTCMDCVM	PROTSCVM	Core Voltage Monitor protection cluster
	CVMDIAG			
Self-programming function	FLMDCNT	FLMDPCMD	FLMDPS	Self-programming protection cluster

Note 1. Each port group has its own protection command register and status register. For details, see **(1) Port protection clusters** on the next page.

Note 2. Follow the procedure below when writing to the PODCn (n = 0, 1, 2, 3, 8), JPODC0, or PDSC0 register of the port protection cluster 0 after returning from DeepSTOP mode (RESF.RESF10 = 1).

1. Write 0000 0000_H twice to the protection command register PPCMD0.
2. **Section 5.1.2, Write Procedure to Write-Protected Registers**.

Write sequence protection error by the protection status register PROTS0 should be checked during processing of the procedure 2. If attempting in the procedure 1, a write sequence protection error occurs.

(1) Port protection clusters

The following port control registers have write protection function:

- Port open drain control registers (PODCn, JPODC0)
- Port drive strength control registers (PDSCn)

The write protected port registers are divided into two port protection clusters as shown in the following table:

Table 5.2 Port Protection Clusters

Port Protection Cluster	Port Group
0	JP0, P0, P1, P2, P3, P8
1	P9, P10, P11, P12, P13, P18, P19, P20

NOTE

Each port group n has its own port protection command register PPCMDn and port protection status register PPROTSn.

However, any port protection command registers of the same port protection cluster can be used in the protection unlock sequence. For instance, PPCMD1 can be used to unlock the protection of PODC2.

5.2 Registers

5.2.1 List of Registers

The following table lists the write-protection control registers.

Table 5.3 List of Write-Protection Control Registers (1/2)

Register Name	Symbol	Address
Control protection clusters		
Protection command register 0	PROTCMD0	FFF8 0000$_H$
Protection command register 1	PROTCMD1	FFF8 8000$_H$
Protection status register 0	PROTS0	FFF8 0004$_H$
Protection status register 1	PROTS1	FFF8 8004$_H$
Clock monitor control and test protection cluster		
Protection command register 0	CLMA0PCMD	FFF8 C010$_H$
Protection command register 1	CLMA1PCMD	FFF8 D010$_H$
Protection command register 2	CLMA2PCMD	FFF8 E010$_H$
Protection status register 0	CLMA0PS	FFF8 C014$_H$
Protection status register 1	CLMA1PS	FFF8 D014$_H$
Protection status register 2	CLMA2PS	FFF8 E014$_H$
Protection command register	PROTCMDCLMA	FFF8 C200$_H$
Protection status register	PROTSCLMA	FFF8 C204$_H$
Port protection cluster 0		
Protection command registers	JPPCMD0	FFC2 04C0$_H$
	PPCMD0	FFC1 4C00$_H$
	PPCMD1	FFC1 4C04$_H$
	PPCMD2	FFC1 4C08$_H$
	PPCMD3	FFC1 4C0C$_H$
	PPCMD8	FFC1 4C20$_H$
Protection status registers	JPPROTS0	FFC2 04B0$_H$
	PPROTS0	FFC1 4B00$_H$
	PPROTS1	FFC1 4B04$_H$
	PPROTS2	FFC1 4B08$_H$
	PPROTS3	FFC1 4B0C$_H$
	PPROTS8	FFC1 4B20$_H$
Port protection cluster 1		
Protection command registers	PPCMD9	FFC1 4C24$_H$
	PPCMD10	FFC1 4C28$_H$
	PPCMD11	FFC1 4C2C$_H$
	PPCMD12	FFC1 4C30$_H$
	PPCMD13	FFC1 4C34$_H$
	PPCMD18	FFC1 4C48$_H$
	PPCMD19	FFC1 4C4C$_H$
	PPCMD20	FFC1 4C50$_H$

Table 5.3 List of Write-Protection Control Registers (2/2)

Register Name	Symbol	Address
Protection status registers	PPROTS9	FFC1 4B24$_H$
	PPROTS10	FFC1 4B28$_H$
	PPROTS11	FFC1 4B2C$_H$
	PPROTS12	FFC1 4B30$_H$
	PPROTS13	FFC1 4B34$_H$
	PPROTS18	FFC1 4B48$_H$
	PPROTS19	FFC1 4B4C$_H$
	PPROTS20	FFC1 4B50$_H$
Core Voltage Monitor protection cluster		
Protection command register	PROTCMDCVM	FFF8 3200$_H$
Protection status register	PROTSCVM	FFF8 3204$_H$
Self-programming protection cluster		
Protection command register	FLMDPCMD	FFA0 0004$_H$
Protection error status register	FLMDPS	FFA0 0008$_H$

5.2.2 Details of Control Protection Cluster Registers

5.2.2.1 PROTCMDn — Protection Command Register

This register is used to initiate the write protection unlock sequence for write-protected registers.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers**.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: XXXX $XX00_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PCMDn[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 5.4 PROTCMDn Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	PCMDn[7:0]	Protection command register bits to enable writing to protection target registers of control protection cluster

5.2.2.2 PROTSn — Protection Status Register

This register indicates the status of the protection unlock sequence performed by PROTCMDn.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: $0000\ 0000_H$
$XXXX\ XXXX_H$ (n = 0) upon return from DeepSTOP mode (RESF.RESF10 = 1)

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PROTSnERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 5.5 PROTSn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	PROTSnERR	Write sequence protection error monitor 0: No protection error occurred 1: Protection error occurred

5.2.3 Details of Clock Monitor Protection Cluster Registers

5.2.3.1 CLMAnPCMD — CLMAn Protection Command Register

This register is a protection command register for the CLMAnCTL0 register.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers**.

Access: This register is a write-only register that can be written in 8-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: 00_H

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

Table 5.6 CLMAnPCMD Register Contents

Bit Position	Bit Name	Function
7 to 0	CLMAnREG[7:0]	Protection command register bits to enable writing to the CLMAnCTL0 register

5.2.3.2 CLMAnPS — CLMAn Protection Status Register

This register is used to verify whether the write-protected register (CLMAnCTL0) has been successfully written or not.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers**.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: 00_H

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

Table 5.7 CLMAnPS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	CLMAnPRERR	Write Protection sequence error monitor 0: No protection error occurred 1: Protection error occurred

5.2.3.3 PROTCMDCLMA — Clock Monitor Test Protection Command Register

This register is a protection command register for the CLMATEST register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: XXXX $XX00_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLMATREG[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 5.8 PROTCMDCLMA Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write “0”.
7 to 0	CLMATREG[7:0]	Protection command register bits to enable writing to the CLMATEST register

5.2.3.4 PROTSCLMA — Clock Monitor Test Protection Status Register

This register is used to verify whether the write-protected register (CLMATEST) has been successfully written or not.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLMAT PRERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 5.9 PROTSCLMA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	CLMATPRERR	Protection write sequence error monitor 0: No protection error occurred 1: Protection error occurred

5.2.4 Details of Core Voltage Monitor Register

5.2.4.1 PROTCMDCVM — Core Voltage Monitor Protection Command Register

This register is a protection command register for the CVMF and CVMDIAG registers.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: XXXX XX00$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CVMFREG[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 5.10 PROTCMDCVM Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	CVMFREG[7:0]	Protection command register bits to enable writing to the CVMF and CVMDIAG registers

5.2.4.2 PROTSCVM — Core Voltage Monitor Protection Status Register

This register is used to verify whether the write-protected register (CVMF, CVMDIAG) has been successfully written or not.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CVMFPRERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 5.11 PROTSCVM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	CVMFPRERR	Protection write sequence error monitor 0: No protection error occurred 1: Protection error occurred

5.2.5 Details of Port Protection Cluster Registers

5.2.5.1 PPCMDn — Port Protection Command Register

PPCMDn is a protection command register for port group n.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers**.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: XXXX $XX00_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PPCMDn[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

NOTE

The protection command register for port group JP0 is JPPCMD0. Its bits are JPPCMD0[7:0].

Table 5.12 PPCMDn Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	PPCMDn[7:0]	Protection command register bits that enable writing to port protection cluster registers

5.2.5.2 PPROTSn — Port Protection Status Register

PPROTSn is a protection status register for write-protected registers of port group n. It indicates the status of the protection sequence operated by PPCMDn.

Index n

An index “n” denotes the number of protection command registers. For details, see **Table 5.1, Write-Protection Target Registers**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PPROTSnPRERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

NOTE

The protection status register for port group JP0 is JPPROTS0. Its bit is JPPROTS0PRERR.

Table 5.13 PPROTSn Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	PPROTSn PRERR	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

5.2.6 Details of Self-Programming Protection Cluster Registers

5.2.6.1 FLMDPCMD — FLMD Protection Command Register

FLMDPCMD is a protection command register for the FLMDCNT register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: XXXX XX00$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	FLMDPC[7:0]							
Value after reset	—	—	—	—	—	—	—	—	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 5.14 FLMDPCMD Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write 0.
7 to 0	FLMDPC[7:0]	Protection command register bits that enable writing to FLMDCNT register

5.2.6.2 FLMDPS — FLMD Protection Error Status Register

This register is used to verify whether the write-protected register (FLMDCNT) has been successfully written or not.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 5.3, List of Write-Protection Control Registers**.

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FLMDPRERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 5.15 FLMDPS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	FLMDPRERR	Protection write sequence error monitor 0: No protection error occurred 1: Protection error occurred

Section 6 Operating Mode

This section describes the operating mode and mode selection of the RH850/F1M.

The RH850/F1M has the operating modes shown below.

- Normal operating mode

  This mode is for execution of the user program. The on-chip debug functions also use this mode. If FLMD0 is pulled up high during operation in this mode, writing to the code flash memory through self-programming is enabled.

- Serial programming mode

  The dedicated flash memory programmer enables erasing/writing to flash memory.

- Boundary scan mode

  This mode allows boundary scan tests compliant with IEEE Standard 1149.1.

When an external reset or power-on-clear reset is generated, the state of the FLMD0, FLMD1, MODE0, and MODE1 pins are used to determine the operating mode after reset is released. The operating mode is fixed by the release of these reset factors. **Table 6.1** lists the relationship between the pin settings and the operating mode.

Table 6.1 Selection of Operating Mode

Pins				
FLMD0	FLMD1 (P10_8)	MODE0 (P10_1)	MODE1 (P10_2)	Operating Mode
0	x	x	x	Normal operating mode
1	0	x	x	Serial programming mode
1	1	0	1	Boundary scan mode
Other than the above				Setting prohibited

CAUTION

To change operating mode, restart from power-on clear reset. (Remove the power supply once and apply it again.) In the case of only by the external reset, some functions are not initialized after the mode transitions. For details of functions not initialized by the external reset, see Section 9.1.1, Reset Sources.

Section 7 Exceptions and Interrupts

7.1 Features

The act of branching from a currently running program to a different program in response to an event is called an exception. This microcontroller supports the following types of exceptions.

The details on exceptions are described in the *RH850G3M User's Manual: Software*.

Table 7.1 List of Exception Sources

Name	Symbol	Source	Priority	Saved to
Reset	RESET	Reset input	High	—
FE level non-maskable interrupt*1	FENMI	FENMI input		FE
System error exception	SYSERR	SYSERR input		FE
FE level maskable interrupt*1	FEINT	FEINT input		FE
Floating-point arithmetic exception (imprecise)	FPI	Execution of FPU instruction		EI
EI level maskable interrupt*1	EIINT	Interrupt controller		EI
Memory protection exception (execution right)	MIP	Memory protection violation		FE
System error exception	SYSERR	Error input at instruction fetch		FE
Reserved instruction exception	RIE	Execution of reserved instruction		FE
Coprocessor unusable exception	UCPOP	Execution of coprocessor instruction/ access right violation		FE
Privileged instruction exception	PIE	Execution of privileged instruction		FE
Misaligned exception	MAE	Generation of misaligned access		FE
Memory protection exception (access right)	MDP	Memory protection violation		FE
Floating-point arithmetic exception (precise)	FPP	Execution of FPU instruction		EI
System call	SYSCALL	Execution of SYSCALL instruction		EI
FE level trap	FETRAP	Execution of FETRAP instruction		FE
EI level trap 0	TRAP0	Execution of TRAP instruction		EI
EI level trap 1	TRAP1	Execution of TRAP instruction	Low	EI

Note 1. These interrupt exceptions are described in this section.

(1) Interrupt sources

The following three exceptions in **Table 7.1** are called interrupts, and are described in this section.

- FE level non-maskable interrupt (FENMI)
  An FENMI interrupt is acknowledged even if another FE level interrupt - FEINT - has been generated.
  - An FENMI interrupt is acknowledged even if the CPU system register PSW.NP = 1.
  - Return from an FENMI interrupt is not possible and neither is recovery.
- FE level maskable interrupt (FEINT)
  - FEINT can be acknowledged if the CPU system register PSW.NP = 0. It is masked if PSW.NP = 1.
  - Return from an FEINT interrupt is possible and so is recovery.
- EI level maskable interrupt (EIINT)
  An EIINT interrupt can be acknowledged if an FE level interrupt - FENMI or FEINT - has not been generated.
  - EIINT can be acknowledged if the CPU system register PSW.NP = 0.
    It is masked if PSW.NP = 1, EIINT with a higher priority is being processed, or PSW.ID = 1.
  - Return from an EIINT interrupt is possible and so is recovery.
  - Interrupt masking can be specified for each interrupt channel.
  - 16 interrupt priority levels can be specified for each interrupt channel.
  - In this section, the EIINT that corresponds to interrupt channel n is indicated by “INTn”, whereas the EIINT that corresponds to interrupt source xxx is indicated by “INTxxx”.

  For the PSW register, see **Table 3.10, PSW Register Contents** and the *RH850G3M User's Manual: Software*.

NOTE

Return: Indicates whether or not the program can resume from where it was interrupted.

Recovery: Indicates whether or not the processor status (status of processor resources including general purpose registers and system registers) can be restored to the status they were in when the program was interrupted.

These interrupt sources are described in **Section 7.2, RH850/F1M Interrupt Sources**.

(2) Overview of interrupts

- Priority levels for interrupt
  16 priority levels of maskable interrupts by request can be set by interrupt control register.
- Detecting methods of external interrupts (TNMI/INTPm)
  A method of detecting external interrupts (TNMI and INTPm) can be selected from five types: rising edge, falling edge, both edges, low level, and high level.
- 2 types of interrupt handler address setting Direct branching method or table referencing method is selectable by register setting.

7.2 RH850/F1M Interrupt Sources

7.2.1 Interrupt Sources

7.2.1.1 FE Level Non-Maskable Interrupts

(1) Priority

See **Table 7.1, List of Exception Sources**.

(2) Return PC

Return or recovery from an FE non-maskable interrupt is not possible.

(3) Status Register

See **Section 7.4.4, FNC — FE Level NMI Status Register**.

(4) Return Instruction

None

Table 7.2 FE Level Non-Maskable Interrupt Requests

Interrupt			Interrupt Request		Unit	Priority	Exception Source Code	Handler Address 00000...
	Control Register							
Symbol	Name	Address	Name	Source				
FENMI	FNC	FFFE EA78$_H$	TNMI	NMI pin	Port	*1	0E0$_H$	0E0$_H$
			WDTA0NMI	WDTA0 FENMI interrupt	WDTA0			
			WDTA1NMI	WDTA1 FENMI interrupt	WDTA1			

Note 1. See **Table 7.1, List of Exception Sources**.

The source of the FENMI interrupt can be evaluated by a dedicated flag register. See **Section 7.2.2, FE Level Non-Maskable Interrupt Sources** for details.

7.2.1.2 FE Level Maskable Interrupts

(1) Priority

See **Table 7.1, List of Exception Sources**.

(2) Return PC

The return PC returned from an interrupt handling routine by the FERET instruction is the PC from when the program was suspended (current PC).

(3) Status Register

See **Section 7.4.5, FIC — FE Level Maskable Interrupt Status Register**.

(4) Return Instruction

FERET

Table 7.3 FE Level Maskable Interrupt Requests

Interrupt			Interrupt Request					
	Control Register							
Symbol	Name	Address	Name	Source	Unit	Priority	Exception Source Code	Handler Address 00000...
FEINT	FIC	FFFE EA7A$_H$	INTLVIL	LVI voltage detection (falling)	LVI	*1	0F0$_H$	0F0$_H$
			INTECCDEEP0	ECC 2-bit error interrupt of data flash	Data flash			
			INTECCDFLRAM	ECC 2-bit error interrupt of FLXA0	FLXA0			
			INTECCDCNRAM0	ECC 2-bit error interrupt of RS-CAN	RSCAN0 (ch0-5)			
			INTECCDCSIH0	ECC 2-bit error interrupt of CSIH0	CSIH0			
			INTECCDCSIH1	ECC 2-bit error interrupt of CSIH1	CSIH1			
			INTECCDCSIH2	ECC 2-bit error interrupt of CSIH2	CSIH2			
			INTECCDCSIH3	ECC 2-bit error interrupt of CSIH3	CSIH3			
			INTECCSCFLI0	ECC 1-bit error correction interrupt of code flash	Code flash			
			INTECCRAM	ECC 1-bit error interrupt of RAM	RAM*5			
			INTOSTM0_FE*2	OSTM0 interrupt	OSTM0*3,*4			
			INTLVIH	LVI voltage detection (rising)	LVI			
			INTGUARD	Guard error detection*6	CPU			
			INTOSTM1_FE	OSTM1 interrupt	OSTM1			
			INTOSTM2_FE	OSTM2 interrupt	OSTM2			
			INTOSTM3_FE	OSTM3 interrupt	OSTM3			
			INTOSTM4_FE	OSTM4 interrupt	OSTM4			

Note 1. See **Table 7.1, List of Exception Sources**.

Note 2. As INTOSTM0 is also assigned to EIINT, using them simultaneously is prohibited. When INTOSTM0 is used as FEINT, it is generated by the TSU (timing supervision unit) function of OSTM0.

Note 3. If OSTM0 is used as the TSU function, use FEINT.

Note 4. If OSTM0 is used as other than the TSU function, use EIINT.

Note 5. “RAM” indicates the following areas of RAM.
Local RAM (CPU1), retention RAM, cache RAM, and cache TAG RAM

Note 6. A logical sum of the following errors:
PEG error, GRG error

* Timing monitor (TSU)
This prevents the illicit use of CPU time by non-trusted programs, manages properties, and controls the intervals over which interrupts are disabled.

7.2.1.3 EI Level Maskable Interrupts

(1) Interrupt Naming Rules

The composition of the interrupt request signal names, their assigned interrupt control registers and the bits in these registers follow special rules.

In the following the name of the specific interrupt request is represented by **<*name*>**.

For details of the names used in **IC<*name*>**, see **Table 7.4**.

- Interrupt request name: **INT<*name*>**
  The prefix **"INT"** is appended to **<*name*>**.
- Interrupt control register: **IC<*name*>**
  The prefix **"IC"** is appended to **<*name*>**.
  The 16-bit register **IC<*name*>** can also be accessed in byte units:
  - Low byte (bits [7:0]) of the **IC<*name*>** register: **IC<*name*>L**
    The suffix **"L"** is appended to the register name **IC<*name*>**.
  - High byte (bits [15:8]) of the **IC<*name*>** register: **IC<*name*>H**
    The suffix **"H"** is appended to the register name **IC<*name*>**.
- Interrupt control register bit names: **CT<*name*>**, **RF<*name*>**, **MK<*name*>**, **TB<*name*>**, **P3<*name*>**, **P2<*name*>**, **P1<*name*>**, **P0<*name*>**
  The bit prefix "**CT**", "**RF**", "**MK**", "**TB**", "**P3**", "**P2**", "**P1**", or "**P0**" is appended to the interrupt **<*name*>**.

Example

The interrupt request from channel 2 of TAUD0 channel (**<*name*>** = *TAUD0I2*) is named

INT*TAUD0I2*

The related interrupt control registers are

IC*TAUD0I2*, **IC***TAUD0I2L*, **IC***TAUD0I2H*

The bits in this register are

CT*TAUD0I2*,**RF***TAUD0I2*, **MK***TAUD0I2*, **TB***TAUD0I2*, **P3***TAUD0I2*,**P2***TAUD0I2*, **P1***TAUD0I2*, **P0***TAUD0I2*

(2) Priority

See **Table 7.1, List of Exception Sources**.

(3) Return PC

The return PC returned from an interrupt handling routine by the EIRET instruction is the PC from when the program was suspended (current PC).

(4) Control Register

EI level maskable interrupt control register

See **Section 7.4.2, ICxxx — EI Level Interrupt Control Registers**.

(5) Return Instruction

EIRET instruction

(6) Configuration

EI-level maskable interrupts are controlled by the two controllers, INTC1 and INTC2.

The interrupts are supported on a total of 298 channels with a cascade connection of INTC1 and INTC2.

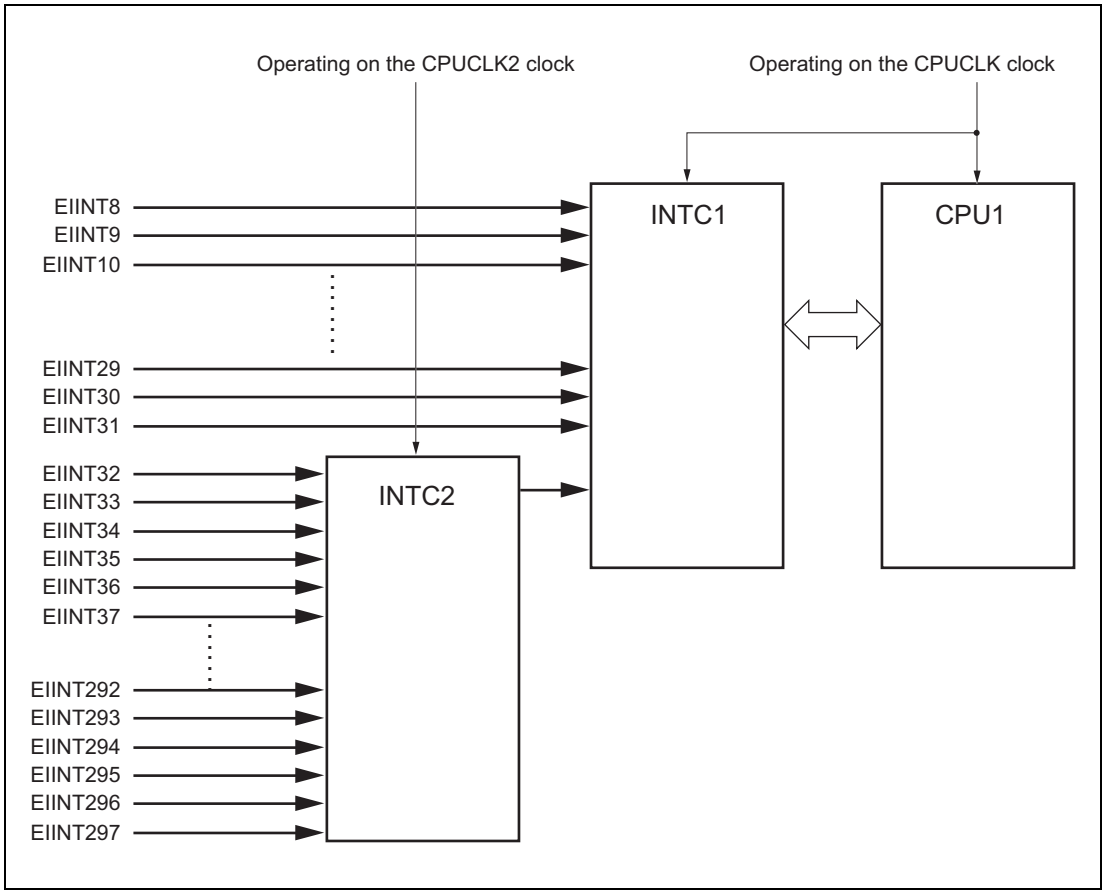


Figure 7.1 Configuration Diagram of EI Level Maskable Interrupt

CAUTION

As CPUCLK2 is the operating clock for INTC2, the EIINT32 to EIINT297 interrupts, which are connected to INTC2, are delayed more than the interrupts directly connected to INTC1.

Table 7.4 lists EI level maskable interrupts.

Table 7.4 EI Level Maskable Interrupt Sources (1/8)

Interrupt			Interrupt Request				Exception Source Code	144 pins	176 pins	233 pins	Handler Address (Offset)*8		
											Direct Jumping to an Address		
Channel No.*1	Control Register										RINT = 0	RINT = 1	Reference to a Table*5
	Name	Address	Name	Source	Unit	Detection Type*2							
0	Reserved	FFFE EA00$_H$					1000$_H$	—	—	—	*3	*4	+000$_H$
1	Reserved	FFFE EA02$_H$					1001$_H$	—	—	—			+004$_H$
2	Reserved	FFFE EA04$_H$					1002$_H$	—	—	—			+008$_H$
3	Reserved	FFFE EA06$_H$					1003$_H$	—	—	—			+00C$_H$
4	Reserved	FFFE EA08$_H$					1004$_H$	—	—	—			+010$_H$
5	Reserved	FFFE EA0A$_H$					1005$_H$	—	—	—			+014$_H$
6	Reserved	FFFE EA0C$_H$					1006$_H$	—	—	—			+018$_H$
7	Reserved	FFFE EA0E$_H$					1007$_H$	—	—	—			+01C$_H$
8	ICTAUD0I0	FFFE EA10$_H$	INTTAUD0I0*6	Interrupt for TAUD0 channel 0	TAUD0	Edge	1008$_H$	√	√	√			+020$_H$
	ICCSIH2IC_1		INTCSIH2IC_1*6	CSIH2 communication status interrupt	CSIH2	Edge							
9	ICTAUD0I2	FFFE EA12$_H$	INTTAUD0I2*6	Interrupt for TAUD0 channel 2	TAUD0	Edge	1009$_H$	√	√	√			+024$_H$
	ICCSIH3IC_1		INTCSIH3IC_1*6	CSIH3 communication status interrupt	CSIH3	Edge							
10	ICTAUD0I4	FFFE EA14$_H$	INTTAUD0I4*6	Interrupt for TAUD0 channel 4	TAUD0	Edge	100A$_H$	√	√	√			+028$_H$
	ICCSIH2IR_1		INTCSIH2IR_1*6	CSIH2 receive status interrupt	CSIH2	Edge							
11	ICTAUD0I6	FFFE EA16$_H$	INTTAUD0I6*6	Interrupt for TAUD0 channel 6	TAUD0	Edge	100B$_H$	√	√	√			+02C$_H$
	ICCSIH2IRE_1		INTCSIH2IRE_1*6	CSIH2 communication error interrupt	CSIH2	Edge							
12	ICTAUD0I8	FFFE EA18$_H$	INTTAUD0I8*6	Interrupt for TAUD0 channel 8	TAUD0	Edge	100C$_H$	√	√	√			+030$_H$
	ICCSIH2IJC_1		INTCSIH2IJC_1*6	CSIH2 job completion interrupt	CSIH2	Edge							
13	ICTAUD0I10	FFFE EA1A$_H$	INTTAUD0I10*6	Interrupt for TAUD0 channel 10	TAUD0	Edge	100D$_H$	√	√	√			+034$_H$
	ICCSIH3IR_1		INTCSIH3IR_1*6	CSIH3 receive status interrupt	CSIH3	Edge							
14	ICTAUD0I12	FFFE EA1C$_H$	INTTAUD0I12*6	Interrupt for TAUD0 channel 12	TAUD0	Edge	100E$_H$	√	√	√			+038$_H$
	ICCSIH3IRE_1		INTCSIH3IRE_1*6	CSIH3 communication error interrupt	CSIH3	Edge							
15	ICTAUD0I14	FFFE EA1E$_H$	INTTAUD0I14*6	Interrupt for TAUD0 channel 14	TAUD0	Edge	100F$_H$	√	√	√			+03C$_H$
	ICCSIH3IJC_1		INTCSIH3IJC_1*6	CSIH3 job completion interrupt	CSIH3	Edge							
16	ICTAPA0IPEK0	FFFE EA20$_H$	INTTAPA0IPEK0*6	TAPA0 peak interrupt 0	TAPA0	Edge	1010$_H$	√	√	√			+040$_H$
	ICCSIH1IC_1		INTCSIH1IC_1*6	CSIH1 communication status interrupt	CSIH1	Edge							
17	ICTAPA0IVLY0	FFFE EA22$_H$	INTTAPA0IVLY0*6	TAPA0 valley interrupt 0	TAPA0	Edge	1011$_H$	√	√	√			+044$_H$
	ICCSIH1IR_1		INTCSIH1IR_1*6	CSIH1 receive status interrupt	CSIH1	Edge							
18	ICADCA0I0	FFFE EA24$_H$	INTADCA0I0	ADCA0 scan group 1 (SG1) end interrupt	ADCA0	Edge	1012$_H$	√	√	√			+048$_H$
19	ICADCA0I1	FFFE EA26$_H$	INTADCA0I1	ADCA0 scan group 2 (SG2) end interrupt	ADCA0	Edge	1013$_H$	√	√	√			+04C$_H$
20	ICADCA0I2	FFFE EA28$_H$	INTADCA0I2*6	ADCA0 scan group 3 (SG3) end interrupt	ADCA0	Edge	1014$_H$	√	√	√			+050$_H$
	ICCSIH0IJC_1		INTCSIH0IJC_1*6	CSIH0 job completion interrupt	CSIH0	Edge							
21	ICDCUTDI	FFFE EA2A$_H$	INTDCUTDI	Dedicated interrupt for on-chip debug function	Port	Edge	1015$_H$	√	√	√			+054$_H$
22	ICRCANGERR0	FFFE EA2C$_H$	INTRCANGERR0	CAN global error interrupt	RS-CAN0	Level	1016$_H$	√	√	√			+058$_H$
23	ICRCANGRECC0	FFFE EA2E$_H$	INTRCANGRECC0	CAN receive FIFO interrupt	RS-CAN0	Level	1017$_H$	√	√	√			+05C$_H$
24	ICRCAN0ERR	FFFE EA30$_H$	INTRCAN0ERR	CAN0 error interrupt	RS-CAN0	Level	1018$_H$	√	√	√			+060$_H$
25	ICRCAN0REC	FFFE EA32$_H$	INTRCAN0REC	CAN0 transmit/receive FIFO receive complete interrupt	RS-CAN0	Level	1019$_H$	√	√	√			+064$_H$
26	ICRCAN0TRX	FFFE EA34$_H$	INTRCAN0TRX	CAN0 transmit interrupt	RS-CAN0	Level	101A$_H$	√	√	√			+068$_H$
27	ICCSIG0IC	FFFE EA36$_H$	INTCSIG0IC*6	CSIG0 communication status interrupt	CSIG0	Edge	101B$_H$	√	√	√			+06C$_H$
	ICCSIH1IRE_1		INTCSIH1IRE_1*6	CSIH1 communication error interrupt	CSIH1	Edge							
28	ICCSIG0IR	FFFE EA38$_H$	INTCSIG0IR*6	CSIG0 receive status interrupt	CSIG0	Edge	101C$_H$	√	√	√			+070$_H$
	ICCSIH1IJC_1		INTCSIH1IJC_1*6	CSIH1 job interrupt	CSIH1	Edge							

Table 7.4 EI Level Maskable Interrupt Sources (2/8)

Interrupt			Interrupt Request				Exception Source Code	144 pins	176 pins	233 pins	Handler Address (Offset)*8		
											Direct Jumping to an Address		
Channel No.*1	Control Register												
	Name	Address	Name	Source	Unit	Detection Type*2					RINT = 0	RINT = 1	Reference to a Table*5
29	ICCSIH0IC	FFFE EA3A$_H$	INTCSIH0IC	CSIH0 communication status interrupt	CSIH0	Edge	101D$_H$	√	√	√	*3	*4	+074$_H$
30	ICCSIH0IR	FFFE EA3C$_H$	INTCSIH0IR	CSIH0 receive status interrupt	CSIH0	Edge	101E$_H$	√	√	√			+078$_H$
31	ICCSIH0IRE	FFFE EA3E$_H$	INTCSIH0IRE	CSIH0 communication error interrupt	CSIH0	Edge	101F$_H$	√	√	√			+07C$_H$
32	ICCSIH0IJC	FFFF B040$_H$	INTCSIH0IJC*6	CSIH0 job completion interrupt	CSIH0	Edge	1020$_H$	√	√	√			+080$_H$
	ICADCA0I2_2		INTADCA0I2_2*6	ADCA0 scan group 3 (SG3) end interrupt	ADCA0	Edge							
33	ICRLIN30	FFFF B042$_H$	INTRLIN30	RLIN30 interrupt	RLIN30	Edge	1021$_H$	√	√	√			+084$_H$
34	ICRLIN30UR0	FFFF B044$_H$	INTRLIN30UR0	RLIN30 transmit interrupt	RLIN30	Edge	1022$_H$	√	√	√			+088$_H$
35	ICRLIN30UR1	FFFF B046$_H$	INTRLIN30UR1	RLIN30 receive complete interrupt	RLIN30	Edge	1023$_H$	√	√	√			+08C$_H$
36	ICRLIN30UR2	FFFF B048$_H$	INTRLIN30UR2	RLIN30 status interrupt	RLIN30	Edge	1024$_H$	√	√	√			+090$_H$
37	ICP0	FFFF B04A$_H$	INTP0	External interrupt	Port	Edge	1025$_H$	√	√	√			+094$_H$
38	ICP1	FFFF B04C$_H$	INTP1	External interrupt	Port	Edge	1026$_H$	√	√	√			+098$_H$
39	ICP2	FFFF B04E$_H$	INTP2	External interrupt	Port	Edge	1027$_H$	√	√	√			+09C$_H$
40	ICWDTA0	FFFF B050$_H$	INTWDTA0	WDTA0 75% interrupt	WDTA0	Edge	1028$_H$	√	√	√			+0A0$_H$
41	ICWDTA1	FFFF B052$_H$	INTWDTA1	WDTA1 75% interrupt	WDTA1	Edge	1029$_H$	√	√	√			+0A4$_H$
42	Reserved	FFFF B054$_H$					102A$_H$	—	—	—			+0A8$_H$
43	ICP3	FFFF B056$_H$	INTP3	External interrupt	Port	Edge	102B$_H$	√	√	√			+0AC$_H$
44	ICP4	FFFF B058$_H$	INTP4	External interrupt	Port	Edge	102C$_H$	√	√	√			+0B0$_H$
45	ICP5	FFFF B05A$_H$	INTP5	External interrupt	Port	Edge	102D$_H$	√	√	√			+0B4$_H$
46	ICP10	FFFF B05C$_H$	INTP10	External interrupt	Port	Edge	102E$_H$	√	√	√			+0B8$_H$
47	ICP11	FFFF B05E$_H$	INTP11	External interrupt	Port	Edge	102F$_H$	√	√	√			+0BC$_H$
48	ICTAUD0I1	FFFF B060$_H$	INTTAUD0I1	Interrupt for TAUD0 channel 1	TAUD0	Edge	1030$_H$	√	√	√			+0C0$_H$
49	ICTAUD0I3	FFFF B062$_H$	INTTAUD0I3	Interrupt for TAUD0 channel 3	TAUD0	Edge	1031$_H$	√	√	√			+0C4$_H$
50	ICTAUD0I5	FFFF B064$_H$	INTTAUD0I5	Interrupt for TAUD0 channel 5	TAUD0	Edge	1032$_H$	√	√	√			+0C8$_H$
51	ICTAUD0I7	FFFF B066$_H$	INTTAUD0I7	Interrupt for TAUD0 channel 7	TAUD0	Edge	1033$_H$	√	√	√			+0CC$_H$
52	ICTAUD0I9	FFFF B068$_H$	INTTAUD0I9	Interrupt for TAUD0 channel 9	TAUD0	Edge	1034$_H$	√	√	√			+0D0$_H$
53	ICTAUD0I11	FFFF B06A$_H$	INTTAUD0I11	Interrupt for TAUD0 channel 11	TAUD0	Edge	1035$_H$	√	√	√			+0D4$_H$
54	ICTAUD0I13	FFFF B06C$_H$	INTTAUD0I13	Interrupt for TAUD0 channel 13	TAUD0	Edge	1036$_H$	√	√	√			+0D8$_H$
55	ICTAUD0I15	FFFF B06E$_H$	INTTAUD0I15	Interrupt for TAUD0 channel 15	TAUD0	Edge	1037$_H$	√	√	√			+0DC$_H$
56	ICADCA0ERR	FFFF B070$_H$	INTADCA0ERR	ADCA0 error interrupt	ADCA0	Edge	1038$_H$	√	√	√			+0E0$_H$
57	ICCSIG0IRE	FFFF B072$_H$	INTCSIG0IRE	CSIG0 communication error interrupt	CSIG0	Edge	1039$_H$	√	√	√			+0E4$_H$
58	ICRLIN20	FFFF B074$_H$	INTRLIN20	RLIN20 interrupt	RLIN240	Edge	103A$_H$	√	√	√			+0E8$_H$
59	ICRLIN21	FFFF B076$_H$	INTRLIN21	RLIN21 interrupt	RLIN240	Edge	103B$_H$	√	√	√			+0EC$_H$
60	ICDMA0	FFFF B078$_H$	INTDMA0	DMA0 transfer completion	DMAC	Edge	103C$_H$	√	√	√			+0F0$_H$
61	ICDMA1	FFFF B07A$_H$	INTDMA1	DMA1 transfer completion	DMAC	Edge	103D$_H$	√	√	√			+0F4$_H$
62	ICDMA2	FFFF B07C$_H$	INTDMA2	DMA2 transfer completion	DMAC	Edge	103E$_H$	√	√	√			+0F8$_H$
63	ICDMA3	FFFF B07E$_H$	INTDMA3	DMA3 transfer completion	DMAC	Edge	103F$_H$	√	√	√			+0FC$_H$
64	ICDMA4	FFFF B080$_H$	INTDMA4	DMA4 transfer completion	DMAC	Edge	1040$_H$	√	√	√			+100$_H$
65	ICDMA5	FFFF B082$_H$	INTDMA5	DMA5 transfer completion	DMAC	Edge	1041$_H$	√	√	√			+104$_H$
66	ICDMA6	FFFF B084$_H$	INTDMA6	DMA6 transfer completion	DMAC	Edge	1042$_H$	√	√	√			+108$_H$
67	ICDMA7	FFFF B086$_H$	INTDMA7	DMA7 transfer completion	DMAC	Edge	1043$_H$	√	√	√			+10C$_H$
68	ICDMA8	FFFF B088$_H$	INTDMA8	DMA8 transfer completion	DMAC	Edge	1044$_H$	√	√	√			+110$_H$
69	ICDMA9	FFFF B08A$_H$	INTDMA9	DMA9 transfer completion	DMAC	Edge	1045$_H$	√	√	√			+114$_H$

Table 7.4 EI Level Maskable Interrupt Sources (3/8)

Interrupt			Interrupt Request				Exception Source Code	144 pins	176 pins	233 pins	Handler Address (Offset)*8		
											Direct Jumping to an Address		
Channel No.*1	Control Register												
	Name	Address	Name	Source	Unit	Detection Type*2					RINT = 0	RINT = 1	Reference to a Table*5
70	ICDMA10	FFFF B08C$_H$	INTDMA10	DMA10 transfer completion	DMAC	Edge	1046$_H$	√	√	√	*3	*4	+118$_H$
71	ICDMA11	FFFF B08E$_H$	INTDMA11	DMA11 transfer completion	DMAC	Edge	1047$_H$	√	√	√			+11C$_H$
72	ICDMA12	FFFF B090$_H$	INTDMA12	DMA12 transfer completion	DMAC	Edge	1048$_H$	√	√	√			+120$_H$
73	ICDMA13	FFFF B092$_H$	INTDMA13	DMA13 transfer completion	DMAC	Edge	1049$_H$	√	√	√			+124$_H$
74	ICDMA14	FFFF B094$_H$	INTDMA14	DMA14 transfer completion	DMAC	Edge	104A$_H$	√	√	√			+128$_H$
75	ICDMA15	FFFF B096$_H$	INTDMA15	DMA15 transfer completion	DMAC	Edge	104B$_H$	√	√	√			+12C$_H$
76	ICRIIC0TI	FFFF B098$_H$	INTRIIC0TI	RIIC transmit data empty interrupt	RIIC0	Edge	104C$_H$	√	√	√			+130$_H$
77	ICRIIC0TEI	FFFF B09A$_H$	INTRIIC0TEI	RIIC transmit complete interrupt	RIIC0	Level	104D$_H$	√	√	√			+134$_H$
78	ICRIIC0RI	FFFF B09C$_H$	INTRIIC0RI	RIIC receive complete interrupt	RIIC0	Edge	104E$_H$	√	√	√			+138$_H$
79	ICRIIC0EE	FFFF B09E$_H$	INTRIIC0EE	RIIC communication error/event interrupt	RIIC0	Level	104F$_H$	√	√	√			+13C$_H$
80	ICTAUJ0I0	FFFF B0A0$_H$	INTTAUJ0I0	Interrupt for TAUJ0 channel 0	TAUJ0	Edge	1050$_H$	√	√	√			+140$_H$
81	ICTAUJ0I1	FFFF B0A2$_H$	INTTAUJ0I1	Interrupt for TAUJ0 channel 1	TAUJ0	Edge	1051$_H$	√	√	√			+144$_H$
82	ICTAUJ0I2	FFFF B0A4$_H$	INTTAUJ0I2	Interrupt for TAUJ0 channel 2	TAUJ0	Edge	1052$_H$	√	√	√			+148$_H$
83	ICTAUJ0I3	FFFF B0A6$_H$	INTTAUJ0I3	Interrupt for TAUJ0 channel 3	TAUJ0	Edge	1053$_H$	√	√	√			+14C$_H$
84	ICOSTM0	FFFF B0A8$_H$	INTOSTM0*9	OSTM0 interrupt	OSTM0	Edge	1054$_H$	√	√	√			+150$_H$
85	ICENCA0IOV	FFFF B0AA$_H$	INTENCA0IOV*7	ENCA0 overflow interrupt	ENCA0	Edge	1055$_H$	√	√	√			+154$_H$
	ICPWGA4		INTPWGA4*7	PWGA4 interrupt	PWGA4	Edge							
86	ICENCA0IUD	FFFF B0AC$_H$	INTENCA0IUD*7	ENCA0 underflow interrupt	ENCA0	Edge	1056$_H$	√	√	√			+158$_H$
	ICPWGA5		INTPWGA5*7	PWGA5 interrupt	PWGA5	Edge							
87	ICENCA0I0	FFFF B0AE$_H$	INTENCA0I0*7	ENCA0 match/capture interrupt 0	ENCA0	Edge	1057$_H$	√	√	√			+15C$_H$
	ICPWGA6		INTPWGA6*7	PWGA6 interrupt	PWGA6	Edge							
88	ICENCA0I1	FFFF B0B0$_H$	INTENCA0I1*7	ENCA0 match/capture interrupt 1	ENCA0	Edge	1058$_H$	√	√	√			+160$_H$
	ICPWGA7		INTPWGA7*7	PWGA7 interrupt	PWGA7	Edge							
89	ICENCA0IEC	FFFF B0B2$_H$	INTENCA0IEC	ENCA0 encoder clear interrupt	ENCA0	Edge	1059$_H$	√	√	√			+164$_H$
90	ICKR0	FFFF B0B4$_H$	INTKR0	KR0 key interrupt	KR0	Edge	105A$_H$	√	√	√			+168$_H$
91	ICQFULL	FFFF B0B6$_H$	INTQFULL	PWSA queue full interrupt	PWSA	Edge	105B$_H$	√	√	√			+16C$_H$
92	ICPWGA0	FFFF B0B8$_H$	INTPWGA0	PWGA0 interrupt	PWGA0	Edge	105C$_H$	√	√	√			+170$_H$
93	ICPWGA1	FFFF B0BA$_H$	INTPWGA1	PWGA1 interrupt	PWGA1	Edge	105D$_H$	√	√	√			+174$_H$
94	ICPWGA2	FFFF B0BC$_H$	INTPWGA2	PWGA2 interrupt	PWGA2	Edge	105E$_H$	√	√	√			+178$_H$
95	ICPWGA3	FFFF B0BE$_H$	INTPWGA3	PWGA3 interrupt	PWGA3	Edge	105F$_H$	√	√	√			+17C$_H$
96	ICPWGA8	FFFF B0C0$_H$	INTPWGA8	PWGA8 interrupt	PWGA8	Edge	1060$_H$	√	√	√			+180$_H$
97	ICPWGA9	FFFF B0C2$_H$	INTPWGA9	PWGA9 interrupt	PWGA9	Edge	1061$_H$	√	√	√			+184$_H$
98	ICPWGA10	FFFF B0C4$_H$	INTPWGA10	PWGA10 interrupt	PWGA10	Edge	1062$_H$	√	√	√			+188$_H$
99	ICPWGA11	FFFF B0C6$_H$	INTPWGA11	PWGA11 interrupt	PWGA11	Edge	1063$_H$	√	√	√			+18C$_H$
100	ICPWGA12	FFFF B0C8$_H$	INTPWGA12	PWGA12 interrupt	PWGA12	Edge	1064$_H$	√	√	√			+190$_H$
101	ICPWGA13	FFFF B0CA$_H$	INTPWGA13	PWGA13 interrupt	PWGA13	Edge	1065$_H$	√	√	√			+194$_H$
102	ICPWGA14	FFFF B0CC$_H$	INTPWGA14	PWGA14 interrupt	PWGA14	Edge	1066$_H$	√	√	√			+198$_H$
103	ICPWGA15	FFFF B0CE$_H$	INTPWGA15	PWGA15 interrupt	PWGA15	Edge	1067$_H$	√	√	√			+19C$_H$
104	Reserved	FFFF B0D0$_H$					1068$_H$	—	—	—			+1A0$_H$
105	Reserved	FFFF B0D2$_H$					1069$_H$	—	—	—			+1A4$_H$
106	Reserved	FFFF B0D4$_H$					106A$_H$	—	—	—			+1A8$_H$
107	Reserved	FFFF B0D6$_H$					106B$_H$	—	—	—			+1AC$_H$
108	Reserved	FFFF B0D8$_H$					106C$_H$	—	—	—			+1B0$_H$
109	Reserved	FFFF B0DA$_H$					106D$_H$	—	—	—			+1B4$_H$
110	ICFLERR	FFFF B0DC$_H$	INTFLERR	Flash sequencer end error interrupt*10	FACI	Level	106E$_H$	√	√	√			+1B8$_H$
111	ICFLENDNM	FFFF B0DE$_H$	INTFLENDNM	Flash sequencer end interrupt*10	FACI	Edge	106F$_H$	√	√	√			+1BC$_H$

Table 7.4 EI Level Maskable Interrupt Sources (4/8)

Interrupt			Interrupt Request				Exception Source Code	144 pins	176 pins	233 pins	Handler Address (Offset)*8		
											Direct Jumping to an Address		
Channel No.*1	Control Register												
	Name	Address	Name	Source	Unit	Detection Type*2					RINT = 0	RINT = 1	Reference to a Table*5
112	ICCWEND	FFFF B0E0$_H$	INTCWEND	LPS0 port polling end interrupt	LPS	Edge	1070$_H$	√	√	√	*3	*4	+1C0$_H$
113	ICRCAN1ERR	FFFF B0E2$_H$	INTRCAN1ERR	CAN1 error interrupt	RS-CAN0	Level	1071$_H$	√	√	√			+1C4$_H$
114	ICRCAN1REC	FFFF B0E4$_H$	INTRCAN1REC	CAN1 transmit/receive FIFO receive complete interrupt	RS-CAN0	Level	1072$_H$	√	√	√			+1C8$_H$
115	ICRCAN1TRX	FFFF B0E6$_H$	INTRCAN1TRX	CAN1 transmit interrupt	RS-CAN0	Level	1073$_H$	√	√	√			+1CC$_H$
116	ICCSIH1IC	FFFF B0E8$_H$	INTCSIH1IC*6	CSIH1 communication status interrupt	CSIH1	Edge	1074$_H$	√	√	√			+1D0$_H$
	ICTAPA0IPEK0_2		INTTAPA0IPEK0_2*6	TAPA0 peak interrupt 0	TAPA0	Edge							
117	ICCSIH1IR	FFFF B0EA$_H$	INTCSIH1IR*6	CSIH1 receive status interrupt	CSIH1	Edge	1075$_H$	√	√	√			+1D4$_H$
	ICTAPA0IVLY0_2		INTTAPA0IVLY0_2*6	TAPA0 valley interrupt 0	TAPA0	Edge							
118	ICCSIH1IRE	FFFF B0EC$_H$	INTCSIH1IRE*6	CSIH1 communication error interrupt	CSIH1	Edge	1076$_H$	√	√	√			+1D8$_H$
	ICCSIG0IC_2		INTCSIG0IC_2*6	CSIG0 communication status interrupt	CSIG0	Edge							
119	ICCSIH1IJC	FFFF B0EE$_H$	INTCSIH1IJC*6	CSIH1 job interrupt	CSIH1	Edge	1077$_H$	√	√	√			+1DC$_H$
	ICCSIG0IR_2		INTCSIG0IR_2*6	CSIG0 receive status interrupt	CSIG0	Edge							
120	ICRLIN31	FFFF B0F0$_H$	INTRLIN31	RLIN31 interrupt	RLIN31	Edge	1078$_H$	√	√	√			+1E0$_H$
121	ICRLIN31UR0	FFFF B0F2$_H$	INTRLIN31UR0	RLIN31 transmit interrupt	RLIN31	Edge	1079$_H$	√	√	√			+1E4$_H$
122	ICRLIN31UR1	FFFF B0F4$_H$	INTRLIN31UR1	RLIN31 receive complete interrupt	RLIN31	Edge	107A$_H$	√	√	√			+1E8$_H$
123	ICRLIN31UR2	FFFF B0F6$_H$	INTRLIN31UR2	RLIN31 status interrupt	RLIN31	Edge	107B$_H$	√	√	√			+1EC$_H$
124	ICPWGA20	FFFF B0F8$_H$	INTPWGA20	PWGA20 interrupt	PWGA20	Edge	107C$_H$	√	√	√			+1F0$_H$
125	ICPWGA21	FFFF B0FA$_H$	INTPWGA21	PWGA21 interrupt	PWGA21	Edge	107D$_H$	√	√	√			+1F4$_H$
126	ICPWGA22	FFFF B0FC$_H$	INTPWGA22	PWGA22 interrupt	PWGA22	Edge	107E$_H$	√	√	√			+1F8$_H$
127	ICPWGA23	FFFF B0FE$_H$	INTPWGA23	PWGA23 interrupt	PWGA23	Edge	107F$_H$	√	√	√			+1FC$_H$
128	ICP6	FFFF B100$_H$	INTP6	External interrupt	Port	Edge	1080$_H$	√	√	√			+200$_H$
129	ICP7	FFFF B102$_H$	INTP7	External interrupt	Port	Edge	1081$_H$	√	√	√			+204$_H$
130	ICP8	FFFF B104$_H$	INTP8	External interrupt	Port	Edge	1082$_H$	√	√	√			+208$_H$
131	ICP12	FFFF B106$_H$	INTP12	External interrupt	Port	Edge	1083$_H$	√	√	√			+20C$_H$
132	ICCSIH2IC	FFFF B108$_H$	INTCSIH2IC*6	CSIH2 communication status interrupt	CSIH2	Edge	1084$_H$	√	√	√			+210$_H$
	ICTAUD0I0_2		INTTAUD0I0_2*6	Interrupt for TAUD0 channel 0	TAUD0	Edge							
133	ICCSIH2IR	FFFF B10A$_H$	INTCSIH2IR*6	CSIH2 communication status interrupt	CSIH2	Edge	1085$_H$	√	√	√			+214$_H$
	ICTAUD0I4_2		INTTAUD0I4_2*6	Interrupt for TAUD0 channel 4	TAUD0	Edge							
134	ICCSIH2IRE	FFFF B10C$_H$	INTCSIH2IRE*6	CSIH2 communication error interrupt	CSIH2	Edge	1086$_H$	√	√	√			+218$_H$
	ICTAUD0I6_2		INTTAUD0I6_2*6	Interrupt for TAUD0 channel 6	TAUD0	Edge							
135	ICCSIH2IJC	FFFF B10E$_H$	INTCSIH2IJC*6	CSIH2 job completion interrupt	CSIH2	Edge	1087$_H$	√	√	√			+21C$_H$
	ICTAUD0I8_2		INTTAUD0I8_2*6	Interrupt for TAUD0 channel 8	TAUD0	Edge							
136	Reserved	FFFF B110$_H$					1088$_H$	—	—	—			+220$_H$
137	Reserved	FFFF B112$_H$					1089$_H$	—	—	—			+224$_H$
138	Reserved	FFFF B114$_H$					108A$_H$	—	—	—			+228$_H$
139	Reserved	FFFF B116$_H$					108B$_H$	—	—	—			+22C$_H$
140	Reserved	FFFF B118$_H$					108C$_H$	—	—	—			+230$_H$
141	Reserved	FFFF B11A$_H$					108D$_H$	—	—	—			+234$_H$
142	ICTAUB0I0	FFFF B11C$_H$	INTTAUB0I0	Interrupt for TAUB0 channel 0	TAUB0	Edge	108E$_H$	√	√	√			+238$_H$
143	ICTAUB0I1	FFFF B11E$_H$	INTTAUB0I1	Interrupt for TAUB0 channel 1	TAUB0	Edge	108F$_H$	√	√	√			+23C$_H$
144	ICTAUB0I2	FFFF B120$_H$	INTTAUB0I2	Interrupt for TAUB0 channel 2	TAUB0	Edge	1090$_H$	√	√	√			+240$_H$
145	ICTAUB0I3	FFFF B122$_H$	INTTAUB0I3*7	Interrupt for TAUB0 channel 3	TAUB0	Edge	1091$_H$	√	√	√			+244$_H$
	ICPWGA16		INTPWGA16*7	PWGA16 interrupt	PWGA16	Edge							
146	ICTAUB0I4	FFFF B124$_H$	INTTAUB0I4	Interrupt for TAUB0 channel 4	TAUB0	Edge	1092$_H$	√	√	√			+248$_H$

Table 7.4 EI Level Maskable Interrupt Sources (5/8)

Interrupt			Interrupt Request				Exception Source Code	144 pins	176 pins	233 pins	Handler Address (Offset)*8		
											Direct Jumping to an Address		
Channel No.*1	Control Register												
	Name	Address	Name	Source	Unit	Detection Type*2					RINT = 0	RINT = 1	Reference to a Table*5
147	ICTAUB0I5	FFFF B126$_H$	INTTAUB0I5*7	Interrupt for TAUB0 channel 5	TAUB0	Edge	1093$_H$	√	√	√	*3	*4	+24C$_H$
	ICPWGA17		INTPWGA17*7	PWGA17 interrupt	PWGA17	Edge							
148	ICTAUB0I6	FFFF B128$_H$	INTTAUB0I6	Interrupt for TAUB0 channel 6	TAUB0	Edge	1094$_H$	√	√	√			+250$_H$
149	ICTAUB0I7	FFFF B12A$_H$	INTTAUB0I7*7	Interrupt for TAUB0 channel 7	TAUB0	Edge	1095$_H$	√	√	√			+254$_H$
	ICPWGA18		INTPWGA18*7	PWGA18 interrupt	PWGA18	Edge							
150	ICTAUB0I8	FFFF B12C$_H$	INTTAUB0I8	Interrupt for TAUB0 channel 8	TAUB0	Edge	1096$_H$	√	√	√			+258$_H$
151	ICTAUB0I9	FFFF B12E$_H$	INTTAUB0I9*7	Interrupt for TAUB0 channel 9	TAUB0	Edge	1097$_H$	√	√	√			+25C$_H$
	ICPWGA19		INTPWGA19*7	PWGA19 interrupt	PWGA19	Edge							
152	ICTAUB0I10	FFFF B130$_H$	INTTAUB0I10	Interrupt for TAUB0 channel 10	TAUB0	Edge	1098$_H$	√	√	√			+260$_H$
153	ICTAUB0I11	FFFF B132$_H$	INTTAUB0I11*7	Interrupt for TAUB0 channel 11	TAUB0	Edge	1099$_H$	√	√	√			+264$_H$
	ICPWGA26		INTPWGA26*7	PWGA26 interrupt	PWGA26	Edge							
154	ICTAUB0I12	FFFF B134$_H$	INTTAUB0I12	Interrupt for TAUB0 channel 12	TAUB0	Edge	109A$_H$	√	√	√			+268$_H$
155	ICTAUB0I13	FFFF B136$_H$	INTTAUB0I13*7	Interrupt for TAUB0 channel 13	TAUB0	Edge	109B$_H$	√	√	√			+26C$_H$
	ICPWGA30		INTPWGA30*7	PWGA30 interrupt	PWGA30	Edge							
156	ICTAUB0I14	FFFF B138$_H$	INTTAUB0I14	Interrupt for TAUB0 channel 14	TAUB0	Edge	109C$_H$	√	√	√			+270$_H$
157	ICTAUB0I15	FFFF B13A$_H$	INTTAUB0I15*7	Interrupt for TAUB0 channel 15	TAUB0	Edge	109D$_H$	√	√	√			+274$_H$
	ICPWGA31		INTPWGA31*7	PWGA31 interrupt	PWGA31	Edge							
158	ICCSIH3IC	FFFF B13C$_H$	INTCSIH3IC*6	CSIH3 communication status interrupt	CSIH3	Edge	109E$_H$	√	√	√			+278$_H$
	ICTAUD0I2_2		INTTAUD0I2_2*6	Interrupt for TAUD0 channel 2	TAUD0	Edge							
159	ICCSIH3IR	FFFF B13E$_H$	INTCSIH3IR*6	CSIH3 receive status interrupt	CSIH3	Edge	109F$_H$	√	√	√			+27C$_H$
	ICTAUD0I10_2		INTTAUD0I10_2*6	Interrupt for TAUD0 channel 10	TAUD0	Edge							
160	ICCSIH3IRE	FFFF B140$_H$	INTCSIH3IRE*6	CSIH3 communication error interrupt	CSIH3	Edge	10A0$_H$	√	√	√			+280$_H$
	ICTAUD0I12_2		INTTAUD0I12_2*6	Interrupt for TAUD0 channel 12	TAUD0	Edge							
161	ICCSIH3IJC	FFFF B142$_H$	INTCSIH3IJC*6	CSIH3 job completion interrupt	CSIH3	Edge	10A1$_H$	√	√	√			+284$_H$
	ICTAUD0I14_2		INTTAUD0I14_2*6	Interrupt for TAUD0 channel 14	TAUD0	Edge							
162	ICRLIN22	FFFF B144$_H$	INTRLIN22	RLIN22 interrupt	RLIN240	Edge	10A2$_H$	√	√	√			+288$_H$
163	ICRLIN23	FFFF B146$_H$	INTRLIN23	RLIN23 interrupt	RLIN240	Edge	10A3$_H$	√	√	√			+28C$_H$
164	ICRLIN32	FFFF B148$_H$	INTRLIN32	RLIN32 interrupt	RLIN32	Edge	10A4$_H$	√	√	√			+290$_H$
165	ICRLIN32UR0	FFFF B14A$_H$	INTRLIN32UR0	RLIN32 transmit interrupt	RLIN32	Edge	10A5$_H$	√	√	√			+294$_H$
166	ICRLIN32UR1	FFFF B14C$_H$	INTRLIN32UR1	RLIN32 receive complete interrupt	RLIN32	Edge	10A6$_H$	√	√	√			+298$_H$
167	ICRLIN32UR2	FFFF B14E$_H$	INTRLIN32UR2	RLIN32 status interrupt	RLIN32	Edge	10A7$_H$	√	√	√			+29C$_H$
168	ICTAUJ1I0	FFFF B150$_H$	INTTAUJ1I0	Interrupt for TAUJ1 channel 0	TAUJ1	Edge	10A8$_H$	√	√	√			+2A0$_H$
169	ICTAUJ1I1	FFFF B152$_H$	INTTAUJ1I1	Interrupt for TAUJ1 channel 1	TAUJ1	Edge	10A9$_H$	√	√	√			+2A4$_H$
170	ICTAUJ1I2	FFFF B154$_H$	INTTAUJ1I2	Interrupt for TAUJ1 channel 2	TAUJ1	Edge	10AA$_H$	√	√	√			+2A8$_H$
171	ICTAUJ1I3	FFFF B156$_H$	INTTAUJ1I3	Interrupt for TAUJ1 channel 3	TAUJ1	Edge	10AB$_H$	√	√	√			+2AC$_H$
172	Reserved	FFFF B158$_H$					10AC$_H$	—	—	—			+2B0$_H$
173	ICFLXA0FDA	FFFF B15A$_H$	INTFLXA0FDA	FIFO transfer interrupt	FLXA0	Level	10AD$_H$	√	√	√			+2B4$_H$
174	ICFLXA0FW	FFFF B15C$_H$	INTFLXA0FW	FIFO transfer warning interrupt	FLXA0	Level	10AE$_H$	√	√	√			+2B8$_H$
175	ICFLXA0IQE	FFFF B15E$_H$	INTFLXA0IQE	Input queue empty interrupt	FLXA0	Level	10AF$_H$	√	√	√			+2BC$_H$
176	ICFLXA0IQF	FFFF B160$_H$	INTFLXA0IQF	Input queue full interrupt	FLXA0	Level	10B0$_H$	√	√	√			+2C0$_H$
177	ICFLXA0OT	FFFF B162$_H$	INTFLXA0OT	Output transfer end interrupt	FLXA0	Level	10B1$_H$	√	√	√			+2C4$_H$
178	ICFLXA0OW	FFFF B164$_H$	INTFLXA0OW	Output transfer warning interrupt	FLXA0	Level	10B2$_H$	√	√	√			+2C8$_H$
179	ICFLXA0LINE0	FFFF B166$_H$	INTFLXA0LINE0	FlexRay0 interrupt	FLXA0	Level	10B3$_H$	√	√	√			+2CC$_H$
180	ICFLXA0LINE1	FFFF B168$_H$	INTFLXA0LINE1	FlexRay1 interrupt	FLXA0	Level	10B4$_H$	√	√	√			+2D0$_H$
181	ICFLXA0TIM0	FFFF B16A$_H$	INTFLXA0TIM0	Timer 0 interrupt	FLXA0	Level	10B5$_H$	√	√	√			+2D4$_H$
182	ICFLXA0TIM1	FFFF B16C$_H$	INTFLXA0TIM1	Timer 1 interrupt	FLXA0	Level	10B6$_H$	√	√	√			+2D8$_H$
183	ICFLXA0TIM2	FFFF B16E$_H$	INTFLXA0TIM2	Timer 2 interrupt	FLXA0	Level	10B7$_H$	√	√	√			+2DC$_H$

Table 7.4 EI Level Maskable Interrupt Sources (6/8)

Interrupt			Interrupt Request				Exception Source Code	144 pins	176 pins	233 pins	Handler Address (Offset)*8		
											Direct Jumping to an Address		
Channel No.*1	Control Register												
	Name	Address	Name	Source	Unit	Detection Type*2					RINT = 0	RINT = 1	Reference to a Table*5
184	ICPWGA24	FFFF B170$_H$	INTPWGA24	PWGA24 interrupt	PWGA24	Edge	10B8$_H$	√	√	√	*3	*4	+2E0$_H$
185	ICPWGA25	FFFF B172$_H$	INTPWGA25	PWGA25 interrupt	PWGA25	Edge	10B9$_H$	√	√	√			+2E4$_H$
186	ICPWGA27	FFFF B174$_H$	INTPWGA27	PWGA27 interrupt	PWGA27	Edge	10BA$_H$	√	√	√			+2E8$_H$
187	ICPWGA28	FFFF B176$_H$	INTPWGA28	PWGA28 interrupt	PWGA28	Edge	10BB$_H$	√	√	√			+2EC$_H$
188	ICPWGA29	FFFF B178$_H$	INTPWGA29	PWGA29 interrupt	PWGA29	Edge	10BC$_H$	√	√	√			+2F0$_H$
189	ICPWGA32	FFFF B17A$_H$	INTPWGA32	PWGA32 interrupt	PWGA32	Edge	10BD$_H$	√	√	√			+2F4$_H$
190	ICPWGA33	FFFF B17C$_H$	INTPWGA33	PWGA33 interrupt	PWGA33	Edge	10BE$_H$	√	√	√			+2F8$_H$
191	ICPWGA34	FFFF B17E$_H$	INTPWGA34	PWGA34 interrupt	PWGA34	Edge	10BF$_H$	√	√	√			+2FC$_H$
192	ICPWGA35	FFFF B180$_H$	INTPWGA35	PWGA35 interrupt	PWGA35	Edge	10C0$_H$	√	√	√			+300$_H$
193	ICPWGA36	FFFF B182$_H$	INTPWGA36	PWGA36 interrupt	PWGA36	Edge	10C1$_H$	√	√	√			+304$_H$
194	ICPWGA37	FFFF B184$_H$	INTPWGA37	PWGA37 interrupt	PWGA37	Edge	10C2$_H$	√	√	√			+308$_H$
195	ICPWGA38	FFFF B186$_H$	INTPWGA38	PWGA38 interrupt	PWGA38	Edge	10C3$_H$	√	√	√			+30C$_H$
196	ICPWGA39	FFFF B188$_H$	INTPWGA39	PWGA39 interrupt	PWGA39	Edge	10C4$_H$	√	√	√			+310$_H$
197	ICPWGA40	FFFF B18A$_H$	INTPWGA40	PWGA40 interrupt	PWGA40	Edge	10C5$_H$	√	√	√			+314$_H$
198	ICPWGA41	FFFF B18C$_H$	INTPWGA41	PWGA41 interrupt	PWGA41	Edge	10C6$_H$	√	√	√			+318$_H$
199	ICPWGA42	FFFF B18E$_H$	INTPWGA42	PWGA42 interrupt	PWGA42	Edge	10C7$_H$	√	√	√			+31C$_H$
200	ICPWGA43	FFFF B190$_H$	INTPWGA43	PWGA43 interrupt	PWGA43	Edge	10C8$_H$	√	√	√			+320$_H$
201	ICPWGA44	FFFF B192$_H$	INTPWGA44	PWGA44 interrupt	PWGA44	Edge	10C9$_H$	√	√	√			+324$_H$
202	ICPWGA45	FFFF B194$_H$	INTPWGA45	PWGA45 interrupt	PWGA45	Edge	10CA$_H$	√	√	√			+328$_H$
203	ICPWGA46	FFFF B196$_H$	INTPWGA46	PWGA46 interrupt	PWGA46	Edge	10CB$_H$	√	√	√			+32C$_H$
204	ICPWGA47	FFFF B198$_H$	INTPWGA47	PWGA47 interrupt	PWGA47	Edge	10CC$_H$	√	√	√			+330$_H$
205	ICP9	FFFF B19A$_H$	INTP9	External interrupt	Port	Edge	10CD$_H$	√	√	√			+334$_H$
206	ICP13	FFFF B19C$_H$	INTP13	External interrupt	Port	Edge	10CE$_H$	√	√	√			+338$_H$
207	ICP14	FFFF B19E$_H$	INTP14	External interrupt	Port	Edge	10CF$_H$	√	√	√			+33C$_H$
208	ICP15	FFFF B1A0$_H$	INTP15	External interrupt	Port	Edge	10D0$_H$	√	√	√			+340$_H$
209	ICRTCA01S	FFFF B1A2$_H$	INTRTCA01S	RTCA0 1 second interval interrupt	RTCA	Edge	10D1$_H$	√	√	√			+344$_H$
210	ICRTCA0AL	FFFF B1A4$_H$	INTRTCA0AL	RTCA0 alarm interrupt	RTCA	Edge	10D2$_H$	√	√	√			+348$_H$
211	ICRTCA0R	FFFF B1A6$_H$	INTRTCA0R	RTCA0 periodic interrupt	RTCA	Edge	10D3$_H$	√	√	√			+34C$_H$
212	ICADCA1ERR	FFFF B1A8$_H$	INTADCA1ERR	ADCA1 error interrupt	ADCA1	Edge	10D4$_H$	√	√	√			+350$_H$
213	ICADCA1I0	FFFF B1AA$_H$	INTADCA1I0	ADCA1 scan group 1 (SG1) end interrupt	ADCA1	Edge	10D5$_H$	√	√	√			+354$_H$
214	ICADCA1I1	FFFF B1AC$_H$	INTADCA1I1	ADCA1 scan group 2 (SG2) end interrupt	ADCA1	Edge	10D6$_H$	√	√	√			+358$_H$
215	ICADCA1I2	FFFF B1AE$_H$	INTADCA1I2	ADCA1 scan group 3 (SG3) end interrupt	ADCA1	Edge	10D7$_H$	√	√	√			+35C$_H$
216	Reserved	FFFF B1B0$_H$					10D8$_H$	—	—	—			+360$_H$
217	ICRCAN2ERR	FFFF B1B2$_H$	INTRCAN2ERR	CAN2 error interrupt	RS-CAN0	Level	10D9$_H$	√	√	√			+364$_H$
218	ICRCAN2REC	FFFF B1B4$_H$	INTRCAN2REC	CAN2 transmit/receive FIFO receive complete interrupt	RS-CAN0	Level	10DA$_H$	√	√	√			+368$_H$
219	ICRCAN2TRX	FFFF B1B6$_H$	INTRCAN2TRX	CAN2 transmit interrupt	RS-CAN0	Level	10DB$_H$	√	√	√			+36C$_H$
220	ICRCAN3ERR	FFFF B1B8$_H$	INTRCAN3ERR	CAN3 error interrupt	RS-CAN0	Level	10DC$_H$	√	√	√			+370$_H$
221	ICRCAN3REC	FFFF B1BA$_H$	INTRCAN3REC	CAN3 transmit/receive FIFO receive complete interrupt	RS-CAN0	Level	10DD$_H$	√	√	√			+374$_H$
222	ICRCAN3TRX	FFFF B1BC$_H$	INTRCAN3TRX	CAN3 transmit interrupt	RS-CAN0	Level	10DE$_H$	√	√	√			+378$_H$
223	ICCSIG1IC	FFFF B1BE$_H$	INTCSIG1IC	CSIG1 communication status interrupt	CSIG1	Edge	10DF$_H$	√	√	√			+37C$_H$
224	ICCSIG1IR	FFFF B1C0$_H$	INTCSIG1IR	CSIG1 receive status interrupt	CSIG1	Edge	10E0$_H$	√	√	√			+380$_H$
225	ICCSIG1IRE	FFFF B1C2$_H$	INTCSIG1IRE	CSIG1 communication error interrupt	CSIG1	Edge	10E1$_H$	√	√	√			+384$_H$

Table 7.4 EI Level Maskable Interrupt Sources (7/8)

Interrupt			Interrupt Request				Exception Source Code	144 pins	176 pins	233 pins	Handler Address (Offset)*8		
											Direct Jumping to an Address		
Channel No.*1	Control Register												
	Name	Address	Name	Source	Unit	Detection Type*2					RINT = 0	RINT = 1	Reference to a Table*5
226	ICRLIN24	FFFF B1C4$_H$	INTRLIN24	RLIN24 interrupt	RLIN241	Edge	10E2$_H$	√	√	√	*3	*4	+388$_H$
227	ICRLIN25	FFFF B1C6$_H$	INTRLIN25	RLIN25 interrupt	RLIN241	Edge	10E3$_H$	√	√	√			+38C$_H$
228	ICRLIN33	FFFF B1C8$_H$	INTRLIN33	RLIN33 interrupt	RLIN33	Edge	10E4$_H$	√	√	√			+390$_H$
229	ICRLIN33UR0	FFFF B1CA$_H$	INTRLIN33UR0	RLIN33 transmit interrupt	RLIN33	Edge	10E5$_H$	√	√	√			+394$_H$
230	ICRLIN33UR1	FFFF B1CC$_H$	INTRLIN33UR1	RLIN33 receive complete interrupt	RLIN33	Edge	10E6$_H$	√	√	√			+398$_H$
231	ICRLIN33UR2	FFFF B1CE$_H$	INTRLIN33UR2	RLIN33 status interrupt	RLIN33	Edge	10E7$_H$	√	√	√			+39C$_H$
232	ICRLIN34	FFFF B1D0$_H$	INTRLIN34	RLIN34 interrupt	RLIN34	Edge	10E8$_H$	√	√	√			+3A0$_H$
233	ICRLIN34UR0	FFFF B1D2$_H$	INTRLIN34UR0	RLIN34 transmit interrupt	RLIN34	Edge	10E9$_H$	√	√	√			+3A4$_H$
234	ICRLIN34UR1	FFFF B1D4$_H$	INTRLIN34UR1	RLIN34 receive complete interrupt	RLIN34	Edge	10EA$_H$	√	√	√			+3A8$_H$
235	ICRLIN34UR2	FFFF B1D6$_H$	INTRLIN34UR2	RLIN34 status interrupt	RLIN34	Edge	10EB$_H$	√	√	√			+3AC$_H$
236	ICRLIN35	FFFF B1D8$_H$	INTRLIN35	RLIN35 interrupt	RLIN35	Edge	10EC$_H$	√	√	√			+3B0$_H$
237	ICRLIN35UR0	FFFF B1DA$_H$	INTRLIN35UR0	RLIN35 transmit interrupt	RLIN35	Edge	10ED$_H$	√	√	√			+3B4$_H$
238	ICRLIN35UR1	FFFF B1DC$_H$	INTRLIN35UR1	RLIN35 receive complete interrupt	RLIN35	Edge	10EE$_H$	√	√	√			+3B8$_H$
239	ICRLIN35UR2	FFFF B1DE$_H$	INTRLIN35UR2	RLIN35 status interrupt	RLIN35	Edge	10EF$_H$	√	√	√			+3BC$_H$
240	ICPWGA48	FFFF B1E0$_H$	INTPWGA48	PWGA48 interrupt	PWGA48	Edge	10F0$_H$	√	√	√			+3C0$_H$
241	ICPWGA49	FFFF B1E2$_H$	INTPWGA49	PWGA49 interrupt	PWGA49	Edge	10F1$_H$	√	√	√			+3C4$_H$
242	ICPWGA50	FFFF B1E4$_H$	INTPWGA50	PWGA50 interrupt	PWGA50	Edge	10F2$_H$	√	√	√			+3C8$_H$
243	ICPWGA51	FFFF B1E6$_H$	INTPWGA51	PWGA51 interrupt	PWGA51	Edge	10F3$_H$	√	√	√			+3CC$_H$
244	ICPWGA52	FFFF B1E8$_H$	INTPWGA52	PWGA52 interrupt	PWGA52	Edge	10F4$_H$	√	√	√			+3D0$_H$
245	ICPWGA53	FFFF B1EA$_H$	INTPWGA53	PWGA53 interrupt	PWGA53	Edge	10F5$_H$	√	√	√			+3D4$_H$
246	ICPWGA54	FFFF B1EC$_H$	INTPWGA54	PWGA54 interrupt	PWGA54	Edge	10F6$_H$	√	√	√			+3D8$_H$
247	ICPWGA55	FFFF B1EE$_H$	INTPWGA55	PWGA55 interrupt	PWGA55	Edge	10F7$_H$	√	√	√			+3DC$_H$
248	ICPWGA56	FFFF B1F0$_H$	INTPWGA56	PWGA56 interrupt	PWGA56	Edge	10F8$_H$	√	√	√			+3E0$_H$
249	ICPWGA57	FFFF B1F2$_H$	INTPWGA57	PWGA57 interrupt	PWGA57	Edge	10F9$_H$	√	√	√			+3E4$_H$
250	ICPWGA58	FFFF B1F4$_H$	INTPWGA58	PWGA58 interrupt	PWGA58	Edge	10FA$_H$	√	√	√			+3E8$_H$
251	ICPWGA59	FFFF B1F6$_H$	INTPWGA59	PWGA59 interrupt	PWGA59	Edge	10FB$_H$	√	√	√			+3EC$_H$
252	ICPWGA60	FFFF B1F8$_H$	INTPWGA60	PWGA60 interrupt	PWGA60	Edge	10FC$_H$	√	√	√			+3F0$_H$
253	ICPWGA61	FFFF B1FA$_H$	INTPWGA61	PWGA61 interrupt	PWGA61	Edge	10FD$_H$	√	√	√			+3F4$_H$
254	ICPWGA62	FFFF B1FC$_H$	INTPWGA62	PWGA62 interrupt	PWGA62	Edge	10FE$_H$	√	√	√			+3F8$_H$
255	ICPWGA63	FFFF B1FE$_H$	INTPWGA63	PWGA63 interrupt	PWGA63	Edge	10FF$_H$	√	√	√			+3FC$_H$
256	ICTAUB1I0	FFFF B200$_H$	INTTAUB1I0	Interrupt for TAUB1 channel 0	TAUB1	Edge	1100$_H$	—	√	√			+400$_H$
257	ICTAUB1I1	FFFF B202$_H$	INTTAUB1I1	Interrupt for TAUB1 channel 1	TAUB1	Edge	1101$_H$	—	√	√			+404$_H$
258	ICTAUB1I2	FFFF B204$_H$	INTTAUB1I2	Interrupt for TAUB1 channel 2	TAUB1	Edge	1102$_H$	—	√	√			+408$_H$
259	ICTAUB1I3	FFFF B206$_H$	INTTAUB1I3	Interrupt for TAUB1 channel 3	TAUB1	Edge	1103$_H$	—	√	√			+40C$_H$
260	ICTAUB1I4	FFFF B208$_H$	INTTAUB1I4	Interrupt for TAUB1 channel 4	TAUB1	Edge	1104$_H$	—	√	√			+410$_H$
261	ICTAUB1I5	FFFF B20A$_H$	INTTAUB1I5	Interrupt for TAUB1 channel 5	TAUB1	Edge	1105$_H$	—	√	√			+414$_H$
262	ICTAUB1I6	FFFF B20C$_H$	INTTAUB1I6	Interrupt for TAUB1 channel 6	TAUB1	Edge	1106$_H$	—	√	√			+418$_H$
263	ICTAUB1I7	FFFF B20E$_H$	INTTAUB1I7	Interrupt for TAUB1 channel 7	TAUB1	Edge	1107$_H$	—	√	√			+41C$_H$
264	ICTAUB1I8	FFFF B210$_H$	INTTAUB1I8	Interrupt for TAUB1 channel 8	TAUB1	Edge	1108$_H$	—	√	√			+420$_H$
265	ICTAUB1I9	FFFF B212$_H$	INTTAUB1I9	Interrupt for TAUB1 channel 9	TAUB1	Edge	1109$_H$	—	√	√			+424$_H$
266	ICTAUB1I10	FFFF B214$_H$	INTTAUB1I10	Interrupt for TAUB1 channel 10	TAUB1	Edge	110A$_H$	—	√	√			+428$_H$
267	ICTAUB1I11	FFFF B216$_H$	INTTAUB1I11	Interrupt for TAUB1 channel 11	TAUB1	Edge	110B$_H$	—	√	√			+42C$_H$
268	ICTAUB1I12	FFFF B218$_H$	INTTAUB1I12	Interrupt for TAUB1 channel 12	TAUB1	Edge	110C$_H$	—	√	√			+430$_H$
269	ICTAUB1I13	FFFF B21A$_H$	INTTAUB1I13	Interrupt for TAUB1 channel 13	TAUB1	Edge	110D$_H$	—	√	√			+434$_H$
270	ICTAUB1I14	FFFF B21C$_H$	INTTAUB1I14	Interrupt for TAUB1 channel 14	TAUB1	Edge	110E$_H$	—	√	√			+438$_H$
271	ICTAUB1I15	FFFF B21E$_H$	INTTAUB1I15	Interrupt for TAUB1 channel 15	TAUB1	Edge	110F$_H$	—	√	√			+43C$_H$
272	ICRCAN4ERR	FFFF B220$_H$	INTRCAN4ERR	CAN4 error interrupt	RS-CAN0	Level	1110$_H$	√	√	√			+440$_H$

Table 7.4 EI Level Maskable Interrupt Sources (8/8)

Interrupt			Interrupt Request				Exception Source Code	144 pins	176 pins	233 pins	Handler Address (Offset)*8		
											Direct Jumping to an Address		
Channel No.*1	Control Register										RINT = 0	RINT = 1	Reference to a Table*5
	Name	Address	Name	Source	Unit	Detection Type*2							
273	ICRCAN4REC	FFFF B222$_H$	INTRCAN4REC	CAN4 transmit/receive FIFO receive complete interrupt	RS-CAN0	Level	1111$_H$	√	√	√	*3	*4	+444$_H$
274	ICRCAN4TRX	FFFF B224$_H$	INTRCAN4TRX	CAN4 transmit interrupt	RS-CAN0	Level	1112$_H$	√	√	√			+448$_H$
275	ICRLIN26	FFFF B226$_H$	INTRLIN26	RLIN26 interrupt	RLIN241	Edge	1113$_H$	—	√	√			+44C$_H$
276	ICRLIN27	FFFF B228$_H$	INTRLIN27	RLIN27 interrupt	RLIN241	Edge	1114$_H$	—	√	√			+450$_H$
277	ICPWGA64	FFFF B22A$_H$	INTPWGA64	PWGA64 interrupt	PWGA64	Edge	1115$_H$	—	√	√			+454$_H$
278	ICPWGA65	FFFF B22C$_H$	INTPWGA65	PWGA65 interrupt	PWGA65	Edge	1116$_H$	—	√	√			+458$_H$
279	ICPWGA66	FFFF B22E$_H$	INTPWGA66	PWGA66 interrupt	PWGA66	Edge	1117$_H$	—	√	√			+45C$_H$
280	ICPWGA67	FFFF B230$_H$	INTPWGA67	PWGA67 interrupt	PWGA67	Edge	1118$_H$	—	√	√			+460$_H$
281	ICPWGA68	FFFF B232$_H$	INTPWGA68	PWGA68 interrupt	PWGA68	Edge	1119$_H$	—	√	√			+464$_H$
282	ICPWGA69	FFFF B234$_H$	INTPWGA69	PWGA69 interrupt	PWGA69	Edge	111A$_H$	—	√	√			+468$_H$
283	ICPWGA70	FFFF B236$_H$	INTPWGA70	PWGA70 interrupt	PWGA70	Edge	111B$_H$	—	√	√			+46C$_H$
284	ICPWGA71	FFFF B238$_H$	INTPWGA71	PWGA71 interrupt	PWGA71	Edge	111C$_H$	—	√	√			+470$_H$
285	ICRLIN28	FFFF B23A$_H$	INTRLIN28	RLIN28 interrupt	RLIN242	Edge	111D$_H$	—	√	√			+474$_H$
286	ICRLIN29	FFFF B23C$_H$	INTRLIN29	RLIN29 interrupt	RLIN242	Edge	111E$_H$	—	√	√			+478$_H$
287	ICRCAN5ERR	FFFF B23E$_H$	INTRCAN5ERR	CAN5 error interrupt	RS-CAN0	Level	111F$_H$	√	√	√			+47C$_H$
288	ICRCAN5REC	FFFF B240$_H$	INTRCAN5REC	CAN5 transmit/receive FIFO receive complete interrupt	RS-CAN0	Level	1120$_H$	√	√	√			+480$_H$
289	ICRCAN5TRX	FFFF B242$_H$	INTRCAN5TRX	CAN5 transmit interrupt	RS-CAN0	Level	1121$_H$	√	√	√			+484$_H$
290	ICPWGA72	FFFF B244$_H$	INTPWGA72	PWGA72 interrupt	PWGA72	Edge	1122$_H$	—	—	√			+488$_H$
291	ICPWGA73	FFFF B246$_H$	INTPWGA73	PWGA73 interrupt	PWGA73	Edge	1123$_H$	—	—	√			+48C$_H$
292	ICPWGA74	FFFF B248$_H$	INTPWGA74	PWGA74 interrupt	PWGA74	Edge	1124$_H$	—	—	√			+490$_H$
293	ICPWGA75	FFFF B24A$_H$	INTPWGA75	PWGA75 interrupt	PWGA75	Edge	1125$_H$	—	—	√			+494$_H$
294	ICPWGA76	FFFF B24C$_H$	INTPWGA76	PWGA76 interrupt	PWGA76	Edge	1126$_H$	—	—	√			+498$_H$
295	ICPWGA77	FFFF B24E$_H$	INTPWGA77	PWGA77 interrupt	PWGA77	Edge	1127$_H$	—	—	√			+49C$_H$
296	ICPWGA78	FFFF B250$_H$	INTPWGA78	PWGA78 interrupt	PWGA78	Edge	1128$_H$	—	—	√			+4A0$_H$
297	ICPWGA79	FFFF B252$_H$	INTPWGA79	PWGA79 interrupt	PWGA79	Edge	1129$_H$	—	—	√			+4A4$_H$

Note 1. Each interrupt is connected to INTC1 channel 0 to 31 and INTC2 channel 32 to 297.

Note 2. This indicates whether an interrupt source is detected at the level or edge. This also affects the value after reset of an EI level interrupt control register. For details, see **Section 7.4.2, ICxxx — EI Level Interrupt Control Registers**. For detection at level, an interrupt source is cleared by accessing to the register that retains an interrupt source. Dummy-reading of the register and execution of the SYNCP instruction are required to reflect the result of the register update to the subsequent instruction. For details, see **Section 3.3.1.1**.

Note 3. Irrespective of interrupt channels, an offset address is determined in the range from +100$_H$ to 1F0$_H$ according to the priority (0 to 15).

Note 4. Irrespective of the priority, offset addresses are uniformly +100$_H$.

Note 5. The table reference method uses a table for reading an exception handler address for each interrupt channel, and it extracts handler address by referencing that table. Table reference position is determined by the following formula.
Exception handler address read position = INTBP register + channel number × 4 bytes

Note 6. The same interrupt source is assigned to different interrupt channels. For details, see **Section 7.5.1, SELB_INTC1 — INTC1 Interrupt Select Register**.

Note 7. Two interrupt sources are assigned to the same interrupt channel. For details, see **Section 7.5.2, SELB_INTC2 — INTC2 Interrupt Select Register**.

Note 8. For details, see **Section 7.10, Exception Handler Address**.

Note 9. INTOSTM0 is assigned to EIINT and FEINT, and using EIINT and FEINT at the same time is not allowed. If OSTM0 is used as an TSU (Timing Supervision Unit), use FEINT. Otherwise, use EIINT.

Note 10. For details on the interrupt source, see the *RH850/F1L, RH850/F1M, RH850/F1H Flash Memory User's Manual: Hardware Interface*.

7.2.2 FE Level Non-Maskable Interrupt Sources

7.2.2.1 WDTNMIF — FENMI Factor Register

This register contains information about which source has generated the FE level non-maskable interrupt (FENMI). This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC0 0000$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTA1 NMIF	WDTA0 NMIF	TNMIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.5 WDTNMIF Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	WDTA1NMIF	WDTA1NMI flag 0: No WDTA1NMI occurred 1: WDTA1NMI has occurred
1	WDTA0NMIF	WDTA0NMI flag 0: No WDTA0NMI occurred 1: WDTA0NMI has occurred
0	TNMIF	Input signal flag from the NMI pin 0: No TNMI occurred 1: TNMI has occurred

7.2.2.2 WDTNMIFC — WDTNMI Factor Clear Register

This register clears the FE level non-maskable interrupt flags of the WDTNMIF register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFC0 0008$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTA1NMIFC	WDTA0NMIFC	TNMIFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W

Table 7.6 WDTNMIFC Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When writing, write the value after reset.
2	WDTA1NMIFC	WDTA1NMIF flag clear 0: — 1: Clear
1	WDTA0NMIFC	WDTA0NMIF flag clear 0: — 1: Clear
0	TNMIFC	TNMIF flag clear 0: — 1: Clear

7.2.3 FE Level Maskable Interrupt Sources

7.2.3.1 FEINTF — FEINT Factor Register

This register contains information about which source has generated the FE level maskable interrupt (FEINT). This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC0 0100$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	OSTM4 FEIF	OSTM3 FEIF	OSTM2 FEIF	OSTM1 FEIF	GUARD FEIF	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVIH FEIF	OSTM0 FEIF	ECCRA MFEIF	ECCSC FLI0 FEIF	—	ECCDC SIH3FE IF	ECCDC SIH2 FEIF	ECCDC SIH1 FEIF	ECCDC SIH0 FEIF	ECCDC NRAM0 FEIF	ECCDF LRAMF EIF	ECCDE EP0 FEIF	—	—	—	LVIL FEIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.7 FEINTF Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21	OSTM4FEIF	INTOSTM4_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
20	OSTM3FEIF	INTOSTM3_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
19	OSTM2FEIF	INTOSTM2_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
18	OSTM1FEIF	INTOSTM1_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
17	GUARDFEIF	INTGUARD interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
16	Reserved	When read, the value after reset is returned.
15	LVIHFEIF	INTLVIH interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
14	OSTM0 FEIF	INTOSTM0_FE interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
13	ECCRAM FEIF	INTECCRAM interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
12	ECCSCFLI0 FEIF	INTECCSCFLI0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
11	Reserved	When read, the value after reset is returned.

Table 7.7 FEINTF Register Contents (2/2)

Bit Position	Bit Name	Function
10	ECCDCSIH3 FEIF	INTECCDCSIH3 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
9	ECCDCSIH2 FEIF	INTECCDCSIH2 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
8	ECCDCSIH1 FEIF	INTECCDCSIH1 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
7	ECCDCSIH0 FEIF	INTECCDCSIH0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
6	ECCDCNRAM0 FEIF	INTECCDCNRAM0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
5	ECCDFLRAM FEIF	INTECCDFLRAM interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
4	ECCDEEP0 FEIF	INTECCDEEP0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
3 to 1	Reserved	When read, the value after reset is returned.
0	LVILFEIF	INTLVIL interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred

7.2.3.2 FEINTFMSK — FEINT Factor Mask Register

This register masks the FE level maskable interrupt (FEINT). This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFC0 0104$_H$

Value after reset: FFFF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	OSTM4FEIFMSK	OSTM3FEIFMSK	OSTM2FEIFMSK	OSTM1FEIFMSK	GUARDFEIFMSK	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVIHFEIFMSK	OSTM0FEIFMSK	ECCRAMFEIFMSK	ECCSCFLI0FEIFMSK	—	ECCDCSIH3FEIFMSK	ECCDCSIH2FEIFMSK	ECCDCSIH1FEIFMSK	ECCDCSIH0FEIFMSK	ECCDCNRAM0FEIFMSK	ECCDFLRAMFEIFMSK	ECCDEEP0FEIFMSK	—	—	—	LVILFEIFMSK
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W

Table 7.8 FEINTFMSK Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	OSTM4 FEIFMSK	INTOSTM4_FE interrupt mask 0: Not masked 1: Masked
20	OSTM3 FEIFMSK	INTOSTM3_FE interrupt mask 0: Not masked 1: Masked
19	OSTM2 FEIFMSK	INTOSTM2_FE interrupt mask 0: Not masked 1: Masked
18	OSTM1 FEIFMSK	INTOSTM1_FE interrupt mask 0: Not masked 1: Masked
17	GUARD FEIFMSK	INTGUARD interrupt mask 0: Not masked 1: Masked
16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	LVIHFEIFMSK	INTLVIH interrupt mask 0: Not masked 1: Masked
14	OSTM0FEI FMSK	INTOSTM0_FE interrupt mask 0: Not masked 1: Masked
13	ECCRAMFEI FMSK	INTECCRAM interrupt mask 0: Not masked 1: Masked
12	ECCSCFLI0 FEIFMSK	INTECCSCFLI0 interrupt mask 0: Not masked 1: Masked
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 7.8 FEINTFMSK Register Contents (2/2)

Bit Position	Bit Name	Function
10	ECCDCSIH3FEIFMSK	INTECCDCSIH3 interrupt mask 0: Not masked 1: Masked
9	ECCDCSIH2FEIFMSK	INTECCDCSIH2 interrupt mask 0: Not masked 1: Masked
8	ECCDCSIH1FEIFMSK	INTECCDCSIH1 interrupt mask 0: Not masked 1: Masked
7	ECCDCSIH0FEIFMSK	INTECCDCSIH0 interrupt mask 0: Not masked 1: Masked
6	ECCDCNRAM0FEIFMSK	INTECCDCNRAM0 interrupt mask 0: Not masked 1: Masked
5	ECCDFLRAMFEIFMSK	INTECCDFLRAM interrupt mask 0: Not masked 1: Masked
4	ECCDEEP0FEIFMSK	INTECCDEEP0 interrupt mask 0: Not masked 1: Masked
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LVILFEIFMSK	INTLVIL interrupt mask 0: Not masked 1: Masked

7.2.3.3 FEINTFC — FEINT Factor Clear Register

This register clears the bits of the FEINT factor register (FEINTF).

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFC0 0108_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	OSTM4 FEIFC	OSTM3 FEIFC	OSTM2 FEIFC	OSTM1 FEIFC	GUARD FEIFC	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LVIH FEIFC	OSTM0 FEIFC	ECC RAM FEIFC	ECCSC FLI0FEI FC	—	ECCDC SIH3 FEIFC	ECCDC SIH2 FEIFC	ECCDC SIH1 FEIFC	ECCDC SIH0 FEIFC	ECCDC NRAM0 FEIFC	ECCDF LRAMF EIFC	ECC DEEP0 FEIFC	—	—	—	LVIL FEIFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	R	W	W	W	W	W	W	W	R	R	R	W

Table 7.9 FEINTFC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 22	Reserved	When writing, write the value after reset.
21	OSTM4FEIFC	OSTM4FEIF flag clear 0: — 1: Clear
20	OSTM3FEIFC	OSTM3FEIF flag clear 0: — 1: Clear
19	OSTM2FEIFC	OSTM2FEIF flag clear 0: — 1: Clear
18	OSTM1FEIFC	OSTM1FEIF flag clear 0: — 1: Clear
17	GUARDFEIFC	GUARDFEIF flag clear 0: — 1: Clear
16	Reserved	When writing, write the value after reset.
15	LVIHFEIFC	LVIHFEIF flag clear 0: — 1: Clear
14	OSTM0 FEIFC	OSTM0FEIF flag clear 0: — 1: Clear
13	ECCRAM FEIFC	ECCRAMFEIF flag clear 0: — 1: Clear
12	ECCSCFLI0 FEIFC	ECCSCFLI0FEIF flag clear 0: — 1: Clear
11	Reserved	When writing, write the value after reset.
10	ECCDCSIH3 FEIFC	ECCDCSIH3FEIF flag clear 0: — 1: Clear

Table 7.9 FEINTFC Register Contents (2/2)

Bit Position	Bit Name	Function
9	ECCDCSIH2 FEIFC	ECCDCSIH2FEIF flag clear 0: — 1: Clear
8	ECCDCSIH1 FEIFC	ECCDCSIH1FEIF flag clear 0: — 1: Clear
7	ECCDCSIH0 FEIFC	ECCDCSIH0FEIF flag clear 0: — 1: Clear
6	ECCDCNRAM0 FEIFC	ECCDCNRAM0FEIF flag clear 0: — 1: Clear
5	ECCDFLRAMF EIFC	ECCDFLRAMFEIF flag clear 0: — 1: Clear
4	ECCDEEP0 FEIFC	ECCDEEP0FEIF flag clear 0: — 1: Clear
3 to 1	Reserved	When writing, write the value after reset.
0	LVILFEIFC	LVILFEIF flag clear 0: — 1: Clear

7.3 Edge/Level Detection

External interrupts (TNMI and INTPm) can be specified to be generated when a rising edge, falling edge, rising or falling edge, low level, or high level is detected at an external interrupt pin.

The following registers are used to specify the edge and level of each interrupt:

Table 7.10 External Interrupt Edge/Level Detection Registers

Interrupt	Register
TNMI	FCLA0CTL0_NMI
INTP0	FCLA0CTL0_INTPL
INTP1	FCLA0CTL1_INTPL
INTP2	FCLA0CTL2_INTPL
INTP3	FCLA0CTL3_INTPL
INTP4	FCLA0CTL4_INTPL
INTP5	FCLA0CTL5_INTPL
INTP6	FCLA0CTL6_INTPL
INTP7	FCLA0CTL7_INTPL
INTP8	FCLA0CTL0_INTPH
INTP9	FCLA0CTL1_INTPH
INTP10	FCLA0CTL2_INTPH
INTP11	FCLA0CTL3_INTPH
INTP12	FCLA0CTL4_INTPH
INTP13	FCLA0CTL5_INTPH
INTP14	FCLA0CTL6_INTPH
INTP15	FCLA0CTL7_INTPH

See **Section 2, Pin Function** for details of these registers.

7.4 Interrupt Controller Control Registers

Writing to the ICxxx, IMRm (m = 0 to 9), FNC, and FIC registers is enabled only in supervisor mode (PSW.UM = 0).

7.4.1 List of Registers

Table 7.11 List of Interrupt Controller Control Registers

Register Name	Symbol	Address
EI level interrupt control register	ICxxx	*1
EI level interrupt mask register	IMRm	IMR0: FFFE EAF0$_H$ IMRm (m = 1 to 9): FFFF B400$_H$ + (04$_H$ × m)
FE level NMI status register	FNC	FFFE EA78$_H$
FE level maskable interrupt status register	FIC	FFFE EA7A$_H$

Note 1. See **Table 7.4, EI Level Maskable Interrupt Sources**.

Among the registers shown in **Table 7.11**, ICxxx and IMR0, which correspond to interrupt channel numbers 0 to 31, are located in INTC1 of the CPU peripheral field included in each CPU.

Writing is only possible in supervisor mode (PSW.UM = 0).

Among the registers shown in **Table 7.11**, ICxxx and IMR1 to IMR9, which correspond to interrupt channel numbers 32 to 297, are located in INTC2 of peripheral group 0. Writing to these registers is only possible in supervisor mode (UM = 0) by CPU1.

When writing to IMR1 to IMR9, only the bits corresponding to the conditions described above are overwritten; other bits are not updated.

Among the registers shown in **Table 7.11**, the values of those listed as reserved for the given channel numbers in **Table 7.4, EI Level Maskable Interrupt Sources**, must retain their values after a reset.

7.4.2 ICxxx — EI Level Interrupt Control Registers

One of these registers is assigned to each EI level maskable interrupt (EIINT) channel and is used to set the conditions for controlling that channel. This register is initialized by any reset. For each source, see **Table 7.4, EI Level Maskable Interrupt Sources**.

CAUTION

If 0 is written to the RFxxx bit immediately after a peripheral module generates the corresponding interrupt request in edge detection mode (before an interrupt is accepted by the CPU), the request may be lost.

Writing 1 to the RFxxx bit immediately after an interrupt is accepted by the CPU may lead to re-issuing of the request.

Writing method to these registers contains the use of bit manipulation instructions (set1, clr1, and not1).

For bit-manipulation instructions, see also Section 3.3.2, Accessing Registers by Using Bit-Manipulation Instructions.

Executing a bit-manipulation instruction to the lower bytes including the MKxxx bit has no effect on the RFxxx bit.

Access: ICxxx can be read or written in 16-bit units.
ICxxxH and ICxxxL can be read or written in 8- or 1-bit units.
Access to bits 14, 13, 11 to 8, 5, and 4 by using a SET1, CLR1, or NOT1 instruction is prohibited.

Address: See **Table 7.4, EI Level Maskable Interrupt Sources**.

Value after reset: $008F_H$ (edge detection), $808F_H$ (level detection)*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTxxx	—	—	RFxxx	—	—	—	—	MKxxx	TBxxx	—	—	P3xxx	P2xxx	P1xxx	P0xxx
Value after reset	0/1*1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Note 1. The value after reset differs depending on the detection type of a given interrupt (edge detection: 0, level detection: 1). For details, see **Table 7.4, EI Level Maskable Interrupt Sources**.

Table 7.12 ICxxx Register Contents (1/2)

Bit Position	Bit Name	Function
15	CTxxx	This bit indicates the type of interrupt detection. This bit is read only. 0: Edge detection 1: Level detection When writing in 8-bit or 16-bit units, write the value after reset.
14, 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 7.12 ICxxx Register Contents (2/2)

Bit Position	Bit Name	Function
12	RFxxx	This is an interrupt request flag. The RFxxx bit can be written from a program. Setting the RFxxx bit to 1 generates an EI level maskable interrupt n (EIINTn), just as when an interrupt request is acknowledged. 0: No interrupt request is made. 1: Interrupt request is made. **Input Interface** / **Operation** Edge detection (CTxxx = 0): This bit is automatically cleared when an interrupt request is acknowledged by the CPU core. It can be set and cleared by software. When the RFxxx bit is set to 1, an EI level maskable interrupt n (EIINTn) is generated in the same way as acceptance of an interrupt request. Level detection (CTxxx = 1): This bit cannot be set or cleared by software. It can only be read. It is not cleared even if an interrupt request is acknowledged by the CPU core.
11 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	MKxxx	This is the interrupt request mask bit. When the MKxxx bit is set, interrupt requests from the channel are masked and are not issued to the CPU core. The interrupt pending status is not reflected in the ICSR.PMEI bit for any channels that are masked. When the interrupt request from the channel is masked with MKxxx = 1, the RFxxx still reflects the interrupt request for the channel and can be polled in software. When the MKxxx bit is cleared, interrupt requests from the channel are issued to the CPU core for subsequent processing. The state of the MKxxx bit is also reflected in the corresponding IMRm register. 0: Enables interrupt processing 1: Disables interrupt processing
6	TBxxx	This bit is used to select the way to determine the interrupt vector. 0: Direct jumping to an address determined from the level of priority 1: Table reference For details on the way to determine the interrupt vector, see the *RH850G3M User's Manual: Software*.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	P3xxx to P0xxx	These bits specify the interrupt priority as one of 16 levels, with 0 as the highest and 15 as the lowest. When multiple EI level-interrupt requests are made simultaneously, the interrupt from the source with the highest priority setting in these bits is selected and conveyed to the CPU core for servicing first. When P3xxx to P0xxx bits specify the same priority level for simultaneously occurring interrupt requests, the source with the lower channel number takes priority.

CAUTION

Do not access ICxxx registers of interrupt channels listed as "Reserved" in **Table 7.4, EI Level Maskable Interrupt Sources** and of the channels which are not incorporated in the product.

7.4.3 IMRm — EI Level Interrupt Mask Registers (m = 0 to 9)

These registers are a collection of the MKxxx bits of the ICxxx registers. Each bit of IMRm reflects the setting of the corresponding MKxxx bit. The setting of IMRm is also reflected in the corresponding MKxxx bit. This register is initialized by any reset.

Access: IMRm can be read or written in 32-bit units.
IMRmH and IMRmL can be read or written in 16-bit units.
IMRmHH, IMRmHL, IMRmLH, and IMRmLL can be read or written in 8- or 1-bit units.

Address: IMR0: FFFE EAF0$_H$
IMRm (m = 1 to 9): FFFF B400$_H$ + (04$_H$ × m)

Value after reset: FFFF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IMRmEIMK (m × 32 + 31)	IMRmEIMK (m × 32 + 30)	IMRmEIMK (m × 32 + 29)	IMRmEIMK (m × 32 + 28)	IMRmEIMK (m × 32 + 27)	IMRmEIMK (m × 32 + 26)	IMRmEIMK (m × 32 + 25)	IMRmEIMK (m × 32 + 24)	IMRmEIMK (m × 32 + 23)	IMRmEIMK (m × 32 + 22)	IMRmEIMK (m × 32 + 21)	IMRmEIMK (m × 32 + 20)	IMRmEIMK (m × 32 + 19)	IMRmEIMK (m × 32 + 18)	IMRmEIMK (m × 32 + 17)	IMRmEIMK (m × 32 + 16)
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMRmEIMK (m × 32 + 15)	IMRmEIMK (m × 32 + 14)	IMRmEIMK (m × 32 + 13)	IMRmEIMK (m × 32 + 12)	IMRmEIMK (m × 32 + 11)	IMRmEIMK (m × 32 + 10)	IMRmEIMK (m × 32 + 9)	IMRmEIMK (m × 32 + 8)	IMRmEIMK (m × 32 + 7)	IMRmEIMK (m × 32 + 6)	IMRmEIMK (m × 32 + 5)	IMRmEIMK (m × 32 + 4)	IMRmEIMK (m × 32 + 3)	IMRmEIMK (m × 32 + 2)	IMRmEIMK (m × 32 + 1)	IMRmEIMK (m × 32 + 0)
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.13 IMRm Register Contents

Bit Position	Bit Name	Function
31 to 0	IMRmEIMK (m × 32 + 31) to IMRmEIMK (m × 32 + 0)	These are interrupt mask bits for EI level maskable interrupt (EIINT) channels 0 to 297. 0: Enables interrupt servicing 1: Disables interrupt servicing

CAUTION

MKxxx bits which correspond to channels listed as “Reserved” in **Table 7.4, EI Level Maskable Interrupt Sources**, and to channels which are not incorporated in the product must be set to “1”.

7.4.4 FNC — FE Level NMI Status Register

This register indicates the status of an FE level non-maskable interrupt (FENMI).

Access: FNC is a read-only register that can be read in 16-bit units.
FNCH is a read-only register that can be read in 8- or 1-bit units.

Address: FNC: FFFE EA78$_H$
FNCH: FFFE EA79$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FNRF	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.14 FNC Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12	FNRF	Interrupt request flag 0: No interrupt request 1: Interrupt request occurred This bit is automatically cleared when an FE level NMI interrupt request is acknowledged by the CPU core.
11 to 0	Reserved	When read, the value after reset is returned.

7.4.5 FIC — FE Level Maskable Interrupt Status Register

This register indicates the status of an FE level maskable interrupt (FEINT).

Access: FIC is a read-only register that can be read in 16-bit units.
FICH is a read-only register that can be read in 8- or 1-bit units.

Address: FIC: FFFE EA7A$_H$
FICH: FFFE EA7B$_H$

Value after reset: 8000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FIRF	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.15 FIC Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12	FIRF	Interrupt request flag 0: No interrupt request 1: Interrupt request occurred This bit cannot be set or cleared by software. It can only be read.
11 to 0	Reserved	When read, the value after reset is returned.

7.5 EI Level Maskable Interrupt Select Registers

The following registers are used to select an EI level maskable interrupt.

7.5.1 SELB_INTC1 — INTC1 Interrupt Select Register

When two interrupt sources are assigned to one interrupt channel, this register selects which interrupt sources is enabled.

NOTE

The channel described in each bit setting indicates the channel of an interrupt and the priority. For details on channels, see **Table 7.4, EI Level Maskable Interrupt Sources**.

Access: This register can be read or written in 16-bit units.

Address: FFC0 1000$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SELB_INTC1_12	SELB_INTC1_11	SELB_INTC1_10	SELB_INTC1_9	SELB_INTC1_8	SELB_INTC1_7	SELB_INTC1_6	SELB_INTC1_5	SELB_INTC1_4	SELB_INTC1_3	SELB_INTC1_2	SELB_INTC1_1	SELB_INTC1_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.16 SELB_INTC1 Register Contents (1/2)

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	SELB_INTC1_12	Interrupt channel selection 0: INTADCA0I2 (channel 20) INTCSIH0IJC (channel 32) 1: INTCSIH0IJC_1 (channel 20) INTADCA0I2_2 (channel 32)
11	SELB_INTC1_11	Interrupt channel selection 0: INTTAUD0I14 (channel 15) INTCSIH3IJC (channel 161) 1: INTCSIH3IJC_1 (channel 15) INTTAUD0I14_2 (channel 161)
10	SELB_INTC1_10	Interrupt channel selection 0: INTTAUD0I12 (channel 14) INTCSIH3IRE (channel 160) 1: INTCSIH3IRE_1 (channel 14) INTTAUD0I12_2 (channel 160)
9	SELB_INTC1_9	Interrupt channel selection 0: INTTAUD0I10 (channel 13) INTCSIH3IR (channel 159) 1: INTCSIH3IR_1 (channel 13) INTTAUD0I10_2 (channel 159)
8	SELB_INTC1_8	Interrupt channel selection 0: INTTAUD0I2 (channel 9) INTCSIH3IC (channel 158) 1: INTCSIH3IC_1 (channel 9) INTTAUD0I2_2 (channel 158)
7	SELB_INTC1_7	Interrupt channel selection 0: INTTAUD0I8 (channel 12) INTCSIH2IJC (channel 135) 1: INTCSIH2IJC_1 (channel 12) INTTAUD0I8_2 (channel 135)

Table 7.16 SELB_INTC1 Register Contents (2/2)

Bit Position	Bit Name	Function
6	SELB_INTC1_6	Interrupt channel selection 0: INTTAUD0I6 (channel 11) INTCSIH2IRE (channel 134) 1: INTCSIH2IRE_1 (channel 11) INTTAUD0I6_2 (channel 134)
5	SELB_INTC1_5	Interrupt channel selection 0: INTTAUD0I4 (channel 10) INTCSIH2IR (channel 133) 1: INTCSIH2IR_1 (channel 10) INTTAUD0I4_2 (channel 133)
4	SELB_INTC1_4	Interrupt channel selection 0: INTTAUD0I0 (channel 8) INTCSIH2IC (channel 132) 1: INTCSIH2IC_1 (channel 8) INTTAUD0I0_2 (channel 132)
3	SELB_INTC1_3	Interrupt channel selection 0: INTCSIG0IR (channel 28) INTCSIH1IJC (channel 119) 1: INTCSIH1IJC_1 (channel 28) INTCSIG0IR_2 (channel 119)
2	SELB_INTC1_2	Interrupt channel selection 0: INTCSIG0IC (channel 27) INTCSIH1IRE (channel 118) 1: INTCSIH1IRE_1 (channel 27) INTCSIG0IC_2 (channel 118)
1	SELB_INTC1_1	Interrupt channel selection 0: INTTAPA0IVLY0 (channel 17) INTCSIH1IR (channel 117) 1: INTCSIH1IR_1 (channel 17) INTTAPA0IVLY0_2 (channel 117)
0	SELB_INTC1_0	Interrupt channel selection 0: INTTAPA0IPEK0 (channel 16) INTCSIH1IC (channel 116) 1: INTCSIH1IC_1 (channel 16) INTTAPA0IPEK0_2 (channel 116)

CAUTION

The operation of peripheral functions should be enabled after setting the corresponding interrupt source by SELB_INTC1.

7.5.2 SELB_INTC2 — INTC2 Interrupt Select Register

When two interrupt sources are assigned to one interrupt channel, this register selects which interrupt sources is enabled.

Access: This register can be read or written in 16-bit units.

Address: FFC0 1004_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SELB_INTC2_10	SELB_INTC2_9	SELB_INTC2_8	SELB_INTC2_7	SELB_INTC2_6	SELB_INTC2_5	SELB_INTC2_4	SELB_INTC2_3	SELB_INTC2_2	SELB_INTC2_1	SELB_INTC2_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.17 SELB_INTC2 Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When writing, write the value after reset.
10	SELB_INTC2_10	Interrupt channel 157 selection 0: INTTAUB0I15 1: INTPWGA31
9	SELB_INTC2_9	Interrupt channel 155 selection 0: INTTAUB0I13 1: INTPWGA30
8	SELB_INTC2_8	Interrupt channel 153 selection 0: INTTAUB0I11 1: INTPWGA26
7	SELB_INTC2_7	Interrupt channel 151 selection 0: INTTAUB0I9 1: INTPWGA19
6	SELB_INTC2_6	Interrupt channel 149 selection 0: INTTAUB0I7 1: INTPWGA18
5	SELB_INTC2_5	Interrupt channel 147 selection 0: INTTAUB0I5 1: INTPWGA17
4	SELB_INTC2_4	Interrupt channel 145 selection 0: INTTAUB0I3 1: INTPWGA16
3	SELB_INTC2_3	Interrupt channel 88 selection 0: INTENCA0I1 1: INTPWGA7
2	SELB_INTC2_2	Interrupt channel 87 selection 0: INTENCA0I0 1: INTPWGA6
1	SELB_INTC2_1	Interrupt channel 86 selection 0: INTENCA0IUD 1: INTPWGA5
0	SELB_INTC2_0	Interrupt channel 85 selection 0: INTENCA0IOV 1: INTPWGA4

Note 1. For the supported products, see **Table 7.4, EI Level Maskable Interrupt Sources**.

CAUTION

The operation of peripheral functions should be enabled after setting the corresponding interrupt source by SELB_INTC2.

7.6 Interrupt Function System Registers

See **Table 3.31, Interrupt Function System Registers**.

7.6.1 FPIPR — FPI Exception Interrupt Priority

See **Table 3.32, FPIPR Register Contents**.

7.6.2 ISPR — Priority of Interrupt being Serviced

See **Table 3.33, ISPR Register Contents**.

7.6.3 PMR — Interrupt Priority Masking

See **Table 3.34, PMR Register Contents**.

7.6.4 ICSR — Interrupt Control Status

See **Table 3.35, ICSR Register Contents**.

7.6.5 INTCFG — Interrupt Function Setting

See **Table 3.36, INTCFG Register Contents**.

7.7 Operation when Acknowledging an Interrupt

Check whether each interrupt that is reported during instruction execution is acknowledged according to the priority. The procedure for acknowledging each interrupt is shown below.

(1) Check whether the acknowledgment conditions are satisfied and whether interrupts are acknowledged according to their priority.

(2) Calculate the exception handler address according to the current PSW value.*1

(3) For FE-level non-maskable/maskable interrupts, the following processing is performed:

- Save the PC to the FEPC.
- Save the PSW to the FEPSW.
- Store the exception source code in the FEIC.
- Update the PSW and MCTL.*2
- Store the exception handler address calculated in (2) in the PC, and then pass its control to the exception handler.

(4) For EI level exceptions, the following processing is performed:

- Save the PC to the EIPC.
- Save the PSW to the EIPSW.
- Store the exception source code in the EIIC.
- Update the PSW and MCTL.*2
- Store the exception handler address calculated in (2) in the PC, and then pass its control to the exception handler.

Note 1. For details, see **Section 7.10, Exception Handler Address**.

Note 2. For the values to be updated, see *Table 4.1 List of Exception Sources* in the *RH850G3M User's Manual: Software*.

The following figure shows steps (1) to (4).

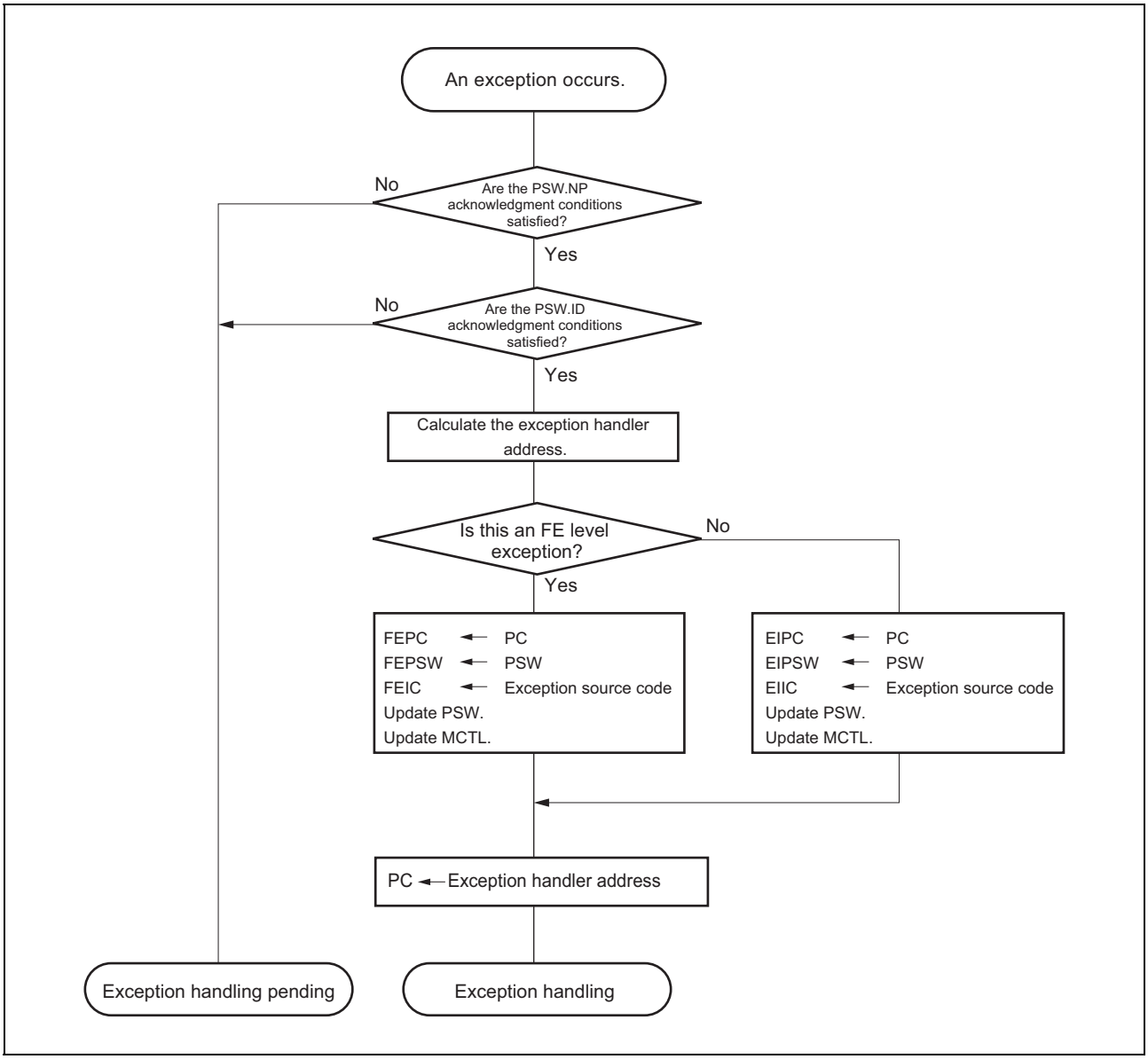


Figure 7.2 Operation when Acknowledging an Interrupt

7.8 Return from Interrupts

To return from interrupt handling, execute the return instruction (EIRET or FERET) corresponding to each relevant interrupt level.

When a context has been saved in a stack and the like, the context must be restored before executing the return instruction.

The EIRET instruction is used to return from the EI level maskable interrupt handling and the FERET instruction is used to return from FE-level maskable interrupt handling.

When the EIRET or FERET instruction is executed, the CPU performs the following processing and then passes its control to the return PC address:

(1) When returning from the service routine for an EI-level exception, the PC and PSW values on return are loaded from the EIPC and EIPSW registers.
When returning from the service routine for an FE-level exception, the PC and PSW values on return are loaded from the FEPC and FEPSW registers.

(2) Control is passed to the addresses indicated by the return PC and PSW that were loaded.

(3) When EP = 0 and INTCFG.ISPC = 0, the CPU updates the ISPR register.

The flows for returning from exception handling using the EIRET and FERET instructions are shown below.

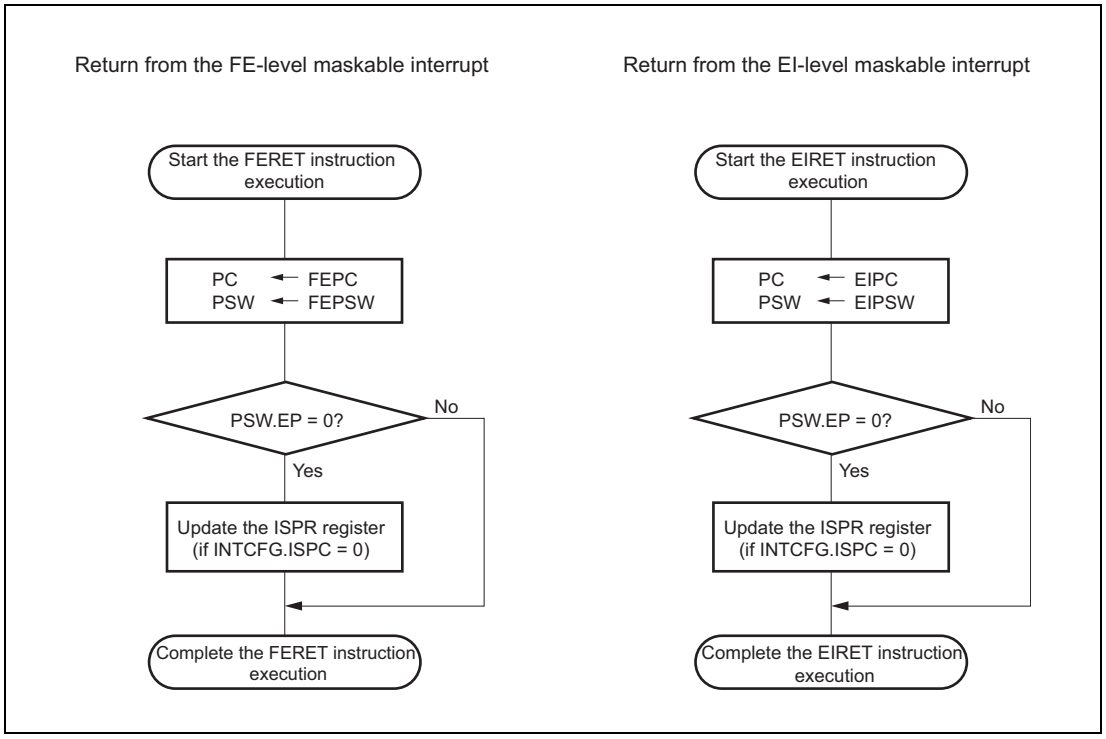


Figure 7.3 Flow of Return from Interrupts

7.9 Interrupt Operation

7.9.1 Interrupt Mask Function of EI Level Maskable Interrupt (EIINT)

Interrupt masking can be specified for each respective interrupt channel of EIINT. Interrupt masking is performed by the following register settings.

Table 7.18 Operation of the MKxxx Bit

ICxxx.MKxxx	Operation
1	Masks interrupt
0	Enables interrupt

The ICxxx.MKxxx bits can also be read and written via the corresponding IMRmEIMKn bits of the IMRm registers. The interrupt mask state is reflected in both the ICxxx registers and the IMRm registers.

[Operation example]

(1) When a 1 is written to an IMRm.IMRmEIMKn bit, interrupts are prohibited for the corresponding channel.

(2) When the corresponding ICxxx.MKxxx bit is read, 1 is returned.

CAUTION

If the MKxxx bit is set to 0 while an interrupt request is pending (RFxxx = 1), the interrupt service routine will be executed at that time (subject to the rules of interrupt prioritization). Even if an interrupt request is issued in software by setting the RFxxx bit to 1, the interrupt will not occur as long as the interrupt is masked with MKxxx = 1.

To cancel an interrupt request that is pending, clear the corresponding RFxxx bit in software.

7.9.2 Interrupt Priority Level Judgment

When FE level non-maskable interrupts (FENMI), FE level maskable interrupts (FEINT), and EI level maskable interrupts (EIINT) are input, priorities including other exceptions are determined, and the exception with the highest priority (including interrupts) is processed. Exceptions occurred at the same time (including interrupts) are processed in a pre-allocated priority order (the default priority order). The priority orders of FENMI, FEINT, and EIINT interrupts are as follows.

FENMI > FEINT > EIINT

See the *RH850G3M User's Manual: Software* for other exceptions.

For EIINT (INTn) interrupts, the interrupt priority can be set independently for each interrupt source. Specify the interrupt priority with the bits P3xxx to P0xxx. The interrupt priority levels can be set from 0 to 15: 0 is the highest and 15 is the lowest. Among multiple EIINT (INTn) interrupts with the same priority level, the interrupt with the lowest interrupt channel number has priority.

Table 7.19 Example of EIINT (INTn) Interrupt Priority Level Settings and Priority Levels During Operation

EIINT (INTn)	ICxxx.P[3:0]xxx Setting	Priority Level During Operation
INT0	3	10
INT1	4	11
INT2	0	1
INT3	0	2
INT4	1	3
INT5	2	6
INT6	2	7
INT7	1	4
INT8	1	5
INT9	2	8
INT10	2	9

The interrupt controller executes multiple interrupt handling when another interrupt request is acknowledged while an interrupt processing is being executed. When multiple EIINT (INTn) interrupts occurs at the same time, the interrupt to be acknowledged is determined by the following procedure.

7.9.2.1 Comparison with the Priority Level of the Interrupt Currently being Handled

Interrupts with the same or lower priority level as the interrupt currently being handled are held pending.

The priority level of the interrupt currently being handled is stored in the ISPR register.

Interrupts with a higher priority level than the interrupt currently being handled proceed to the next priority judgment stage.

7.9.2.2 Masking through Priority Mask Register (PMR)

Only interrupts enabled by the PMR register proceed to the next priority judgment stage.

For the PMR register, see **Table 3.34, PMR Register Contents**, or *the RH850G3M User's Manual: Software*.

7.9.2.3 The Requested Interrupt Source with the Highest Priority Level is Selected

When interrupts are requested simultaneously from multiple sources, the interrupt set the highest priority by ICxxx.P[3:0]xxx bits takes priority.

When there are multiple highest priority interrupts (ICxxx.P[3:0]xxx bits = 0), the lowest interrupt channel number is selected.

7.9.2.4 Interrupt Hold by CPU

Interrupt acknowledgment is held according to the state of the NP and ID bits of the PSW register. At this time, priority judgment among EIINT interrupts, and priority judgment among EIINT, FEINT and FENMI interrupts are performed even while interrupt acknowledgment is pending, and the interrupt with the highest priority is selected when the acknowledgment condition is satisfied.

Example

An EIINT interrupt with the priority level 5 has already been requested and interrupt generation is pending because the value of the PSW.ID bit is 1. If a subsequent EIINT interrupt with the priority

level 3 is requested and the PSW.ID bit is cleared to 0, the latter EIINT interrupt (with the priority level 3) will be generated.

Figure 7.4 shows an example of multiple interrupt handling when another interrupt request is acknowledged while interrupt processing is being executed.

When an interrupt request signal is acknowledged, the PSW.ID flag is automatically set to 1. Therefore, the ID flag should be cleared to 0 to execute multiple interrupt handling. Specifically, execute the EI instruction and the like in an interrupt handling program to enable the interrupt.

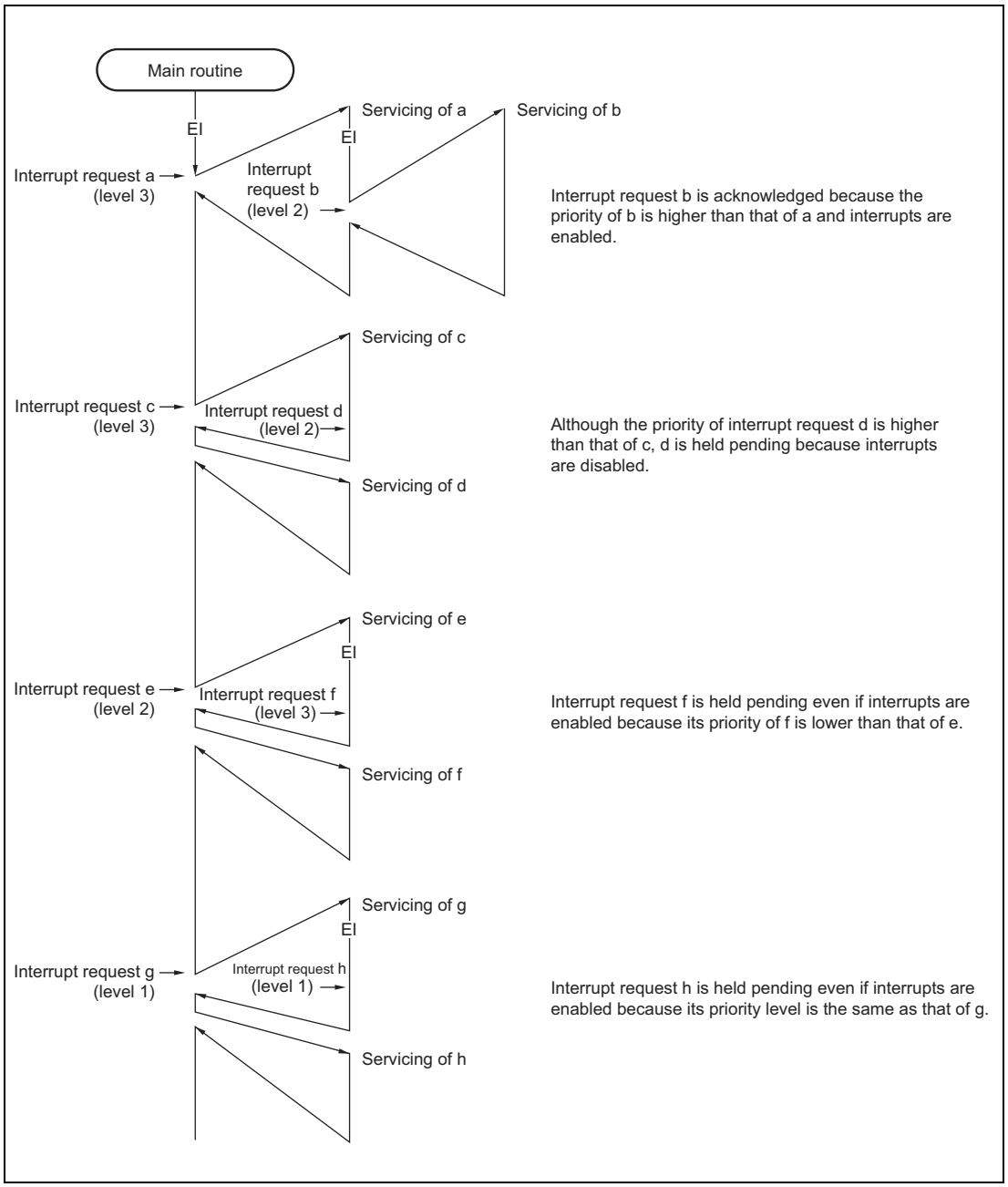
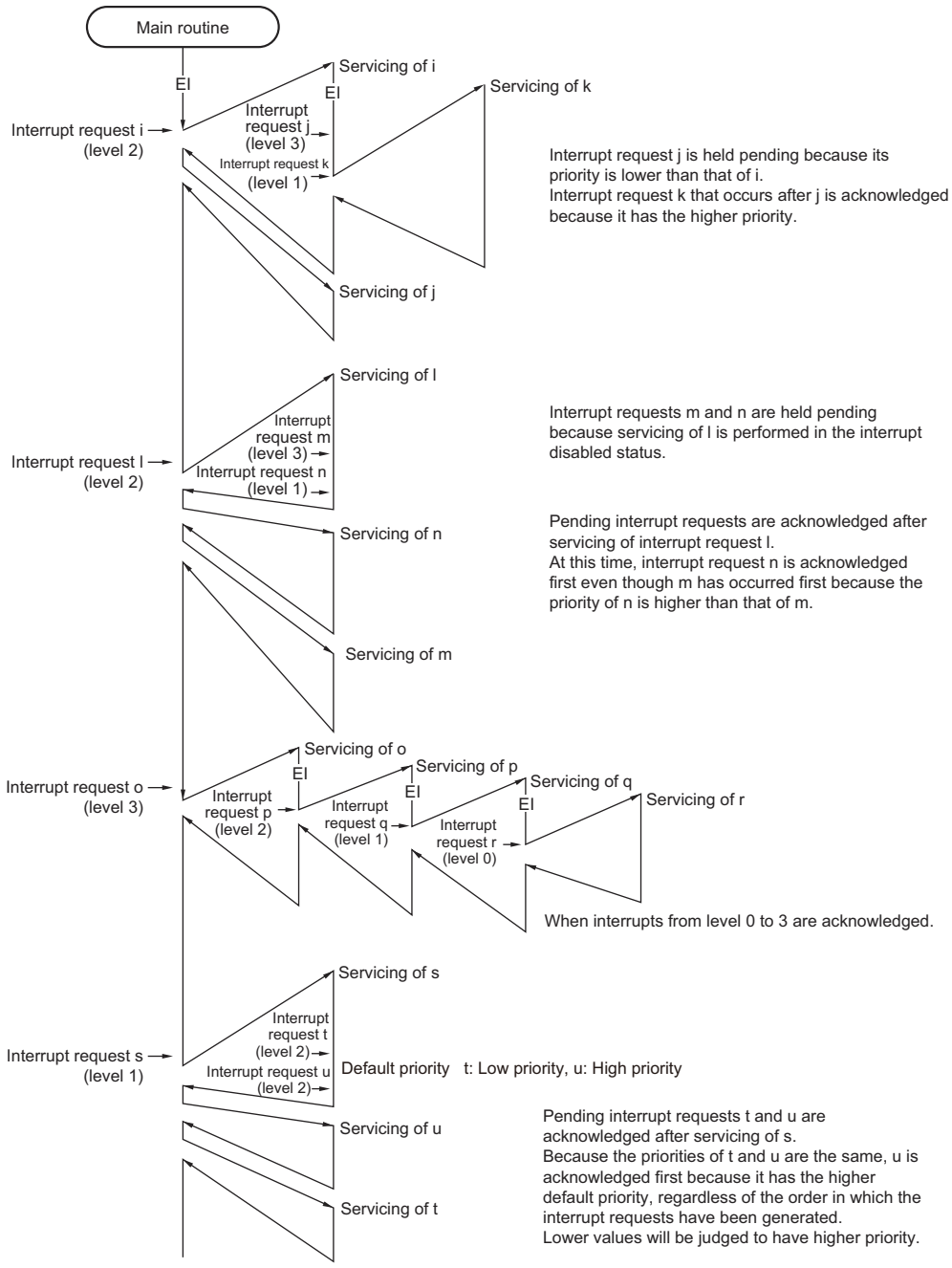


Figure 7.4 Example of Processing in which an Interrupt Request Signal is Issued while Another Interrupt is being Handled (1)



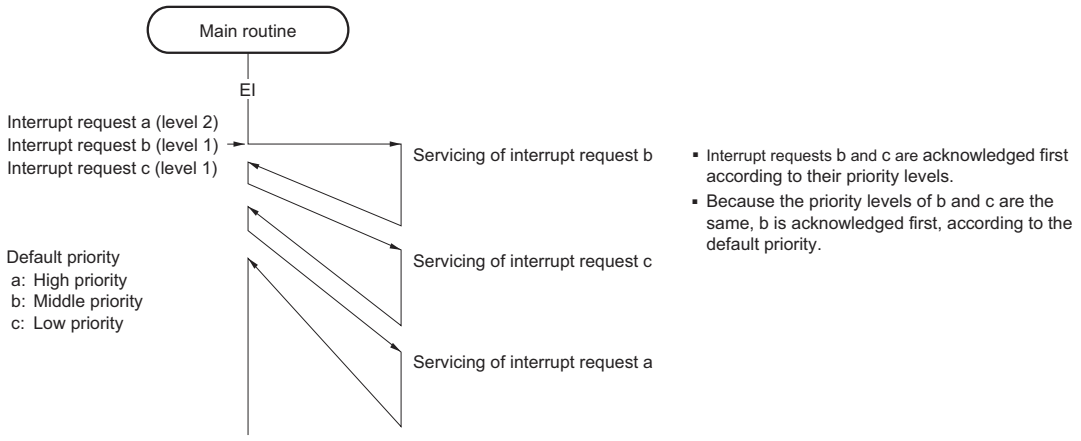
Note 1. “a” to “u” in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

Note 2. The default priority (high/low) in the figure indicates the relative priority between two interrupt request signals.

Figure 7.5 Example of Processing in which an Interrupt Request Signal is Issued while Another Interrupt is being Handled (2)

CAUTION

To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.



Note 1. "a" to "c" in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

Note 2. The default priority (high/low) in the figure indicates the relative priority between two interrupt request signals.

Figure 7.6 Example of Servicing Simultaneously Generated Interrupt Requests

7.9.3 Interrupt Request Acknowledgement Conditions and the Priority

See the *RH850G3M User's Manual: Software*.

7.9.4 Exception Priority of Interrupts and the Priority Mask

See the *RH850G3M User's Manual: Software*.

7.9.5 Interrupt Priority Mask

See the *RH850G3M User's Manual: Software*.

7.9.6 Priority Mask Function

The priority mask function prohibits all EIINT interrupts of the specified interrupt priority level.

The interrupt priority levels to be masked are specified in the PMR register. Masking and acknowledgment can be set for each priority level.

The following operations are possible using this function:

- Temporary prohibition of interrupts that have a priority level that is lower than a given priority level
- Temporary prohibition of interrupts that have a given priority level

Table 7.20 Operation of PMR.PMm bit

PMR.PMm	Operation
0	Acknowledges requests from priority level m interrupt source.
1	Masks requests from priority level m interrupt source.

Note: m = 0 to 15

The presence of EIINT interrupts held pending with this function can be checked with **Section 7.9.7, Exception Management**.

For details on the PMR register, see **Table 3.34, PMR Register Contents**, or the *RH850G3M User's Manual: Software*.

7.9.7 Exception Management

Pending interrupts can be checked in the RH850/F1M. For details, see the *RH850G3M User's Manual: Software*.

7.10 Exception Handler Address

In the RH850/F1M, the exception handler address from which the handler is executed after a reset is input or when an exception or interrupt is acknowledged can be changed according to a setting.

The exception handler address for resets and exceptions (including interrupts) is determined with the direct vector method, in which the reference point of the exception handler address can be changed by using the PSW.EBV bit, the RBASE register, and the EBASE register. For interrupts, the direct vector method and table reference method can be selected for each channel. If the table reference method is selected, execution can branch to the address indicated by the exception handler table allocated in the memory.

CAUTION

- **The exception handler address of EIINT (INTn) selected using the direct vector method differs from that of the V850E2 core products. In the V850E2 core products, a different exception handler address is individually assigned to each interrupt channel (EIINT(INTn)). In the RH850/F1M, one exception handler address is assigned to each interrupt priority. Consequently, interrupts that have the same priority level branch to the same exception handler.**
- **FENMI, FEINT, EIINT (direct vector method) need insertion of SYNCP instruction before the exception handler.**
  For details, see. the *RH850G3M User's Manual: Software*.

7.10.1 Direct Vector Method

The CPU uses the result of adding the offset shown in **Table 7.21, Selection of Base Register/ Offset Address** to the base address indicated by the RBASE or EBASE register as the exception handler address.

Select whether the RBASE or EBASE register is used as the base address by using the PSW.EBV bit*1. When the PSW.EBV bit is set to 1, the value of the EBASE register is used as the base address. When the PSW.EBV bit is cleared to 0, the value of the RBASE register is used as the base address.

For reset input and some exceptions*2, however, the RBASE register is always used for reference.

In addition, user interrupts see the RINT bit of the corresponding base register, and reduce the offset address according to the bit status. If the RBASE.RINT bit or EBASE.RINT bit is set to 1, all user interrupts are handled using an offset of 100_H. If the bit is cleared to 0, the offset address is determined according to **Table 7.21, Selection of Base Register/Offset Address**.

Note 1. Exception acknowledgment itself may sometimes update the status of the PSW.EBV bit. In this case, the base register is selected based on the updated value.

Note 2. The exceptions that always see the RBASE register are determined according to the hardware specifications.

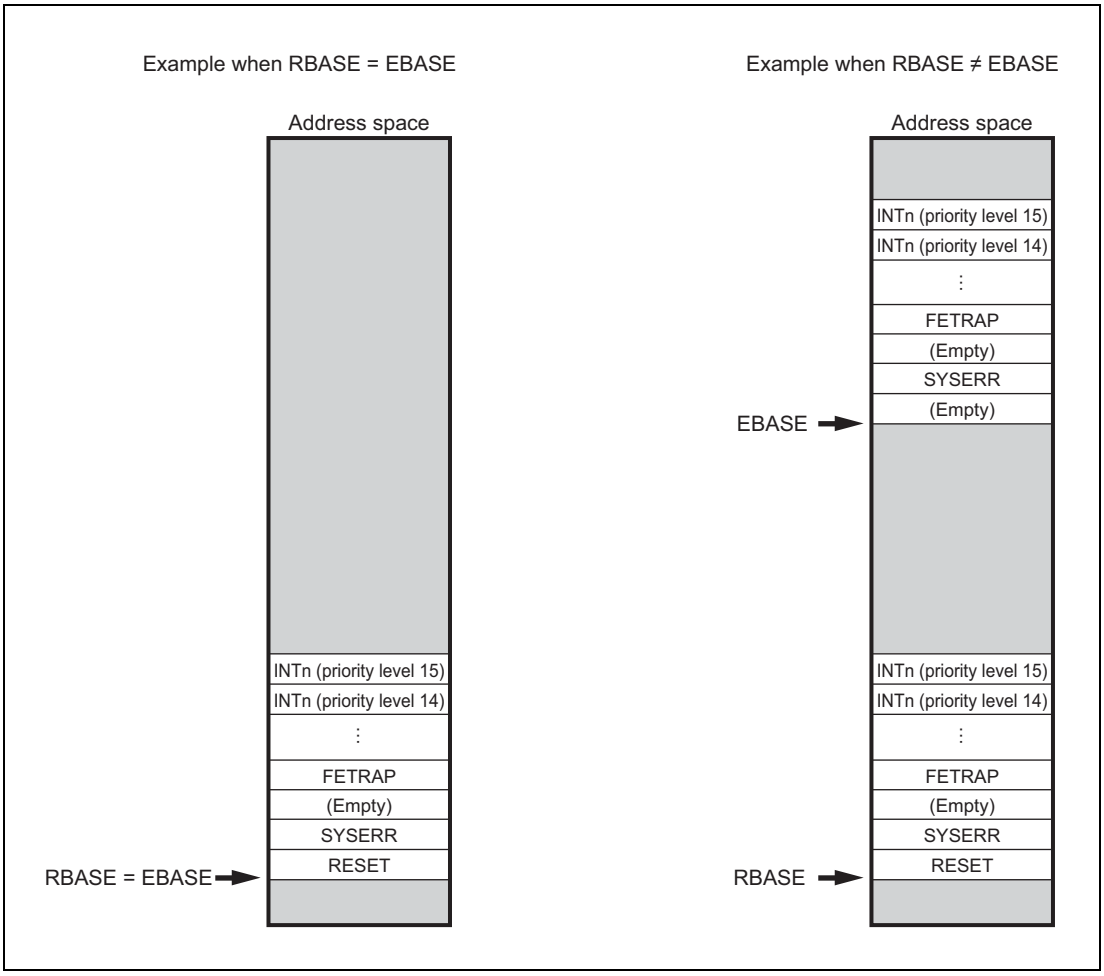


Figure 7.7 Direct Vector Method

The table below shows how base register selection and offset address reduction function for each exception to determine the exception handler address. The value of the PSW bit determines the exception handler address on the basis of the value updated by the acknowledgment of an exception.

Table 7.21 Selection of Base Register/Offset Address

Exception/Interrupt	PSW.EBV = 0	PSW.EBV = 1	RINT = 0	RINT = 1
	Base Register		Offset Address	
RESET	RBASE	N.A.	000_H	000_H
SYSERR		EBASE	010_H	010_H
Reserved			020_H	020_H
FETRAP			030_H	030_H
TRAP0			040_H	040_H
TRAP1			050_H	050_H
RIE			060_H	060_H
FPP/FPI			070_H	070_H
UCPOP			080_H	080_H
MIP/MDP			090_H	090_H
PIE			$0A0_H$	$0A0_H$
Debug			$0B0_H$	$0B0_H$
MAE			$0C0_H$	$0C0_H$
Reserved			$0D0_H$	$0D0_H$
FENMI			$0E0_H$	$0E0_H$
FEINT			$0F0_H$	$0F0_H$
INTn (Priority level 0)			100_H	100_H
INTn (Priority level 1)			110_H	
INTn (Priority level 2)			120_H	
INTn (Priority level 3)			130_H	
INTn (Priority level 4)			140_H	
INTn (Priority level 5)			150_H	
INTn (Priority level 6)			160_H	
INTn (Priority level 7)			170_H	
INTn (Priority level 8)			180_H	
INTn (Priority level 9)			190_H	
INTn (Priority level 10)			$1A0_H$	
INTn (Priority level 11)			$1B0_H$	
INTn (Priority level 12)			$1C0_H$	
INTn (Priority level 13)			$1D0_H$	
INTn (Priority level 14)			$1E0_H$	
INTn (Priority level 15)			$1F0_H$	

Base register selection is used to execute exception handling for resets and some hardware errors by using the programs in a relatively reliable area such as ROM instead of the areas that are easily affected by software errors such as RAM and cache area. The user interrupt offset address reduction function is used to reduce the memory occupation size required by the exception handler for specific system-internal operating modes. The main purpose of this is to minimize the amount of memory consumed in operating modes that use only the minimum functionality, for example, during system maintenance and diagnosis.

7.10.2 Table Reference Method

With the direct vector method, there is one user-interrupt exception handler for each interrupt priority level, and interrupt channels that indicate multiple interrupts with the same priority branch to the same interrupt handler, but some users might want to use different code areas for each interrupt handler from the beginning.

The RH850/F1M uses the table reference method for interrupts that assume the above usage.

If the table reference method is specified as the interrupt channel vector selection method in the interrupt controller and the like, the method for determining the exception handler address when an interrupt request corresponding to that interrupt channel is acknowledged differs as follows.

<1> In any of the following cases, the exception handler address is determined by using the direct vector method:

- When PSW.EBV = 0 and RBASE.RINT = 1
- When PSW.EBV = 1 and EBASE.RINT = 1
- When the interrupt channel setting is not the table reference method

<2> In cases other than <1>, calculate the table reference position.
Exception handler address read position = INTBP register + channel number × 4 bytes

<3> Read word data starting at the interrupt handler address read position calculated in <2>.

<4> Use the word data read in <3> as the exception handler address.

Table 7.22 shows the exception handler address read positions corresponding to each interrupt channel and **Figure 7.8** shows an overview of the allocation in memory.

Table 7.22 Exception Handler Address Expansion

Type of Interrupt	Exception Handler Address Read Position
EI level maskable interrupt channel 0	INTBP register value + 0 × 4
EI level maskable interrupt channel 1	INTBP register value + 1 × 4
EI level maskable interrupt channel 2	INTBP register value + 2 × 4
:	:
EI level maskable interrupt channel 296	INTBP register value + 296 × 4
EI level maskable interrupt channel 297	INTBP register value + 297 × 4

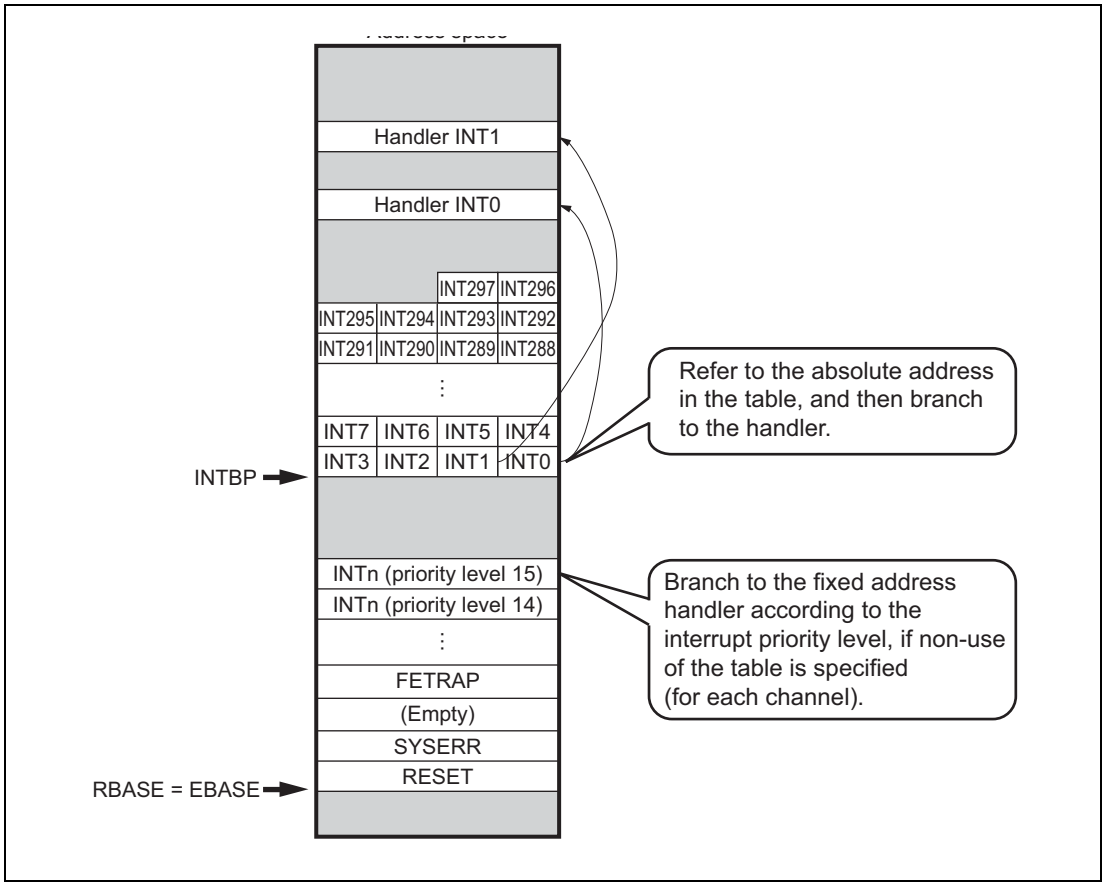


Figure 7.8 Table Reference Method

Section 8 DMA

This section explains about the DMA controller (DMA) in general.

The first part in this section describes the features specific to the RH850/F1M, including the number of channels and register base addresses. The ensuing sections describe the DMA functions and registers.

8.1 Features of RH850/F1M DMA

8.1.1 Number of Channels

The products of the RH850/F1M series incorporate a DMA with the number of channels indicated below.

Table 8.1 Number of Channels

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of channels	16 ch (8 ch × 2)		

Table 8.2 Indices

Index	Description
n	In this section, the DMA unit are identified with an index “n” (n = 0, 1) for example, as DMACn.
m	In this section, the DMA channels are identified with an index “m” (m = 0 to 15). For example, DSAm is the DMA source address register.
i	In this section, the DMAC channels are identified with an index “i” (i = 0 to 7). For example, DMniCM is the DMAC channel master.

8.1.2 Register Base Address

The DMA base address is indicated in the following table.

The DMA register addresses are expressed as an offset of the base address.

Table 8.3 Register Base Address

Base Address Name	Base Address
<DMA_base>	FFFF 8000_H

8.1.3 Interrupt Requests

The DMA interrupt requests are listed in the following table.

Table 8.4 Interrupt Requests

Unit Interrupt Name	Description	Interrupt Number
INTDMA0	DMA00 transfer completion	60
INTDMA1	DMA01 transfer completion	61
INTDMA2	DMA02 transfer completion	62
INTDMA3	DMA03 transfer completion	63
INTDMA4	DMA04 transfer completion	64
INTDMA5	DMA05 transfer completion	65
INTDMA6	DMA06 transfer completion	66
INTDMA7	DMA07 transfer completion	67
INTDMA8	DMA10 transfer completion	68
INTDMA9	DMA11 transfer completion	69
INTDMA10	DMA12 transfer completion	70
INTDMA11	DMA13 transfer completion	71
INTDMA12	DMA14 transfer completion	72
INTDMA13	DMA15 transfer completion	73
INTDMA14	DMA16 transfer completion	74
INTDMA15	DMA17 transfer completion	75

8.1.4 DMA Trigger Factors

DMA trigger factors can be selected by setting the DTFRm.REQSEL[6:0] bits.

The following table lists all DMA trigger factors which can be selected by the DTFRm register.

Table 8.5 DMA Trigger Factors (1/4)

DMA Trigger Number DTFRm.REQSEL[6:0]	DMA Trigger Factor	144 pins	176 pins	233 pins
DMACTRG[0]	INTTAUD0I0	√	√	√
DMACTRG[1]	INTTAUD0I4	√	√	√
DMACTRG[2]	INTTAUD0I8	√	√	√
DMACTRG[3]	INTTAUD0I12	√	√	√
DMACTRG[4]	INTADCA0I0	√	√	√
DMACTRG[5]	INTADCA0I1	√	√	√
DMACTRG[6]	INTADCA0I2	√	√	√
DMACTRG[7]	ADC_CONV_END0	√	√	√
DMACTRG[8]	INTCSIG0IC	√	√	√
DMACTRG[9]	INTCSIG0IR	√	√	√
DMACTRG[10]	INTRLIN30UR0	√	√	√
DMACTRG[11]	INTRLIN30UR1	√	√	√
DMACTRG[12]	INTP0	√	√	√
DMACTRG[13]	INTP2	√	√	√
DMACTRG[14]	INTP4	√	√	√
DMACTRG[15]	INTTAUD0I1	√	√	√
DMACTRG[16]	INTTAUD0I5	√	√	√
DMACTRG[17]	INTTAUD0I9	√	√	√
DMACTRG[18]	INTTAUD0I13	√	√	√
DMACTRG[19]	INTRIIC0TI	√	√	√
DMACTRG[20]	INTRIIC0RI	√	√	√
DMACTRG[21]	INTTAUJ0I0	√	√	√
DMACTRG[22]	INTTAUJ0I3	√	√	√
DMACTRG[23]	Setting prohibited	—	—	—
DMACTRG[24]	Setting prohibited	—	—	—
DMACTRG[25]	Setting prohibited	—	—	—
DMACTRG[26]	Setting prohibited	—	—	—
DMACTRG[27]	Setting prohibited	—	—	—
DMACTRG[28]	INTCSIH1IC	√	√	√
DMACTRG[29]	INTCSIH1IR	√	√	√
DMACTRG[30]	INTCSIH1IJC	√	√	√
DMACTRG[31]	INTP6	√	√	√
DMACTRG[32]	INTP8	√	√	√
DMACTRG[33]	INTTAUB0I0	√	√	√
DMACTRG[34]	INTTAUB0I2	√	√	√
DMACTRG[35]	INTTAUB0I4	√	√	√
DMACTRG[36]	INTTAUB0I6	√	√	√
DMACTRG[37]	INTTAUB0I9	√	√	√
DMACTRG[38]	INTTAUB0I11	√	√	√

Table 8.5 DMA Trigger Factors (2/4)

DMA Trigger Number DTFRm.REQSEL[6:0]	DMA Trigger Factor	144 pins	176 pins	233 pins
DMACTRG[39]	INTTAUB0I13	√	√	√
DMACTRG[40]	INTTAUB0I15	√	√	√
DMACTRG[41]	INTCSIH3IC	√	√	√
DMACTRG[42]	INTCSIH3IR	√	√	√
DMACTRG[43]	INTCSIH3IJC	√	√	√
DMACTRG[44]	INTRLIN32UR0	√	√	√
DMACTRG[45]	INTRLIN32UR1	√	√	√
DMACTRG[46]	INTTAUJ1I0	√	√	√
DMACTRG[47]	INTTAUJ1I2	√	√	√
DMACTRG[48]	Setting prohibited	—	—	—
DMACTRG[49]	Setting prohibited	—	—	—
DMACTRG[50]	INTRLIN34UR0	√	√	√
DMACTRG[51]	INTRLIN34UR1	√	√	√
DMACTRG[52]	INTTAUB1I0	—	√	√
DMACTRG[53]	INTTAUB1I2	—	√	√
DMACTRG[54]	INTTAUB1I4	—	√	√
DMACTRG[55]	INTTAUB1I6	—	√	√
DMACTRG[56]	INTTAUB1I9	—	√	√
DMACTRG[57]	INTTAUB1I11	—	√	√
DMACTRG[58]	INTTAUB1I13	—	√	√
DMACTRG[59]	INTTAUB1I15	—	√	√
DMACTRG[60]	Setting prohibited	—	—	—
DMACTRG[61]	Setting prohibited	—	—	—
DMACTRG[62]	Setting prohibited	—	—	—
DMACTRG[63]	Setting prohibited	—	—	—
DMACTRG[64]	INTTAUD0I2	√	√	√
DMACTRG[65]	INTTAUD0I6	√	√	√
DMACTRG[66]	INTTAUD0I10	√	√	√
DMACTRG[67]	INTTAUD0I14	√	√	√
DMACTRG[68]	Setting prohibited	—	—	—
DMACTRG[69]	Setting prohibited	—	—	—
DMACTRG[70]	INTCSIH0IC	√	√	√
DMACTRG[71]	INTCSIH0IR	√	√	√
DMACTRG[72]	INTCSIH0IJC	√	√	√
DMACTRG[73]	INTP1	√	√	√
DMACTRG[74]	INTP3	√	√	√
DMACTRG[75]	INTP5	√	√	√
DMACTRG[76]	INTTAUD0I3	√	√	√
DMACTRG[77]	INTTAUD0I7	√	√	√
DMACTRG[78]	INTTAUD0I11	√	√	√
DMACTRG[79]	INTTAUD0I15	√	√	√
DMACTRG[80]	INTTAUJ0I1	√	√	√
DMACTRG[81]	INTTAUJ0I2	√	√	√

Table 8.5 DMA Trigger Factors (3/4)

DMA Trigger Number DTFRm.REQSEL[6:0]	DMA Trigger Factor	144 pins	176 pins	233 pins
DMACTRG[82]	Setting prohibited	—	—	—
DMACTRG[83]	Setting prohibited	—	—	—
DMACTRG[84]	Setting prohibited	—	—	—
DMACTRG[85]	INTDMAFL*1	√	√	√
DMACTRG[86]	INTRLIN31UR0	√	√	√
DMACTRG[87]	INTRLIN31UR1	√	√	√
DMACTRG[88]	INTP7	√	√	√
DMACTRG[89]	INTCSIH2IC	√	√	√
DMACTRG[90]	INTCSIH2IR	√	√	√
DMACTRG[91]	INTCSIH2IJC	√	√	√
DMACTRG[92]	INTTAUB0I1	√	√	√
DMACTRG[93]	INTTAUB0I3	√	√	√
DMACTRG[94]	INTTAUB0I5	√	√	√
DMACTRG[95]	INTTAUB0I7	√	√	√
DMACTRG[96]	INTTAUB0I8	√	√	√
DMACTRG[97]	INTTAUB0I10	√	√	√
DMACTRG[98]	INTTAUB0I12	√	√	√
DMACTRG[99]	INTTAUB0I14	√	√	√
DMACTRG[100]	INTTAUJ1I1	√	√	√
DMACTRG[101]	INTTAUJ1I3	√	√	√
DMACTRG[102]	INTP9	√	√	√
DMACTRG[103]	INTADCA1I0	√	√	√
DMACTRG[104]	INTADCA1I1	√	√	√
DMACTRG[105]	INTADCA1I2	√	√	√
DMACTRG[106]	ADC_CONV_END1	√	√	√
DMACTRG[107]	Setting prohibited	—	—	—
DMACTRG[108]	Setting prohibited	—	—	—
DMACTRG[109]	INTCSIG1IC	√	√	√
DMACTRG[110]	INTCSIG1IR	√	√	√
DMACTRG[111]	INTRLIN33UR0	√	√	√
DMACTRG[112]	INTRLIN33UR1	√	√	√
DMACTRG[113]	INTRLIN35UR0	√	√	√
DMACTRG[114]	INTRLIN35UR1	√	√	√
DMACTRG[115]	INTTAUB1I1	—	√	√
DMACTRG[116]	INTTAUB1I3	—	√	√
DMACTRG[117]	INTTAUB1I5	—	√	√
DMACTRG[118]	INTTAUB1I7	—	√	√
DMACTRG[119]	INTTAUB1I8	—	√	√
DMACTRG[120]	INTTAUB1I10	—	√	√
DMACTRG[121]	INTTAUB1I12	—	√	√
DMACTRG[122]	INTTAUB1I14	—	√	√
DMACTRG[123]	Setting prohibited	—	—	—
DMACTRG[124]	Setting prohibited	—	—	—

Table 8.5 DMA Trigger Factors (4/4)

DMA Trigger Number DTFRm.REQSEL[6:0]	DMA Trigger Factor	144 pins	176 pins	233 pins
DMACTRG[125]	Setting prohibited	—	—	—
DMACTRG[126]	Setting prohibited	—	—	—
DMACTRG[127]	Setting prohibited	—	—	—

Note 1. For details, see the *RH850/F1L, RH850/F1M, RH850/F1H Flash Memory User's Manual: Hardware Interface*.

8.2 Overview

8.2.1 Overview

Direct memory access (DMA) is used to access data without intervention of the CPU.

DMA includes a DMA transfer module called DMAC. A DMAC includes registers for storing transfer information. DMA has two 8-channel DMAC modules.

In this manual, DTFR denotes the function to select among hardware DMA transfer sources for a DMAC and retain the DMA transfer request. The DTFR can handle 128 types of hardware DMA transfer sources.

The address space that can be used for DMA transfer is a 4 GB address space represented by a 32-bit address. For information about which resource is assigned to a particular area in the 4 GB address space and which area is accessible from DMA, see **Section 4.2.2, Data Space Accessible by DMA**.

CAUTION

DMA can be used after PEG unset.

When accessing from the DMA to Local RAM, the PEGGnBA register (n = 0 to 3) and DMniCM register (ni = 00 to 07, 10 to 17) must be set.

8.2.2 Term Definition

Table 8.6 shows the terms used in this section.

Table 8.6 List of Term Definitions (1/2)

Term	Meaning
DMA transfer	A general term for data transfer carried out by DMA.
DMA cycle	A series of actions that consist of reading an amount of data specified by the transfer size (8/16/32/64/128 bits) from the address specified by the source address and writing it to the address specified by the destination address. The first half of the DMA cycle (reading part) is called a read cycle, and the second half (writing part) is called a write cycle.
Hardware DMA transfer source	A trigger for a DMA transfer request given by an internal peripheral device.
Hardware DMA transfer request	A DMA transfer request generated by a hardware DMA transfer source
Software DMA transfer request	A DMA transfer request generated by software writing to a register.
DMA transfer request	A trigger to start DMA transfer with a DMAC.
Transfer information	The information required for DMA transfer, including the source address, destination address, transfer data size, and transfer count.
Single transfer	DMA transfer consisting of one DMA cycle started by one DMA transfer request.
Block transfer 1	DMA transfer consisting of the number of DMA cycles specified by the transfer count in the transfer information, started by one DMA transfer request.
Block transfer 2	DMA transfer consisting of the number of DMA cycles specified by the address reload count in the transfer information, started by one DMA transfer request.
Block transfer	A general term for both block transfer 1 and block transfer 2.
Last transfer	The DMA cycle carried out when the transfer count in the transfer information is 1.
Address reload transfer	The DMA cycle carried out when the address reload count in the transfer information is 1 if the reload function 2 is used.

Table 8.6 List of Term Definitions (2/2)

Term	Meaning
Suspension	An action of pausing DMA transfer during block transfer. You can resume DMA transfer after suspension.
Resume	An action of resuming suspended DMA transfer.
Transfer abort	An action of aborting DMA transfer in the middle. You cannot resume DMA transfer after that.

8.3 DMA Function

8.3.1 Basic Operation of DMA Transfer

8.3.1.1 Transfer Mode

DMA has three transfer modes.

Single Transfer

One DMA cycle is executed when a DMA transfer request is accepted.

Block Transfer 1

The number of DMA cycles specified in the transfer count register are executed when a DMA transfer request is accepted.

Block Transfer 2

The number of DMA cycles specified by the address reload count are executed when a DMA transfer request is accepted. If the value specified in the address reload count is larger than the value in the transfer count register, the number of DMA cycles specified in the transfer count register are executed.

8.3.1.2 Executing a DMA Cycle

DMA always executes a write cycle after a read cycle is complete.

For example, if the transfer data size is 128 bits, a write cycle is executed after a read cycle for the 128-bit data is finished. A write cycle never starts in the middle of a read cycle.

8.3.1.3 Updating Transfer Information

When a DMA cycle is executed, DMA updates transfer information as follows:

Source Address and Destination Address

Transfer information will be updated as described in **Table 8.7** according to the settings in the transfer control register such as the count directions of source address and destination address and transfer data size.

Table 8.7 Updating the Source Addresses and the Destination Addresses

Count Direction	Transfer Data Size	Address after Update
Increment	8 bits	(address before update) + 0000_0001_H
	16 bits	(address before update) + 0000_0002_H
	32 bits	(address before update) + 0000_0004_H
	64 bits	(address before update) + 0000_0008_H
	128 bits	(address before update) + 0000_0010_H
Decrement	8 bits	(address before update) - 0000_0001_H
	16 bits	(address before update) - 0000_0002_H
	32 bits	(address before update) - 0000_0004_H
	64 bits	(address before update) - 0000_0008_H
	128 bits	(address before update) - 0000_0010_H
Fixed	—	Same as the address before update.

When you use the reload function, a special update rule is applied other than the one described in **Table 8.7** for the last transfer and the address reload transfer. For details, see **Section 8.3.3, Reload Function**.

Transfer Count/Address Reload Count

The transfer count is decremented by one for every DMA cycle.

The address reload count is decremented by one for every DMA cycle when the reload function 2 or block transfer 2 is used. When the reload function 2 or block transfer 2 is not used, it is not updated.

When you use the reload function, a special update rule is applied for the last transfer and the address reload transfer. For details, see **Section 8.3.3, Reload Function**.

Other transfer information

Other transfer information is not updated during execution of a DMA cycle.

8.3.1.4 Last Transfer and Address Reload Transfer

The last transfer means a DMA cycle executed when the value in the transfer count register, which shows the remaining number of transfers, is one. The last transfer differs in operation compared to other DMA cycles as follows.

- The transfer completion flag (DCSTm.TC) is set when the last transfer is complete. The channel operation enable (DCENm.DTE) bit is cleared when the last transfer is complete (when the continuous transfer is disabled).
- When the transfer completion interrupt output enable is set, a transfer completion interrupt is output when the last transfer is complete.
- When the reload function 1 is enabled, the reload function 1 is executed at the timing of the last transfer. For details, see **Section 8.3.3, Reload Function**.

The address reload transfer means a DMA cycle executed when the reload function 2 is enabled and the address reload count is one. The reload function 2 is executed during the address reload transfer. For details, see **Section 8.3.3, Reload Function**.

8.3.1.5 Transfer Completion Interrupt Output

DMA can output a transfer completion interrupt to external devices.

Transfer Completion Interrupt Output

When the transfer completion interrupt output enable (DTCTm.TCE) is set in the transfer control register, a DMAC requests a transfer completion interrupt when the last transfer is complete.

Figure 8.1 shows the operation of the transfer completion interrupt.

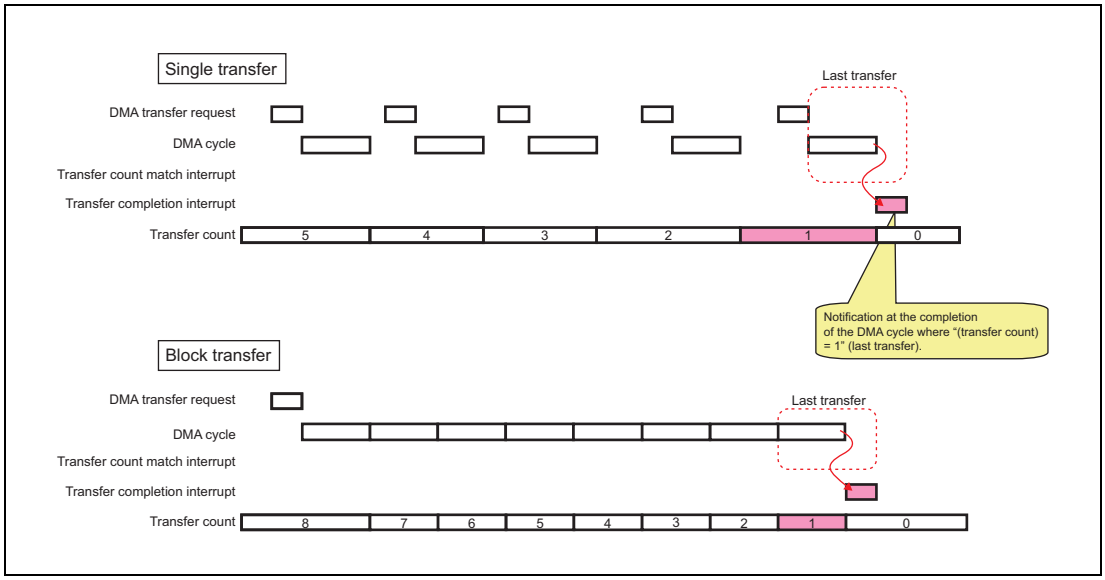


Figure 8.1 Transfer Completion Interrupt

8.3.1.6 Continuous Transfer

If the continuous transfer is not used, a DMAC sets the transfer completion flag (DCSTm.TC) and clears the channel operation enable (DCENm.DTE) bit when the last transfer is complete. In this case, a DMA transfer request is not accepted when the request is generated after the completion of the last transfer.

If the continuous transfer is used, the channel operation enable (DCENm.DTE) bit is not cleared when the last transfer is complete, and a DMA transfer request can be accepted even when the transfer completion flag is set. If DMA is used for a case where a specified number of DMA transfers are executed repetitively, software overhead associated with clearing the transfer completion flag and setting the channel operation enable bit after the completion of the last transfer can be reduced by using the continuous transfer.

The continuous transfer is enabled by setting the continuous transfer enable (DTCTm.MLE) in the DMAC transfer control register.

The continuous transfer is designed to work with the reload function 1. The continuous transfer function itself does not update the source address register, destination address register, and transfer count register. If, after the last transfer is complete, you want to restore the source address register, destination address register, and transfer count register to the state before the DMA transfer starts, use the reload function 1 and set the values of the source address register, destination address register, and the transfer count register before the DMA transfer starts to the reload source address register, reload destination address register, and reload transfer count register respectively.

Figure 8.2 shows an operation of continuous transfer by a DMAC.

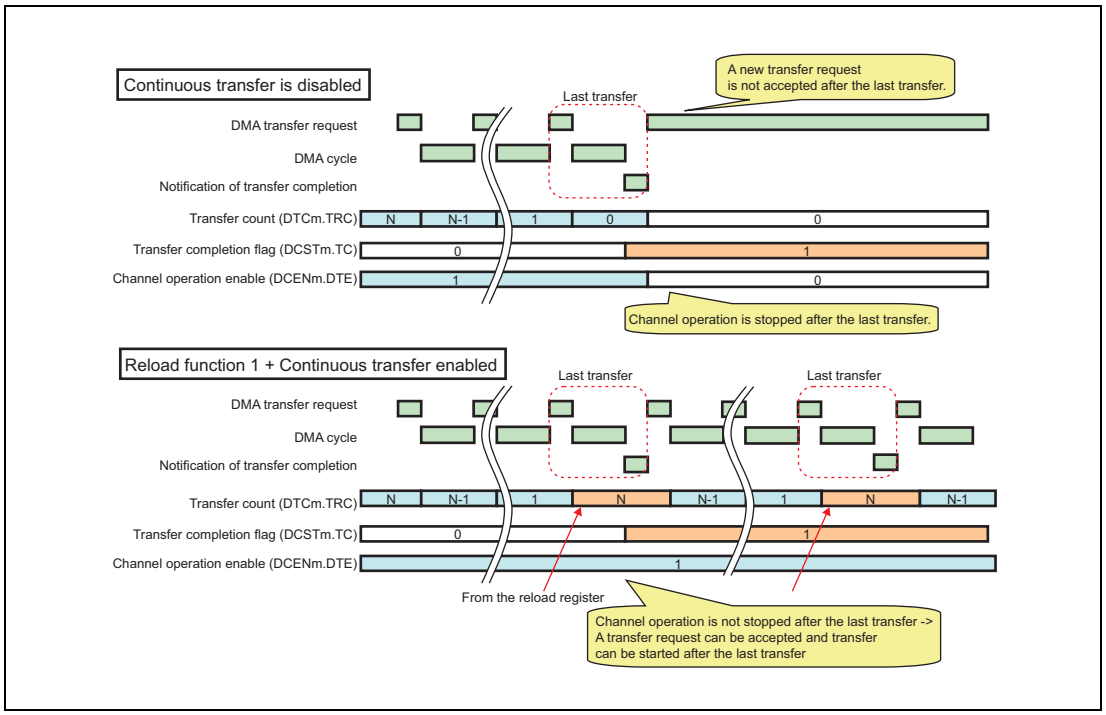


Figure 8.2 Operation of Continuous Transfer by a DMAC

8.3.2 Channel Priority Order

This subsection explains arbitration between multiple DMA channels.

8.3.2.1 DMAC Channel Arbitration

A DMAC select one channel out of eight channels with arbitration. Arbitration is done according to the fixed priority order. The priority order is “channel 0 > channel 1 > channel 2 > channel 3 > channel 4 > channel 5 > channel 6 > channel 7” for DMAC0, “channel 8 > channel 9 > channel 10 > channel 11 > channel 12 > channel 13 > channel 14 > channel 15” for DMAC1.

Arbitration is done for every DMA cycle. No arbitration occurs between the read and write of a DMA cycle.

If, at the timing when one DMA cycle completes in the middle of a block transfer of a channel, there is a DMA transfer request from a channel with a higher priority, a DMA cycle of the channel with the higher priority will be executed as the result of arbitration.

If a DMAC executes the block transfer 1 or block transfer 2, DMAC channel arbitration is done for every DMA cycle, and possibly a DMA cycle of another DMAC channel with a higher priority may interrupt.

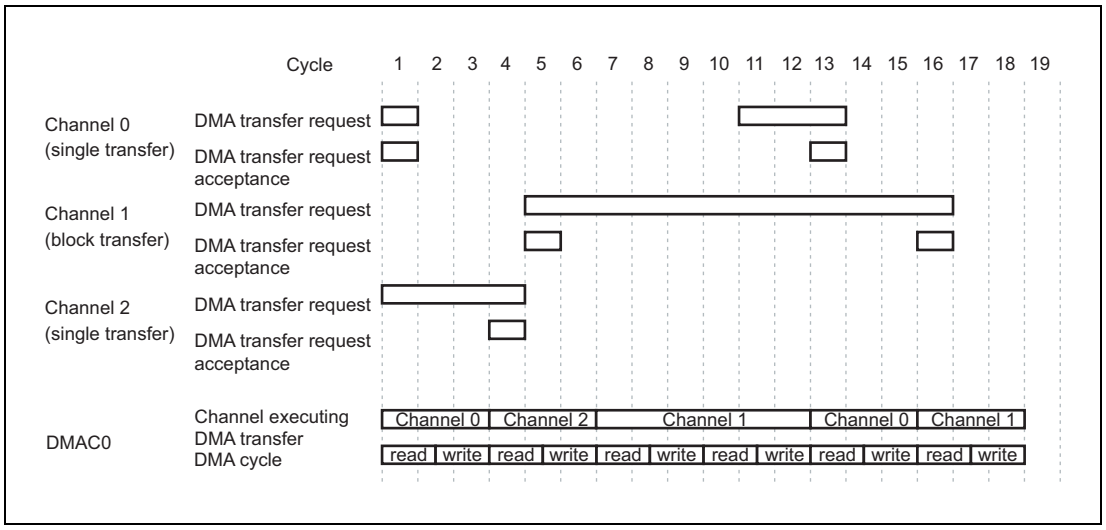


Figure 8.3 DMAC Channel Arbitration

Cycle numbers shown in **Figure 8.3** are for explanation purpose only. They do not indicate an actual number of cycles necessary for executing DMA transfer.

In **Figure 8.3**, DMA transfer requests for channels 0 and 2 are generated at Cycle 1. As a result of arbitration, a DMA cycle for channel 0 starts because its priority is higher. At Cycle 4, a DMA cycle for channel 2 starts. At Cycle 5, a DMA transfer request for channel 1 is generated. However, since the DMA cycle for channel 2 is still ongoing, no arbitration is done at this point. At Cycle 7, a DMA cycle for channel 1 starts. Because channel 1 uses block transfer, this DMA cycle continues at Cycle 10 where there are no other DMA transfer request from other channel. At Cycle 11, a DMA transfer request for channel 0 is generated. However, since the DMA cycle for channel 1 is still ongoing, no arbitration is done at this point. At Cycle 12, the DMA cycle for channel 1 is complete. At Cycle 13, a DMA cycle for channel 0 starts as a result of arbitration between DMA channels 0 and 1. It should be noted that, even though a block transfer of channel 1 has been already started, a DMA cycle of not channel 1 but channel 0 is executed at Cycle 13 because the priority of the latter is higher. At Cycle 15,

the DMA cycle for channel 0 is complete. At Cycle 16, a DMA cycle for channel 1 starts again. At Cycle 18, the last DMA cycle of the block transfer of channel 1 is complete.

8.3.2.2 Interface Arbitration

DMAC0 and DMAC1 work independently and execute DMA transfer.

8.3.3 Reload Function

8.3.3.1 Overview of the Reload Function

The reload function updates a portion of transfer information, more specifically, the source address, destination address, transfer count, and address reload count, to the predefined values during DMA transfer.

The reload function has two types of functions: reload function 1 and reload function 2.

8.3.3.2 Operation of Reload Function 1

When the reload function 1 is enabled, actions described in **Table 8.8** are executed at the timing of the last transfer according to the reload function 1 setting.

Table 8.8 Operation of Reload Function 1

Reload Function 1 Setting (DTCTm.RLD1M[1:0])	Register	Action at the Last Transfer
00 (Reload function 1 disabled.)	Source address	Not reloaded.
	Destination address	Not reloaded.
	Transfer count	Not reloaded.
	Address reload count	Not reloaded.
01 (Reload function 1 enabled. Reloading source address and transfer count.)	Source address	The reload source address is copied to this.
	Destination address	Not reloaded.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	• If the reload function 2 is disable: Not reloaded. • If the reload function 2 is enabled: The reload address reload count is copied to this.
10 (Reload function 1 enabled. Reloading destination address and transfer count.)	Source address	Not reloaded.
	Destination address	The reload destination address is copied to this.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	• If the reload function 2 is disable: Not reloaded. • If the reload function 2 is enabled: The reload address reload count is copied to this.
11 (Reload function 1 enabled. Reloading source address, destination address, and transfer count.)	Source address	The reload source address is copied to this.
	Destination address	The reload destination address is copied to this.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	• If the reload function 2 is disable: Not reloaded • If the reload function 2 is enabled: The reload address reload count is copied to this.

Figure 8.4 shows an operation of the reload function 1.

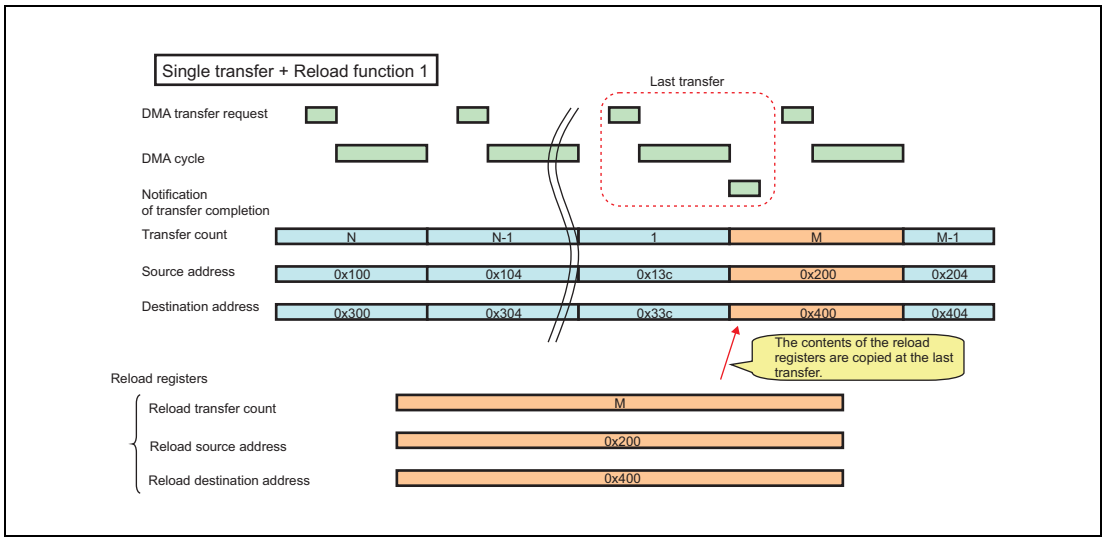


Figure 8.4 Operation of Reload Function 1

8.3.3.3 Reload Function 2

When the reload function 2 is enabled, actions described in **Table 8.9** are executed at the timing of the address reload transfer according to the reload function 2 setting.

Table 8.9 Operation of Reload Function 2

Reload Function 2 Setting (DTCTm.RLD2M[1:0])	Register	Action at the Address Reload Transfer
00 (Reload function 2 disabled.)	Source address	Not reloaded.
	Destination address	Not reloaded.
	Address reload count	Not reloaded.
01 (Reload function 2 enabled. Reloading source address and reload address reload count.)	Source address	The reload source address is copied to this.
	Destination address	Not reloaded.
	Address reload count	The reload address reload count is copied to this.
10 (Reload function 2 enabled. Reloading destination address and reload address reload count.)	Source address	Not reloaded.
	Destination address	The reload destination address is copied to this.
	Address reload count	The reload address reload count is copied to this.
11 (Reload function 2 enabled. Reloading source address and reload address reload count.)	Source address	The reload source address is copied to this.
	Destination address	The reload destination address is copied to this.
	Address reload count	The reload address reload count is copied to this.

Figure 8.5 shows an operation of the reload function 2.

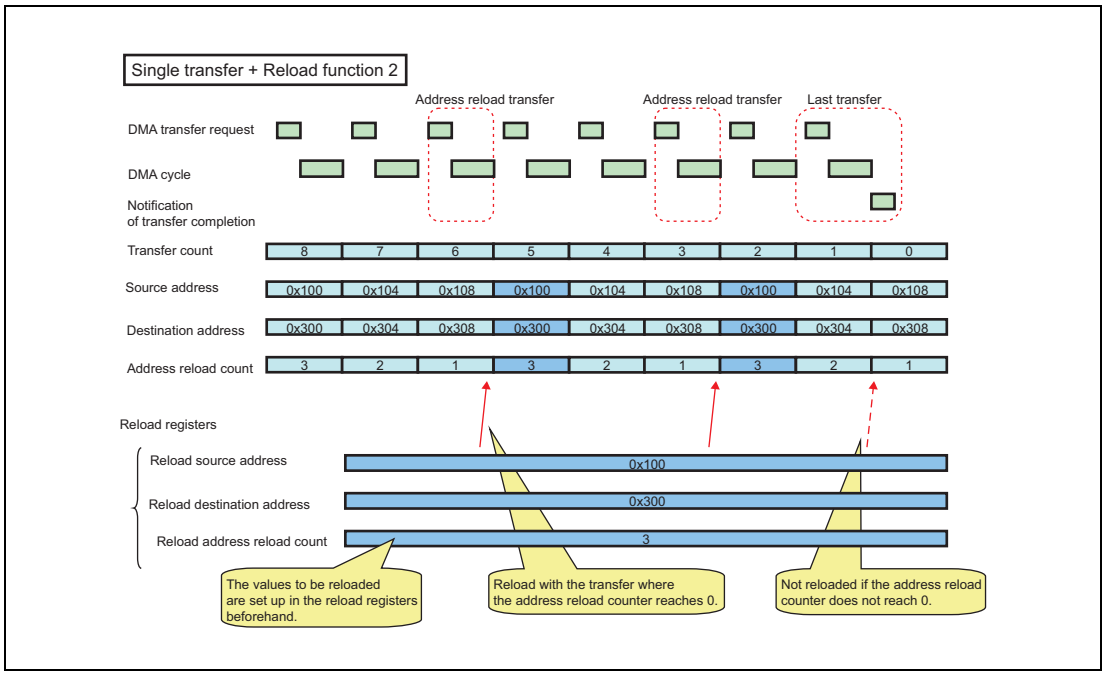


Figure 8.5 Operation of Reload Function 2

Figure 8.6 shows an operation when both the reload function 1 and the reload function 2 are used simultaneously.

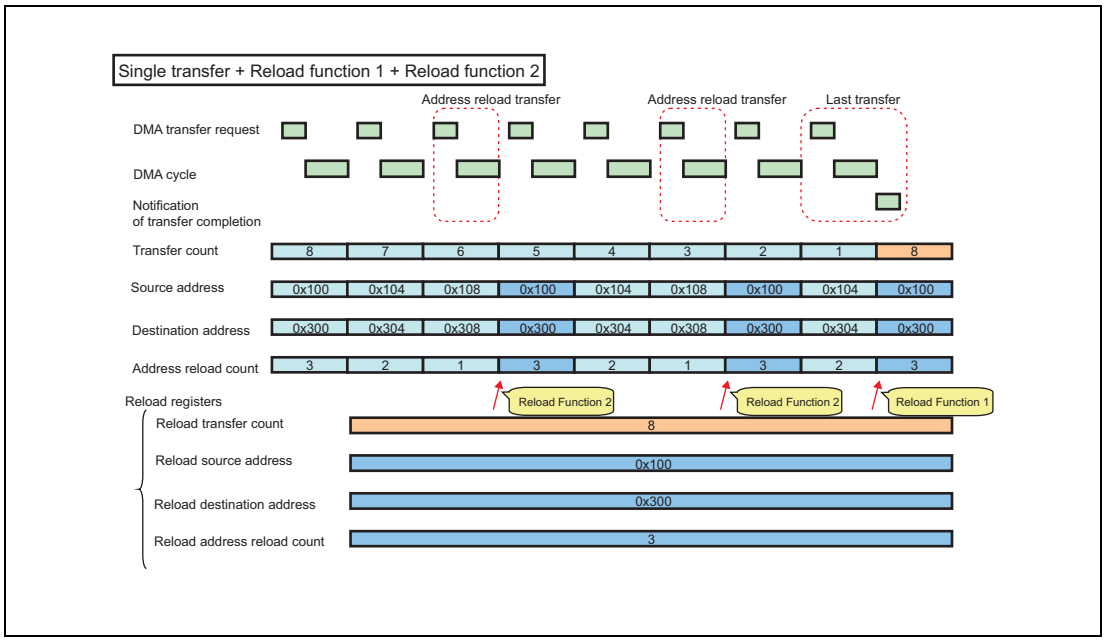


Figure 8.6 Operation when combining the reload function 1 and the reload function 2

8.3.3.4 Timing of Setting DMAC Reload Registers

You can set up the reload source address register, reload destination address register, and reload transfer count register any time (even during DMA transfer). However, if you update the reload source address register, reload destination address register, and reload transfer count register during DMA transfer, there may be a conflict between reloading at the last transfer or address reload transfer and update of the reload registers by user. In order to avoid this conflict, setting up of the reload registers must be completed before the last transfer or address reload transfer starts.

8.3.4 Chain Function

8.3.4.1 Overview

DMA offers a function called chain function. With this function, the completion of the DMA cycle or last transfer for one channel can trigger a DMA transfer request for another channel.

A DMA transfer request for another channel initiated by the chain function is called a chain request.

You can select the condition for generating a chain request from the following two options:

- Always chain: A chain request is generated at the completion of every DMA cycle.
- Chain at the last transfer: A chain request is generated at the completion of the last transfer.

Figure 8.7 shows an operation of “always chain.”

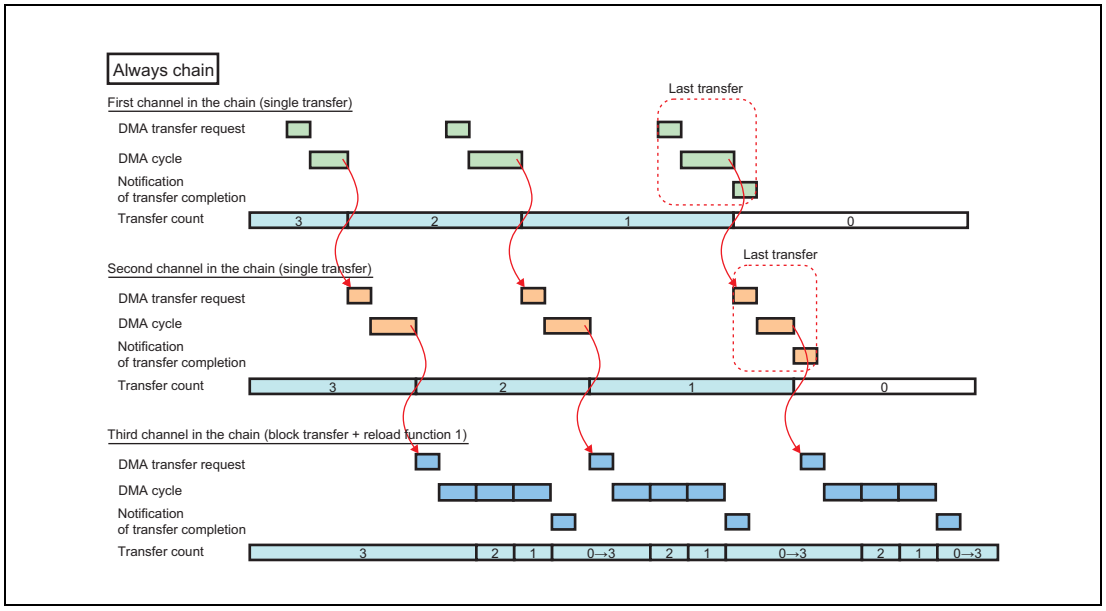


Figure 8.7 Operation of “Always Chain”

Figure 8.8 shows an operation of “chain at the last transfer”.

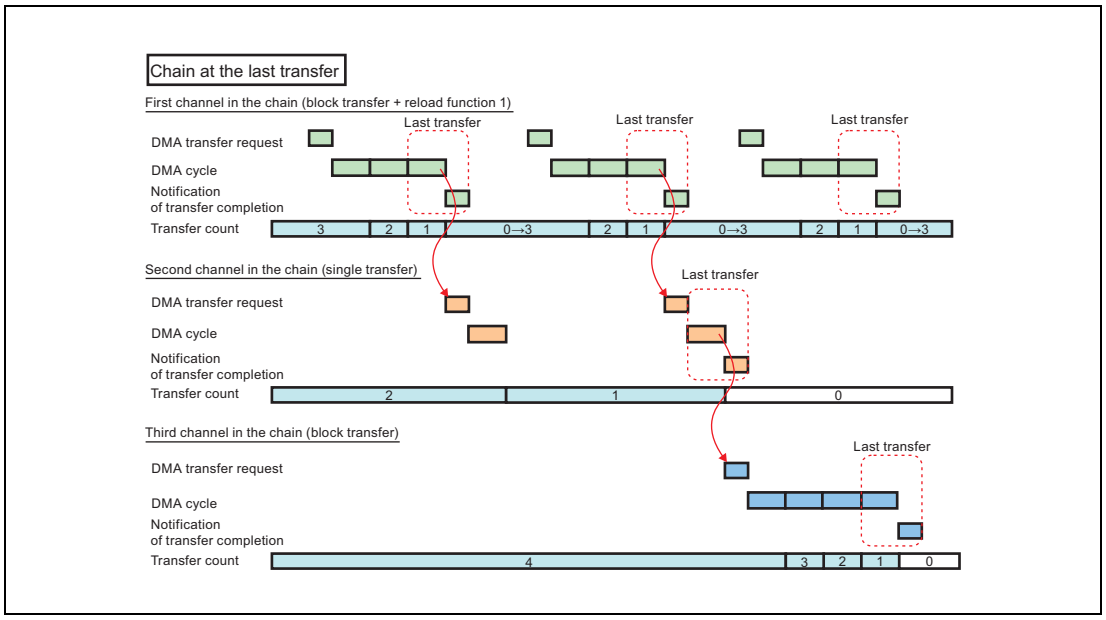


Figure 8.8 Operation of “Chain at the Last Transfer”.

8.3.4.2 Setting Up the Chain Function

For a DMAC, you need to write to the chain enable (DTCTm.CHNE) and the next channel in the chain selection (DTCTm.CHNSEL) in the DMAC transfer control register in order to set up the type of chain function and the next channel number in the chain.

8.3.4.3 Caution for Using the Chain Function

The chain function is enabled by setting the software DMA transfer request flag of the next channel in the chain as a part of its function. Therefore, you need to set up the channel settings of the next channel in the chain in the same way as when the software DMA transfer request is used. If you specify a channel using the hardware DMA transfer request for the next channel in the chain, the chain function does not work.

A channel and its next channel in the chain must belong to the same module (DMAC0 and DMAC1). You cannot specify a channel in another module for its next channel in the chain.

8.3.5 DMAC Operation

8.3.5.1 Types of DMA Transfer Requests and Assigning DMA Transfer Requests

A DMAC starts DMA transfer by accepting a hardware DMA transfer request or software DMA transfer request. The DMA transfer request selection assignment (DTCTm.DRS) bit in the DMAC transfer control register (DTCTm) determines whether a hardware DMA transfer request or a software DMA transfer request is used.

In the case of a hardware DMA transfer request for a DMAC, DTFR selects one out of 128 hardware DMA transfer factor and assigns to each channel of the DMAC. This assignment is configured in the DTFR setting registers.

8.3.5.2 Generating and Accepting a Software DMA Transfer Request

By setting the software DMA transfer request flag (DCSTm.SR) in the DMAC transfer status register (DCSTm) using the DMAC transfer status set register (DCSTSm), a software DMA transfer request can be generated.

The software DMA transfer request flag is automatically cleared when the DMAC processes the DMA transfer request. The timing when the software DMA transfer request flag is automatically cleared differs depending on the transfer mode of the DMA transfer to be executed.

- In the single transfer mode, the software DMA transfer request flag is cleared whenever the software DMA transfer request is accepted.
- In the block transfer 1 mode, the software DMA transfer request flag is cleared when the last transfer starts.
- In the block transfer 2 mode, the software DMA transfer request flag is cleared when the last transfer or address reload transfer starts.

The software DMA transfer request flag can also be cleared by software using the DMAC transfer status clear register (DCSTCm). When you abort a DMA transfer of a DMAC channel, you must clear the software DMA transfer request flag.

8.3.5.3 Executing DMA Transfer

When the DMAC accepts a DMA transfer request for a channel, the DMAC executes DMA transfer of the channel. If there are DMA transfer requests from multiple channels, DMAC does DMAC channel arbitration and decides a channel to be acknowledged.

8.4 Suspension, Resume, Transfer Abort, and Clearing of a DMA Transfer Request

8.4.1 DMA Suspension and Resume by Software Control

The DMA control register (DMACTL) is used to suspend DMA transfer for all channels.

When the DMA suspension bit (DMACTL.DMASPD) in the DMA control register is set, DMA puts all channels into the suspended state. If all channels are in the suspended state and the DMA suspension bit in the DMA control register is cleared, DMA restores all channels from the suspended state to the normal state and resumes the DMA transfer of the suspended channel.

When all channels are in the suspended state, DMA transfer is suspended for all channels without changing the value of the DCENm.DTE bit of each DMAC channel.

8.4.2 Suspension, Resume, and Transfer Abort of a DMAC Channel

You can suspend the DMA transfer of a DMAC channel by clearing the channel operation enable bit (DCENm.DTE) in the DMAC channel operation enable setting register for the channel. If a DMA cycle is ongoing, the DMA transfer of the channel is suspended after the currently ongoing DMA cycle is finished. If you set the DCENm.DTE bit again while the DMA transfer of the channel is suspended, the DMA transfer of the channel is resumed.

If you want to abort the currently ongoing DMA transfer of a DMAC channel, similarly clear the channel operation enable bit (DCENm.DTE) in the DMAC channel operation enable setting register, and then clear the hardware DMA transfer request in the DTFR in the case of a hardware DMA transfer request, or clear the software DMA transfer request flag (DCSTm.SR) using the DMAC transfer request flag clear bit (DCSTCm.SRC) in the DMAC transfer status clear register in the case of a software DMA transfer request.

In case that the continuous transfer enable bit (DTCTm.MLE) is set, the channel operation enable bit (DCENm.DTE) is kept to be set. Even though the channel operation enable bit (DCENm.DTE) is cleared by software during a DMA cycle in a last transfer, the function of the continuous transfer enable bit (DTCTm.MLE) is given high priority and the channel operation enable bit (DCENm.DTE) is set after completion of the last transfer. If you want to abort an ongoing DMA transfer of a DMAC channel when continuous transfer function is enabled, please clear the continuous transfer enable bit (DTCTm.MLE) first and then clear the channel operation enable bit (DCENm.DTE) to abort DMA transfer of the DMAC channel. Only for the operation, DMAC Transfer Control Register (DTCTm) can be written under the channel operation is enabled (DCENm.DTE = 1).

Figure 8.9 shows an example of suspension, resume, and transfer abort of a DMAC channel.

In **Figure 8.9**, both channels 0 and 1 execute block transfer. At time tick 1, DMA transfer of channel 1 starts. At time tick 2, a DMA transfer request for channel 0 is accepted. As a result of DMAC channel arbitration, DMA transfer of channel 0 starts because channel 0 has a higher priority than channel 1. At time tick 3, the last transfer of channel 0 is complete, and the remaining DMA transfer in the block transfer of channel 1 starts. At time tick 4, the last transfer of channel 1 is complete. After time tick 5, DMA transfer of channel 0 and DMA transfer of channel 1 are executed similarly. At time tick 7, the DMA transfer of channel 0 is suspended and, as a result of DMAC channel arbitration, the DMA transfer of channel 1 starts. At time tick 8, the last transfer of channel 1 is complete, and then, at time tick 9, the DMA transfer of channel 0 resumes. At time tick 10, the DMA transfer of channel 0 is suspended again, and then, at time tick 11, the DMA transfer of channel 0 is aborted. At time tick 12, the suspended state for channel 0 is cleared, but the DMA transfer of channel 0 is not executed because the DMA transfer is aborted at time tick 11.

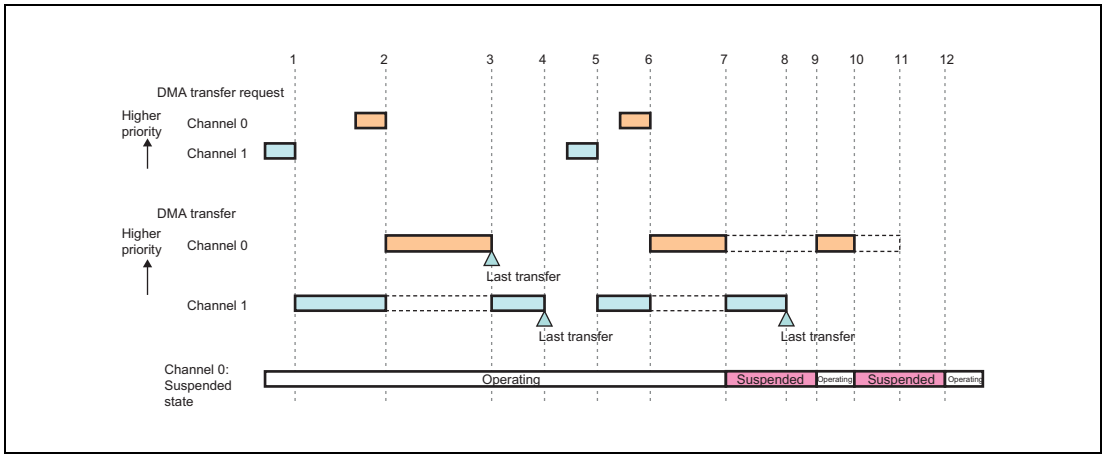


Figure 8.9 Example of Suspension, Resume, and Transfer Abort of a DMAC Channel

8.4.3 Masking and Clearing a Hardware DMA Transfer Request by the DTFR

If a DMAC uses a hardware DMA transfer request, you can temporarily disable (mask) the hardware DMA transfer request output from the DTFR to the DMAC by clearing the hardware DMA transfer source selection enable bit (DTFRm.REQEN) in the DTFR setting register.

Also, if a hardware DMA transfer source is used, you can clear a hardware DMA transfer request retained in the DTFR by using the hardware DMA transfer request clear (DTFRRQCm.DRQC) bit in the DTFR transfer request clear register.

Even if you suspend or abort DMA transfer of a DMAC channel, the hardware DMA transfer request selection/hold circuit of the DTFR is still running, and consequently, the DTFR may retain a hardware DMA transfer request that was input to the DTFR during the suspension or transfer abort period of the DMAC channel. When you resume or start DMA transfer of a DMAC channel, clear the hardware DMA transfer request retained in the DTFR as required.

When a DMAC is used with hardware transfer request and block transfer 1 or 2, if the hardware transfer source selection enable bit is disabled (DTFRm.REQEN = 0) by software during the DMAC block transfer, the ongoing block transfer is aborted.

8.4.4 List of Suspend, Resume, and Transfer Abort Functions

Table 8.10 List of Suspend, Resume, and Transfer Abort Functions

Function	How to execute the function	Operation	Possibility of DMA transfer abort	Master that can execute the function (See Section 8.6, Reliability Function.)
DMA suspension and resume by software control	Setting and clearing the DMACCTL.DMASPD	All channels are in the suspended state.	Not possible*1	Special master
Suspension and resume of a DMAC channel	Clearing and setting the DCENm.DTE in each channel register*2	DMA transfer of a channel is suspended.	Possible (by clearing the DMA transfer request flag in suspension state)	Special master, and general master assigned to the channel.

Note 1. In order to abort DMA transfer, you need to abort transfer for the DMAC channel.

Note 2. In case that the continuous transfer enable bit (DTCTm.MLE) is set, please clear (or set) the continuous transfer enable bit (DTCTm.MLE) first.

8.5 Error Control

8.5.1 Type of Error

DMA can generate the following type of error.

- DMA Transfer Error
  This error is generated when error is detected in the read cycle or write cycle in a DMA cycle.
  This error can be generated in all DMAC channels during execution of DMA transfer.

8.5.2 DMA Transfer Error

8.5.2.1 Operation of a DMAC When DMA Transfer Error Occurs

When DMA transfer error occurs in a DMAC, the transfer error flag (DCSTm.ER) in the DMAC transfer status register of the channel where the DMA transfer error occurred is set. The DMAC error register (DMACER) shows the transfer error flags of all 16 DMAC channels.

On a channel where the transfer error flag is set, a new DMA cycle is not executed if the DMA transfer disable on transfer error setting (DTCTm.ESE) bit is set. On the other hand, a DMA cycle is executed regardless of the value of the transfer error flag if the transfer error case DMA transfer disable setting (DTCTm.ESE) bit is cleared.

If you want to abort the DMA transfer of a channel where the DMA transfer error occurred is set, follow the procedure to abort DMA transfer of the DMAC channel.

If DMA transfer error occurs during the read cycle of a DMA cycle, the write cycle is not executed. If DMA transfer error occurs during the write cycle of a DMA cycle, the validity of the result of write operation is not guaranteed.

Regardless of whether DMA transfer error occurs in the read cycle or write cycle of a DMA cycle, the source address register, destination address register, transfer count register, and address reload count register are updated.

8.6 Reliability Function

8.6.1 Overview

In this product, DMA is a resource used by multiple masters. In order for DMA to support multi-core configuration, the following reliability functions are offered:

- Register access protection function
- Master information inherit function

8.6.2 Register Access Protection Function

The register access protection function allows access to the transfer information of each DMA channel only from the master assigned to the channel and prohibits access from other masters.

The register access protection function enables you, for example, to prevent the settings of the channel from being read or updated by masters other than the one assigned to the channel.

8.6.2.1 Identifying the Accessing Master

DMA identifies a master based on the processor element ID number (PEID) of the accessing master, and whether the CPU is in the supervisor mode (PSW.UM=0) or the user mode (PSW.UM=1).

8.6.2.2 Master Access

The following two types of master access methods can be used:

- Special master access (CPU1 supervisor mode (UM = 0))
- General master access (all accesses other than the above special master access)

The master can access all registers in special master access mode.

In master access, access to the following registers is allowed.

- The following global registers
  DMACER
- Channel registers of the channels assigned by the channel assignment (For details, see **Section 8.6.2.3, Channel Assignment**.)

In general master access, access to registers other than the above is not allowed.

8.6.2.3 Channel Assignment

To each channel, DMA can assign a master so that the master is allowed to use the channel. Channel assignment is configured in the channel master setting register (DMniCM in the case of a DMAC) by the CPU1 in the supervisor mode (UM = 0).

In general master access, the master assigned to a channel by the channel assignment is allowed to access the channel registers of the channel. If the channel registers of a channel is accessed by a master other than the master assigned to the channel, the access is called illegal access. For information about illegal access, see **Section 8.6.2.4, Illegal Access**.

8.6.2.4 Illegal Access

DMA handles the following access as illegal access:

(a) General master access to the global registers
Except for the following registers: DMACER

(b) General master access to the channel registers by a master other than the master assigned to the channel

DMA's actions against illegal access are as follows:

For both cases (a) and (b),

- Write access is ignored.
- Read access returns 0 as read data.

Only for the case (b),

- The information about the illegal access is stored in a register access protection violation register.
- The DMAC0 and DMAC1 have their own register access protection violation registers (DM0CMV and DM1CMV respectively).

Only the special master can access the register access protection violation registers. The special master can check whether illegal access has occurred by checking the register access protection violation registers periodically.

In addition, it is recommended that, when a master uses DMA and configures transfer information in the channel registers, the master should check whether the configuration has been successfully completed without illegal access by, for example, reading back the settings.

8.6.3 Master Information Inherit Function

In this product, DMA inherits master information that is equivalent to the master information of the CPU to which the DMA channel is set.

The master information that is output from DMA is as in **Table 8.11**.

Table 8.11 Master Information That Is Output from DMA

Meaning	Value that is output from DMA
UM	UM bit value in the channel master setting register
SPID	SPID bit value in the channel master setting register
PEID	PEID bit value in the channel master setting register
DMA	1

8.6.4 Other Reliability Functions

8.6.4.1 Restriction on the Next Channel in the Chain

The reliability function limits the channels you can select as the next channel in the chain.

When you use the chain function, the channel master settings of a channel and its next channel in the chain must be the same.

The chain function is designed so that a channel and its next channel in the chain are managed by the same master.

When DMA detects that different masters are assigned to a channel and its next channel in the chain, it is deemed illegal and the chain function is suppressed. More specifically, when DMA tries to execute the chain function, DMA compares the chain master settings of the channel and its next channel in the chain, and if the settings are the same for both PEID and UM, the chain function is allowed and a chain request is sent to the next channel. If the channel master settings are not the same for either PEID or UM, a chain request is not sent.

8.7 Setting Up DMA Transfer

8.7.1 Overview of Setting Up DMA

Table 8.12 Overview of Setting Up DMA

No.	Master that Configures the Setting	Description	Register		Necessity of the Setting
1	Special master (CPU1 in the supervisor mode (UM = 0))	Overall DMA operation setting	DM00CM to DM17CM	DMAC channel master setting	Mandatory
2		Status clear	CMVC	Channel protection violation clear register	Recommended
3	Master assigned to the DMAC channel	Channel setting	DSAm	DMAC source address	Mandatory
4			DDAm	DMAC destination address	Mandatory
5			DTCm	DMAC transfer count	Mandatory
6			DTCTm	DMAC transfer control	Mandatory
7			DRSAm	DMAC reload source address	Mandatory if the reload function is used
8			DRDAm	DMAC reload destination address	Mandatory if the reload function is used
9			DRTCm	DMAC reload transfer count	Mandatory if the reload function is used
10			DTFRm	DTFR setting register	Mandatory
11		Status clear	DCSTCm	DMAC transfer status clear	Mandatory
12			DTFRRQCm	DTFR transfer request clear	Recommended
13		Channel operation enable	DCENm	DMAC channel operation enable setting	Mandatory

8.7.2 Setting Up the Overall DMA Operation

You need to set up the overall DMA operation before you start using DMA.

To configure the overall DMA operation, the special master (a CPU1 in the supervisor mode (UM = 0)) needs to set up global registers. Global registers can be set up only by special master access. For details, see **Section 8.6, Reliability Function**.

The following register must be set up to configure the overall DMA operation.

- DMAC channel master setting registers (DMniCM)

Those registers configure channel assignment. (For details, see **Section 8.6, Reliability Function**.)

If the DMAC channel master setting registers are not properly set, DMA channel setting and DMA transfer cannot be executed properly.

Also, if errors are detected in the following registers while the overall DMA operation is set up, clearing the errors is recommended.

- DMAC0 register access protection violation register (DM0CMV)
- DMAC1 register access protection violation register (DM1CMV)

8.7.3 Setting Up the DMA Channel Setting

The DMA channel setting defines the transfer information and transfer source for each DMAC.

To configure the DMA channel setting, each channel's master assigned by the channel assignment sets the channel registers.

8.7.3.1 Setting Up the DMAC Channel Setting

Follow the procedure below to set up the DMAC channel setting in case of using DMAC.

(1) Disabling the DMAC Channel Operation

If the channel operation enable (DCENm.DTE) in the DMAC channel operation enable setting register (DCENm) is set, clear the DCENm.DTE bit to disable the channel operation.

(2) Setting Up the Transfer Information

When you set up the transfer information of the DMAC, the following registers need to be set up.

- DMAC source address register (DSAm)
- DMAC destination address register (DDAm)
- DMAC transfer count register (DTCm)
- DMAC transfer control register (DTCTm)
- DMAC reload source address register (DRSAm)
- DMAC reload destination address register (DRDAm)
- DMAC reload transfer count register (DRTCm)

(3) Setting Up the DMA Transfer Request

While setting the transfer information, you need to set up the DMA transfer request selection assignment (DTCTm.DRS) bit in the DMAC transfer control register (DTCTm) to define whether the hardware or software DMA transfer request is used.

You cannot use both the hardware and software DMA transfer requests for the same channel at the same time.

If you use the hardware DMA transfer request, you need to select the source used as the hardware DMA transfer request out of 128 hardware DMA transfer sources using the hardware DMA transfer source selection (DTFRm.REQSEL) in the DTFR setting register. Also, you need to enable the hardware DMA transfer source selection (DTFRm.REQEN) in the same register.

The DTFR may retain a hardware DMA transfer request before the hardware DMA transfer source is selected. Clear the hardware DMA transfer request (DTFRRQm.DRQ) retained in the DTFR using the DTFR transfer request clear register (DTFRRQCm) if necessary.

If you use the software DMA transfer request, disable the hardware DMA transfer source selection (DTFRm.REQEN) in the DTFR setting register.

(4) Clearing the Transfer Status

The DMAC transfer status register (DCSTm) may retain the result of the previous DMA transfer, so clear the flags in the DMAC transfer status register using the DMAC transfer status clear register (DCSTCm).

(5) Enabling the DMAC Channel Operation

Set the channel operation enable (DCENm.DTE) bit in the DMAC channel operation enable setting register to enable the channel operation.

After the channel operation enable bit is set, the DMAC can accept a DMA transfer request and become ready for DMA transfer.

8.8 Global Register

8.8.1 List of Global Register Addresses

The global register addresses are listed in the table below.

For <DMA_base>, see **Section 8.1.2, Register Base Address**.

Table 8.13 List of Global Register Addresses

Unit Name	Register Name	Register Symbol	Address	Access Permission	
				Special Master	General Master
DMA	DMA control register	DMACTL	<DMA_base> + 0000_H	√	—
	DMAC error register	DMACER	<DMA_base> + 0020_H	√	√
	DMAC0 register access protection violation register	DM0CMV	<DMA_base> + 0030_H	√	—
	DMAC1 register access protection violation register	DM1CMV	<DMA_base> + 0034_H	√	—
	Register access protection violation clear register	CMVC	<DMA_base> + $003C_H$	√	—
	DMAC0 channel0 channel master setting	DM00CM	<DMA_base> + 0100_H	√	—
	DMAC0 channel1 channel master setting	DM01CM	<DMA_base> + 0104_H	√	—
	DMAC0 channel2 channel master setting	DM02CM	<DMA_base> + 0108_H	√	—
	DMAC0 channel3 channel master setting	DM03CM	<DMA_base> + $010C_H$	√	—
	DMAC0 channel4 channel master setting	DM04CM	<DMA_base> + 0110_H	√	—
	DMAC0 channel5 channel master setting	DM05CM	<DMA_base> + 0114_H	√	—
	DMAC0 channel6 channel master setting	DM06CM	<DMA_base> + 0118_H	√	—
	DMAC0 channel7 channel master setting	DM07CM	<DMA_base> + $011C_H$	√	—
	DMAC1 channel0 channel master setting	DM10CM	<DMA_base> + 0120_H	√	—
	DMAC1 channel1 channel master setting	DM11CM	<DMA_base> + 0124_H	√	—
	DMAC1 channel2 channel master setting	DM12CM	<DMA_base> + 0128_H	√	—
	DMAC1 channel3 channel master setting	DM13CM	<DMA_base> + $012C_H$	√	—
	DMAC1 channel4 channel master setting	DM14CM	<DMA_base> + 0130_H	√	—
	DMAC1 channel5 channel master setting	DM15CM	<DMA_base> + 0134_H	√	—
	DMAC1 channel6 channel master setting	DM16CM	<DMA_base> + 0138_H	√	—
	DMAC1 channel7 channel master setting	DM17CM	<DMA_base> + $013C_H$	√	—

8.8.2 Details of Global Registers

8.8.2.1 DMACTL — DMA Control Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + 0000_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMA SPD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.14 DMACTL Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DMASPD	DMA suspension This bit indicated that DMA transfer for all channels is suspended. If a user writes 1 to this bit, DMA transfer for all channels can be suspended. If a user writes 0 to this bit, suspension of DMA transfer for all channels can be released. The suspension controlled by this bit is independent from the suspension controlled by the transfer enable bit (DCENm.DTE) of each DMAC channel. That means, if this bit is set to 1, all DMA transfers are suspended regardless of the values of the DCENm.DTE bit of each DMAC channel. Writing to this bit does not affect the DCENm.DTE bit of each DMAC channel. 0: DMA suspension cleared 1: DMA suspension request/DMA suspension ongoing

8.8.2.2 DMACER — DMAC Error Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMA_base> + 0020_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM1ER7	DM1ER6	DM1ER5	DM1ER4	DM1ER3	DM1ER2	DM1ER1	DM1ER0	DM0ER7	DM0ER6	DM0ER5	DM0ER4	DM0ER3	DM0ER2	DM0ER1	DM0ER0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.15 DMACER Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 8	DM1ER[7:0]	DMAC1 DMA transfer error status These bits show the DMA transfer error status of channels 0 through 7 of the DMAC1. Each bit is mapped from the DCSTm.ER bit of each channel of the DMAC1 and is read-only. 0: DMA transfer error is not generated 1: DMA transfer error is generated
7 to 0	DM0ER[7:0]	DMAC0 DMA transfer error status These bits show the DMA transfer error status of channels 0 through 7 of the DMAC0. Each bit is mapped from the DCSTm.ER bit of each channel of the DMAC0 and is read-only. 0: DMA transfer error is not generated 1: DMA transfer error is generated

8.8.2.3 DM0CMV — DMAC0 Register Access Protection Violation Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMA_base> + 0030_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	MINF[6:1]						—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	VCH[2:0]			—	—	—	VF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.16 DM0CMV Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned.
22 to 17	MINF[6:1]	Illegal access master information These bits retain the accessing master information of the first illegal access after the DM0CMV.VF bit is cleared to 0. If illegal access occurs while the DM0CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. DM0CMV.MINF[6:1] retains the following accessing master information. DM0CMV.MINF[6:4]: Access source PEID DM0CMV.MINF[3:2]: Access source SPID DM0CMV.MINF[1]: Access source UM
16 to 7	Reserved	When read, the value after reset is returned.
6 to 4	VCH[2:0]	Channel where an illegal access occurred These bits retain the channel number (0 to 7) of the first illegal access after the DM0CMV.VF bit is cleared to 0. If illegal access occurs while the DM0CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
3 to 1	Reserved	When read, the value after reset is returned.
0	VF	Illegal access flag This bit shows whether illegal access occurred in the DMAC0. 0: No illegal access has occurred in the DMAC0 1: Illegal access has occurred in the DMAC0 If illegal access occurs in the DMAC0 while this bit is 0, this bit is set, and DM0CMV.MINF[6:1] and DM0CMV.VCH[2:0] store their respective information. If illegal access occurs in the DMAC0 while this bit is 1, this bit remains 1, and DM0CMV.MINF[6:1] and DM0CMV.VCH[2:0] do not change. This bit can be cleared by using the CMVC register.

8.8.2.4 DM1CMV — DMAC1 Register Access Protection Violation Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMA_base> + 0034_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	MINF[6:1]						—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	VCH[2:0]			—	—	—	VF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.17 DM1CMV Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned.
24 to 17	MINF[6:1]	Illegal access master information These bits retain the accessing master information of the first illegal access after the DM1CMV.VF bit is cleared to 0. If illegal access occurs while the DM1CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. DM1CMV.MINF[6:1] retains the following access source master information. DM1CMV.MINF[6:4]: Access source PEID DM1CMV.MINF[3:2]: Access source SPID DM1CMV.MINF[1]: Access source UM
16 to 7	Reserved	When read, the value after reset is returned.
6 to 4	VCH[2:0]	Channel where an illegal access occurred These bits retain the channel number (0 to 7) of the first illegal access after the DM1CMV.VF bit is cleared to 0. If illegal access occurs while the DM1CMV.VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
3 to 1	Reserved	When read, the value after reset is returned.
0	VF	Illegal access flag This bit shows whether illegal access occurred in the DMAC1. 0: No illegal access has occurred in the DMAC1 1: Illegal access has occurred in the DMAC1 If illegal access occurs in the DMAC1 while this bit is 0, this bit is set, and DM1CMV.MINF[6:1] and DM1CMV.VCH[2:0] store their respective information. If illegal access occurs in the DMAC1 while this bit is 1, this bit remains 1, and DM1CMV.MINF[6:1] and DM1CMV.VCH[2:0] do not change. This bit can be cleared by using the CMVC register.

8.8.2.5 CMVC — Register Access Protection Violation Clear Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + $003C_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DM1VC	DM0VC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 8.18 CMVC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DM1VC	DMAC1 illegal access flag clear The DMAC1 illegal access flag (DM1CMV.VF) can be cleared by writing 1 to this bit. When read, this bit is always read as 0.
0	DM0VC	DMAC0 illegal access flag clear The DMAC0 illegal access flag (DM0CMV.VF) can be cleared by writing 1 to this bit. When read, this bit is always read as 0.

8.8.2.6 DMniCM — DMAC Channel Master Setting (ni = 00 to 07 and 10 to 17)

Access: This register can be read or written in 32-bit units.

Address: DM0iCM: <DMA_base> + 0100_H + 4_H × Ch. No. i (i = 0 to 7)
DM1iCM: <DMA_base> + 0120_H + 4_H × Ch. No. i (i = 0 to 7)

Value after reset: $0000\ 0010_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	PEID[2:0]			SPID[1:0]		UM	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 8.19 DMniCM Register Contents

Bit Position	Bit Name	Function ion
31 to 7	Reserved	When read, the value after reset is returned.
6 to 4	PEID[2:0]	Channel master PEID setting Specifies the PEID information of the master assigned to the channel.
3, 2	SPID[1:0]	Channel master SPID setting Specifies the SPID information used by the master assigned to the channel.
1	UM	Channel master UM setting 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode.
0	Reserved	When read, the value after reset is returned.

CAUTION

DM00CM to DM07CM configure the channel master information of the DMAC0 channel 0 to 7 respectively.

DM10CM to DM17CM configure the channel master information of the DMAC1 channel 0 to 7 respectively.

For information about the functions this register offers, see **Section 8.6, Reliability Function**.

8.9 DMAC Channel Register

8.9.1 DMAC Channel Register Address

The DMAC channel registers are listed in the table below.

For <DMA_base>, see **Section 8.1.2, Register Base Address**.

Table 8.20 DMAC Channel Register Addresses

Unit Name	Register Name	Register Symbol	Address	Accessed Permission	
				Special Master	General Master
DMA	DMAC source address	DSAm	<DMA_base> + 0400_H + 40_H × [channel number]	√	√
	DMAC destination address	DDAm	<DMA_base> + 0404_H + 40_H × [channel number]	√	√
	DMAC transfer count	DTCm	<DMA_base> + 0408_H + 40_H × [channel number]	√	√
	DMAC transfer control	DTCTm	<DMA_base> + $040C_H$ + 40_H × [channel number]	√	√
	DMAC reload source address	DRSAm	<DMA_base> + 0410_H + 40_H × [channel number]	√	√
	DMAC reload destination address	DRDAm	<DMA_base> + 0414_H + 40_H × [channel number]	√	√
	DMAC reload transfer count	DRTCm	<DMA_base> + 0418_H + 40_H × [channel number]	√	√
	DMAC channel operation enable setting	DCENm	<DMA_base> + 0420_H + 40_H × [channel number]	√	√
	DMAC transfer status	DCSTm	<DMA_base> + 0424_H + 40_H × [channel number]	√	√
	DMAC transfer status set	DCSTSm	<DMA_base> + 0428_H + 40_H × [channel number]	√	√
	DMAC transfer status clear	DCSTCm	<DMA_base> + $042C_H$ + 40_H × [channel number]	√	√
	DTFR setting	DTFRm	<DMA_base> + 0430_H + 40_H × [channel number]	√	√
	DTFR transfer request status	DTFRRQm	<DMA_base> + 0434_H + 40_H × [channel number]	√	√
	DTFR transfer request clear	DTFRRQCm	<DMA_base> + 0438_H + 40_H × [channel number]	√	√

Note 1. The [channel number] in the offset addresses and “m” in the register symbols are numbers in the range from 0 to 15, and the correspondence is as follows.

Channel number m	Channel	Channel number m	Channel
0	DMAC0 channel 0	8	DMAC1 channel 0
1	DMAC0 channel 1	9	DMAC1 channel 1
2	DMAC0 channel 2	10	DMAC1 channel 2
3	DMAC0 channel 3	11	DMAC1 channel 3
4	DMAC0 channel 4	12	DMAC1 channel 4
5	DMAC0 channel 5	13	DMAC1 channel 5
6	DMAC0 channel 6	14	DMAC1 channel 6
7	DMAC0 channel 7	15	DMAC1 channel 7

8.9.2 Details of DMAC Channel Registers

The “m” in the register symbols indicates the DMA channel number (m = 0 to 15).

8.9.2.1 DSAm — DMAC Source Address Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + 0400_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.21 DSAm Register Contents

Bit Position	Bit Name	Function
31 to 0	SA[31:0]	Source address Specifies the DMA transfer source address. Those bits are updated whenever a DMA cycle is executed. If you read from those bits, the transfer source address for the next DMA cycle is read.

CAUTIONS

1. It is forbidden to write to those bits when the channel operation is enabled (DCENm.DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the source address is updated.
3. DMA transfer for misaligned data is not supported. The lower 4 bits of the address corresponding to each transfer data size is as follows. (× denotes an arbitrary one bit.)
   The correct operation is not guaranteed if you select other setting.

Data Size	SA3	SA2	SA1	SA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.2 DDAm — DMAC Destination Address Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + 0404_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.22 DDAm Register Contents

Bit Position	Bit Name	Function
31 to 0	DA[31:0]	Destination address Specifies the DMA transfer destination address. Those bits are updated whenever a DMA cycle is executed. If you read from those bits, the transfer destination address for the next DMA cycle is read.

CAUTIONS

1. It is forbidden to write to those bits when the channel operation is enabled (DCENm.DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the destination address is updated.
3. DMA transfer for misaligned data is not supported. The lower 4 bits of the address corresponding to each transfer data size is as follows. (× denotes an arbitrary one bit.)
   The correct operation is not guaranteed if you select other setting.

Data Size	DA3	DA2	DA1	DA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.3 DTCm — DMAC Transfer Count Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + 0408_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ARC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.23 DTCm Register Contents

Bit Position	Bit Name	Function
31 to 16	ARC[15:0]	Address reload count Specifies the number of transfers until the address reload when the reload function 2 is used, and also specifies the number of transfers when the block transfer 2 is used. If you read from those bits during DMA transfer, the address reload count for the next DMA cycle is read. When the reload function 2 or block transfer 2 is used, DTCm.ARC[15:0] is decremented by one for every DMA cycle. When the reload function 2 or block transfer 2 is not used, DTCm.ARC[15:0] is not updated. If the value is 0000H, it means that the number of transfers until the address reload when the reload function 2 is used and the number of transfers when the block transfer 2 is used are 65536.
15 to 0	TRC[15:0]	Transfer count Specifies the number of transfers. DTCm.TRC[15:0] is decremented by one whenever a DMA cycle is executed. If you read from those bits, the remaining number of transfers for the next DMA cycle is read. If the reload function is not used, after the last transfer is complete, the value at the completion (0000_H) is retained.

TRC15-0	Operation
0000_H	The number of transfers is 65536, or the transfer is complete.
0001_H	The number of transfers is 1, or remaining transfer count is 1.
:	:
$FFFF_H$	The number of transfers is 65535, or remaining transfer count is 65535.

CAUTIONS

1. It is forbidden to write to those bits when the channel operation is enabled (DCENm.DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the transfer count and the address reload count are updated.

8.9.2.4 DTCTm — DMAC Transfer Control Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + $040C_H$ + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ESE	DRS	—	—	—	—	—	CHNSEL[2:0]			CHNE[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	TCE	MLE	RLD2M[1:0]		RLD1M[1:0]		DACM[1:0]		SACM[1:0]		DS[2:0]			TRM[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.24 DTCTm Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	ESE	DMA transfer disable on transfer error setting Specifies whether to execute a DMA cycle when the DCSTm.ER bit is set due to DMA transfer error. If this bit is cleared to 0, even when the DCSTm.ER bit is set due to DMA transfer error, the following DMA cycles can be executed. If this bit is set to 1, the following DMA cycles are not executed when the DCSTm.ER bit is set due to DMA transfer error. 0: DMA cycles are executed while the DCSTm.ER bit is set. 1: DMA cycles are not executed while the DCSTm.ER bit is set.
26	DRS	DMA transfer request selection assignment Selects the type of DMA transfer requests to be accepted. 0: Software DMA transfer request 1: Hardware DMA transfer request
25 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 18	CHNSEL[2:0]	Selection of next channel in the chain Specifies the next channel in the chain. The next channel must be another channel in the same DMAC. You cannot specify a channel in the different DMAC. It is prohibited to specify the channel as the next channel in the chain. (If you do, the correct operation is not guaranteed.)
17, 16	CHNE[1:0]	Chain enable Selects the chain function. 00: Disabled 01: Chain at the last transfer A chain request is generated at the completion of the DMA cycle when the remaining transfer count is one. 10: Setting prohibited. (The operation is not guaranteed.) 11: Always chain A chain request is generated at the completion of every DMA cycle.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	TCE	Transfer completion interrupt enable If this bit is set, a transfer completion interrupt is generated at the completion of the last transfer.

Table 8.24 DTCTm Register Contents (2/3)

Bit Position	Bit Name	Function
13	MLE	Continuous transfer enable If this bit is set, the DCENm.DTE bit is not cleared at the completion of DMA transfer. Even if the DCSTm.TC bit is not cleared, DMA transfer starts when there is a DMA transfer request. 0: The DCENm.DTE bit is cleared at the completion of DMA transfer. The next DMA transfer starts only after the DCSTm.TC bit is cleared. 1: The DCENm.DTE bit is not cleared at the completion of DMA transfer. Even if the DCSTm.TC bit is not cleared, DMA transfer starts when there is a DMA transfer request.
12, 11	RLD2M[1:0]	Reload function 2 setting Specifies the reload function 2. 00: Reload function 2 is disabled. 01: Reload function 2 is enabled. The source address and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1. 10: Reload function 2 is enabled. The destination address and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1. 11: Reload function 2 is enabled. The source address, destination address, and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1.
10, 9	RLD1M[1:0]	Reload function 1 setting Specifies the reload function 1. 00: Reload function 1 is disabled. 01: Reload function 1 is enabled. The source address and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.) 10: Reload function 1 is enabled. The destination address and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.) 11: Reload function 1 is enabled. The source address, destination address, and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.)
8, 7	DACM[1:0]	Destination address count direction Specifies the count direction of the destination address.

DACM1	DACM0	Direction of Count
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited (The operation is not guaranteed.)

6, 5	SACM[1:0]	Source address count direction Specifies the count direction of the source address.

SACM1	SACM0	Direction of Count
0	0	**Increment**
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited (The operation is not guaranteed.)

Table 8.24 DTCTm Register Contents (3/3)

Bit Position	Bit Name	Function
4 to 2	DS[2:0]	Transfer data size Specifies the transfer data size.

DS2	DS1	DS0	Transfer Data Size
0	0	0	8 bits
0	0	1	16 bits
0	1	0	32 bits
0	1	1	64 bits*1
1	0	0	128 bits*1
Other than the above			Setting prohibited (The operation is not guaranteed.)

Note 1. These bits can be specified when transferring to the area other than the external memory area.

1, 0	TRM[1:0]	Transfer mode Specifies the DMA transfer mode. 00: Single transfer 01: Block transfer 1 (The number of transfers is specified by the transfer count.) 10: Block transfer 2 (The number of transfers is specified by the address reload count.) 11: Setting prohibited (The operation is not guaranteed.)

CAUTIONS

1. Except for the case to clear DTCTm.MLE bit, it is forbidden to write to those bits when the channel operation is enabled (DCENm.DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If forbidden settings are used for each bits, the correct operation is not guaranteed.

8.9.2.5 DRSAm — DMAC Reload Source Address Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + 0410_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.25 DRSAm Register Contents

Bit Position	Bit Name	Function
31 to 0	RSA[31:0]	Reload source address Specifies the source address to be reloaded to the DMA source address register when the reload function 1 or reload function 2 is used.

CAUTION

DMA transfer for misaligned data is not supported. The lower 4 bits of the address corresponding to each transfer data size is as follows. (× denotes an arbitrary one bit.)

The correct operation is not guaranteed if you set otherwise than the following table.

Data Size	RSA3	RSA2	RSA1	RSA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.6 DRDAm — DMAC Reload Destination Address Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + 0414_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.26 DRDAm Register Contents

Bit Position	Bit Name	Function
31 to 0	RDA[31:0]	Reload destination address Specifies the destination address to be reloaded to the DMA destination address register when the reload function 1 or reload function 2 is used.

CAUTION

DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (× denotes an arbitrary one bit.)

The correct operation is not guaranteed if you set otherwise than the following table.

Data Size	RDA3	RDA2	RDA1	RDA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.7 DRTCm — DMAC Reload Transfer Count Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + 0418_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RARC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTRC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.27 DRTCm Register Contents

Bit Position	Bit Name	Function
31 to 16	RARC[15:0]	Reload address reload count Specifies the value to be reloaded to the address reload count in the transfer count register at the timing of reload when the reload function 2 is used.
15 to 0	RTRC[15:0]	Reload transfer count Specifies the value to be reloaded to the transfer count in the transfer count register at the timing of reload when the reload function 1 is used.

8.9.2.8 DCENm — DMAC Channel Operation Enable Setting Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + 0420_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.28 DCENm Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DTE	Channel operation enable Specifies whether to enable or disable the transfer operation of the channel. If the DCSTm.DTE bit is 1, DMA transfer starts when there is a DMA transfer request. If the DTCTm.MLE bit is 0, this bit is automatically cleared when DMA transfer is completed. In addition, if 0 is written to the DCSTm.DTE bit during DMA transfer, the DMA transfer is suspended. If 1 is written to the DCSTm.DTE bit during suspension, the suspension is cleared and the DMA transfer resumes. 0: Channel operation is disabled/Channel suspended 1: Channel operation is enabled/Channel suspension cleared

8.9.2.9 DCSTm — DMAC Transfer Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMA_base> + 0424_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ERWR	—	—	CY	ER	—	—	TC	—	—	DR	SR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.29 DCSTm Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned.
11	ERWR	DMA Transfer Error occurring cycle This bit is updated at the same time as setting of the DMA transfer error flag (DCSTm.ER), indicating in which cycle of read or write the DMA transfer error occurred. This bit is not updated when a new DMA transfer error occurs after the DCSTm.ER bit has been set. If the DCSTm.ER bit is cleared, this bit is also cleared to 0. 0: DMA transfer error occurs in the read cycle. 1: DMA transfer error occurs in the write cycle.
10, 9	Reserved	When read, the value after reset is returned.
8	CY	DMA cycle execution state This bit shows whether a DMA cycle is ongoing in this channel. 0: DMA cycle is not ongoing. 1: DMA cycle is ongoing.
7	ER	Transfer error flag This bit is set when DMA transfer error is generated. If this bit is 1 and the DTCTm.ESE bit is set, a DMA cycle is not executed even when a DMA transfer request is generated. 0: No DMA transfer error is generated 1: DMA transfer error is generated
6, 5	Reserved	When read, the value after reset is returned.
4	TC	Transfer completion flag This bit is set at the completion of the last transfer and indicates that the DMA transfer is complete. If the DTCTm.MLE bit is 0 and this bit is 1, a DMA cycle is not executed when a DMA transfer request is generated. 0: DMA transfer is incomplete. 1: DMA transfer is complete.
3, 2	Reserved	When read, the value after reset is returned.
1	DR	Hardware DMA transfer request status This bit shows whether there is a hardware DMA transfer request (DMARQ) from the DTFR. This bit changes regardless of the value of the DCENm.DTE bit when a hardware DMA transfer request from the DTFR is generated. If the software DMA transfer request has been selected in the transfer request selection bit (DTCTm.DRS) in the DMAC transfer control register, this bit is not set even when a hardware DMA transfer request is input from the DTFR. 0: There is no hardware DMA transfer request 1: There is a hardware DMA transfer request

Table 8.29 DCSTm Register Contents (2/2)

Bit Position	Bit Name	Function
0	SR	Software DMA transfer request flag This bit shows whether there is a software DMA transfer request. This bit is automatically cleared when executing the DMA transfer. A user can set this bit by writing 1 to the DCSTSm.SRS bit in the DMAC transfer status set register (DCSTSm). In addition, a user can clear this bit by writing 1 to the DCSTCm.SRC bit in the DMAC transfer status clear register (DCSTCm), but if this is done, the ongoing DMA transfer is aborted and cannot be resumed. 0: There is no software DMA transfer request 1: There is a software DMA transfer request

8.9.2.10 DCSTSm — DMAC Transfer Status Set Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + 0428_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.30 DCSTSm Set Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SRS	Software DMA transfer request flag A user can set the software DMA transfer request flag (DCSTm.SR) by writing 1 to this bit. When read, this bit is always read as 0.

8.9.2.11 DCSTCm — DMAC Transfer Status Clear Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + $042C_H$ + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ERC	—	—	TCC	—	—	—	SRC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R	R	R	R/W

Table 8.31 DCSTCm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ERC	Transfer error flag clear The DMA transfer error flag (DCSTm.ER) can be cleared by writing 1 to this bit. When read, this bit is always read as 0.
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	TCC	Transfer completion flag clear The transfer completion flag (DCSTm.TC) can be cleared by writing 1 to this bit. When read, this bit is always read as 0.
3 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SRC	Software DMA transfer request flag clear The software DMA transfer request flag (DCSTm.SR) can be cleared by writing 1 to this bit. When read, this bit is always read as 0.

8.9.2.12 DTFRm — DTFR Setting Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + 0430_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	REQSEL[6:0]							REQEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.32 DTFRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 1	REQSEL[6:0]	Hardware DMA transfer source selection Selects one out of 128 hardware DMA transfer sources as the hardware DMA transfer request. 000_0000: Select the DMACTRG[0] input : 111_1111: Select the DMACTRG[127] input
0	REQEN	Hardware DMA transfer source selection enable This bit enables/disables the hardware DMA transfer source selection. 0: Hardware DMA transfer source selection is disabled. 1: Hardware DMA transfer source selection is enabled. If this bit is 0, even when the hardware DMA transfer source selected by the DTFRm.REQSEL[6:0] bits is activated, it is not recognized as a hardware DMA transfer request, and a hardware DMA transfer request is not generated.

8.9.2.13 DTFRRQm — DTFR Transfer Request Status Register

Access: This register is a read-only register that can be read in 32-bit units.

Address: <DMA_base> + 0434_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQ
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.33 DTFRRQm Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	DRQ	Hardware DMA transfer request status This bit indicates that a hardware DMA transfer request exists or is retained. • If the hardware DMA transfer request is an edge detection type*1 This bit indicates whether a hardware DMA transfer request generated by edge detection is retained or not. When the DMA transfer request acceptance signal from the DMAC is asserted, this bit is automatically cleared. A user can clear this bit by writing 1 to the DTFRRQCm.DRQC bit. • If the hardware DMA transfer request is a level input type*1 This bit indicates whether there is a hardware DMA transfer request input from the outside or not. Even when the DMA transfer request acceptance signal from the DMAC is asserted, this bit is not automatically cleared. In addition, this bit can not be cleared even when a user writes to the DTFRRQCm.DRQC bit. This bit changes regardless of the value of the DTFRm.REQEN bit when a hardware DMA transfer request from the outside is generated. 0: There is no hardware DMA transfer request 1: There is a hardware DMA transfer request

Note 1. Whether the hardware DMA transfer request is an edge detection type or level input type depends on the hardware DMA transfer source selected by DTFRm.REQSEL. The hardware DMA transfer request is only an edge detection type.

8.9.2.14 DTFRRQCm — DTFR Transfer Request Clear Register

Access: This register can be read or written in 32-bit units.

Address: <DMA_base> + 0438_H + 40_H × Ch. No. m (m = 0 to 15)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.34 DTFRRQCm Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DRQC	Hardware DMA transfer request clear If the hardware DMA transfer request is an edge detection type*1, a user can clear the DTFRRQm.DRQ bit by writing 1 to this bit. If the hardware DMA transfer request is a level input type*1, the DTFRRQm.DRQ bit cannot be cleared by writing to this bit. When read, this bit is always read as 0.

Note 1. Whether the hardware DMA transfer request is an edge detection type or level input type depends on the hardware DMA transfer source selected by DTFRm.REQSEL. The hardware DMA transfer request is only an edge detection type.

Section 9 Reset Controller

9.1 Overview

Several system reset functions are provided in order to initialize CPU core and peripheral functions as well as their associated registers.

A reset can be caused by the following events:

- External reset ($\overline{\text{RESET}}$)
- Power-on-clear (POCRES)
- Watchdog timer reset (WDTA0RES, WDTA1RES)
- Clock monitor reset ($\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$)
- Low-voltage indicator reset ($\overline{\text{LVIRES}}$)
- Software reset (SWRES)
- Debugger reset ($\overline{\text{DBRES}}$)
- Core voltage monitor reset ($\overline{\text{CVMRES}}$)
- Transition to DeepSTOP mode

9.1.1 Reset Sources

Reset levels and reset sources are shown below.

Various reset sources are assigned to the different levels of the reset.

Table 9.1 Reset Sources and Reset Targets

Reset Level	Reset Source	Clock Generator (except PLL)/ Real-Time Clock/CVM/ LVI	AWO Modules*1	ISO Modules *3
1	Power-on-clear (POCRES) Debugger reset (DBRES)	Reset*2	Reset	Reset
2	External reset (RESET) Watchdog timer reset (WDTA0RES, WDTA1RES) Clock monitor reset (CLMA0RES, CLMA1RES, CLMA2RES) Core voltage monitor reset (CVMRES) Low voltage indicator reset (LVIRES) Software reset (SWRES)	Not reset target*4	Reset	Reset
3	Reset by DeepSTOP mode	Not reset target	Not reset target	Reset

Note 1. Clock generator, real-time clock, CVM, and LVI are excluded.

Note 2. For a debugger reset, the CVM related registers (CVMF, CVMDE, and CVMDIAG) are not reset.

Note 3. PLL is included.

Note 4. In clock monitor reset, oscillator-related registers for clock monitoring are initialized.

Reset level 1: Initializes the entire microcontroller.

Reset level 2: For a quick return to normal operating mode by eliminating the oscillator stabilization time*1, initializes the entire microcontroller except for the clock generator and the real-time clock.

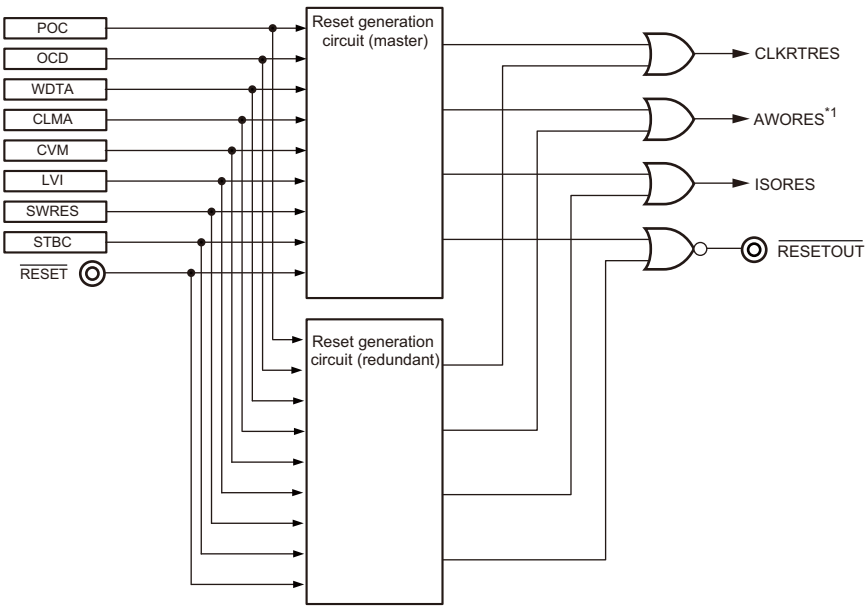
Reset level 3: At the transition to DeepSTOP mode, initializes all the isolated areas (ISO).

Note 1. MOSCSTPMSK=1

9.1.2 Reset Controller Redundancy

The reset controller of the microcontroller have redundant configuration, and includes duplicated reset generation circuits. Such configuration enables initialization of the reset targeted area without failure even if one of the two reset generation circuits fails.

The configuration of the reset controller is shown in the figure below.



Note 1. Real-time clock is excluded.

Figure 9.1 Reset Controller Redundancy

At the generation of a reset, the same reset source signal is input to two reset generation circuits.

According to the reset source, the two reset generation circuits output the Always-ON area reset signal (AWORES), Isolated area reset signal (ISORES), clock generator/real-time clock reset signal (CLKRTRES), and $\overline{\text{RESETOUT}}$ signal.

The AWORES, ISORES, CLKRTRES, and $\overline{\text{RESETOUT}}$ signals are generated by executing the logical OR of the signals output from the two reset generation circuits. Thus, a reset signal is generated normally even if one of the two reset generation circuits fails.

Whether a reset generation circuit operates normally can be checked by reading and comparing the reset factor registers of the respective reset generation circuits.

9.1.3 Reset Output ($\overline{\text{RESETOUT}}$)

When a reset source of reset level 1 or 2 is generated, a reset output signal ($\overline{\text{RESETOUT}}$) is output to the outside. Reset output is used to reset external devices at the same time as a reset is generated inside the microcontroller.

For details, refer to **Section 2.11.1.1, P0_0: $\overline{\text{RESETOUT}}$**.

9.1.4 Reset Flag

To identify a reset source, two registers with a flag for each reset source are provided. The main elements of the reset controller are shown in **Figure 9.2, Block Diagram of the Reset Controller**.

9.1.5 Clock Supply

The clock supply to the reset controller is shown in the following table.

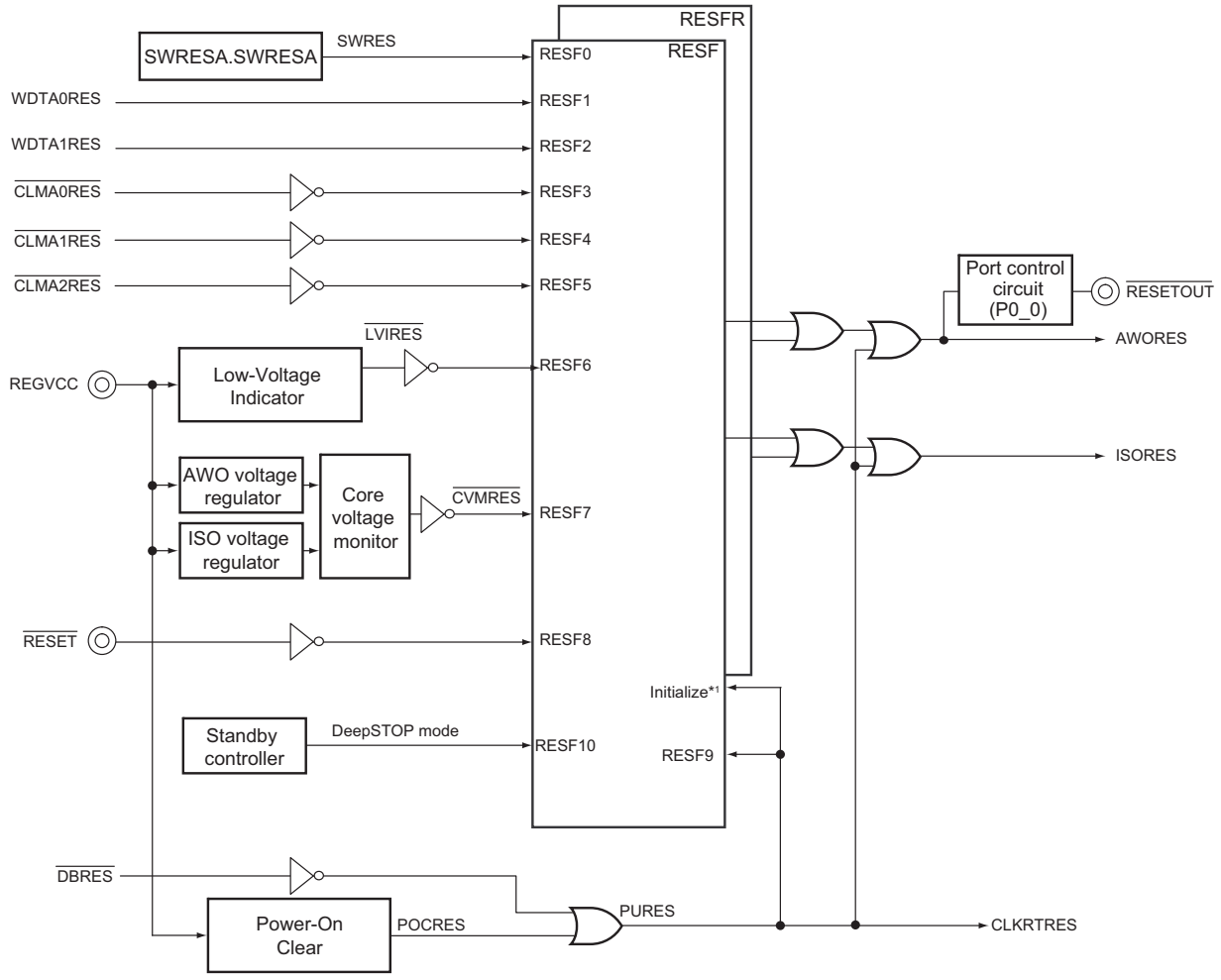
Table 9.2 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
Reset	Register access clock	EMCLK

9.2 Configuration

9.2.1 Block Diagram

A block diagram of the reset controller is shown below.



Note 1. Registers RESF and RESFR are initialized by PURES.
After an initialization, bits RESF9 and RESFR9 are set to 1.

Figure 9.2 Block Diagram of the Reset Controller

(1) Reset Signals

The reset controller manages the generation of three reset signals upon occurrence of reset signals from various reset sources:

- AWO area reset (AWORES)
  AWORES is generated by all reset sources except the transition to DeepSTOP mode.
  AWORES resets all modules in the AWO area except clock generation circuit, real-time clock, core voltage monitor, and low-voltage detection circuit.
- ISO area reset (ISORES)
  ISORES is generated by all reset sources.
  ISORES resets all modules (including PLL) in the ISO area.
- CLKRTRES
  CLKRTRES is generated by the power-on clear or debugger reset sources.
  CLKRTRES resets the clock generation circuit (excluding PLL) and real-time clock.

The power-up reset (PURES) is generated by the power-on clear and debugger reset sources.

Following the generation of an AWORES reset, with the exception of the PLL, all clock-generation circuits that were operating at the time (LS IntOSC, HS IntOSC, MainOSC, SubOSC) continue to operate. However, when MOSCSTPMSK is 0, MainOSC stops during any reset and resumes operation after release from the reset. On the generation of a $\overline{\text{CLMA0RES}}$ reset, the HS IntOSC that was the target for CLMA0 monitoring is reset. On the generation of a $\overline{\text{CLMA1RES}}$ reset, the MainOSC that was the target for CLMA1 monitoring is reset.

The PURES initializes all of the clock generation circuits. It is necessary to restart the clock generation circuit after recovery from the PURES.

The CPU reset is the ISO area reset (ISORES) to the CPU sub system.

(2) Reset Flags

The reset factor register (RESF) and the redundant reset factor register (RESFR) hold a flag for each reset source, and this flag is set when the corresponding reset is asserted.

All reset flags except RESF9 and RESFR9 are initialized by a power-up reset (PURES). (Bits RESF9 and RESFR9 are set to 1 after initialization.) In addition, all the bits can be cleared by software.

For details, see **Section 9.1.4, Reset Flag**.

(3) On-Chip Module Resets

(a) Watchdog Timer Reset

The watchdog timers 0, 1 can generate two types of resets: WDTA0RES and WDTA1RES.
For details, see **Section 9.4.6, Watchdog Timer (WDTA) Reset**.

(b) Clock Monitor Resets

The clock monitors can generate three resets: $\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, and $\overline{\text{CLMA2RES}}$.
For details, see **Section 9.4.8, Clock Monitor (CLMA) Reset**.

(c) Debugger Reset

A reset is generated by a command from a debugger. This leads to a generation of power-up reset PURES. For details, see **Section 9.4.9, Debugger Reset**.

(4) Software Controlled Resets (SWRES)

A software reset SWRES can be generated by use of the software reset register SWRESA.

For details, see **Section 9.4.7, Software Reset**.

(5) Reset Output Signal

During reset and after release from the reset, port P0_0 outputs low level as $\overline{\text{RESETOUT}}$ function.

For details, see **Section 2.11.1.1, P0_0: $\overline{\text{RESETOUT}}$**.

(6) Power Supply Monitoring

The following power supply detection circuits observe the level of the external power supply REGVCC and core voltage.

(a) Low-Voltage Indicator

The low-voltage indicator (LVI) generates the $\overline{\text{LVIRES}}$ reset if the voltage level of REGVCC drops below a certain level. The level can be adjusted and the $\overline{\text{LVIRES}}$ can be masked.
For details, see **Section 9.4.3, Low-Voltage Indicator (LVI) Reset**.

(b) Power-On-Clear

The power-on-clear circuit (POC) continuously compares the power supply voltage REGVCC with an internal reference voltage. Thus, a reset is generated when the power supply voltage goes below a certain level.
For details, see **Section 9.4.2, Power-On-Clear (POC) Reset**.

(c) Core Voltage Monitor

A reset can be generated when the core voltage monitor (CVM) detects over- or undervoltage in core voltage. (Output/not output can be set by option byte.)
For details, see **Section 9.4.4, Core Voltage Monitor (CVM) Reset**.

(7) Masking of Reset Sources in Debugging Mode

The following reset sources can be masked during debugging:

Table 9.3 Reset Sources to be Masked during Debugging

Reset Source	Maskable/Non-maskable
Power-on clear (POCRES)	—
Debugger reset ($\overline{\text{DBRES}}$)	—
External reset ($\overline{\text{RESET}}$)	√
Low-voltage indicator reset ($\overline{\text{LVIRES}}$)	√
Clock monitor reset ($\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$)	√
Watchdog timer reset (WDTA0RES, WDTA1RES)	√
Core voltage monitor reset ($\overline{\text{CVMRES}}$)	√
Software reset (SWRES)	√
Reset by DeepSTOP mode	—

9.3 Registers

This section contains a description of all registers of the reset controller.

9.3.1 Reset Controller Registers Overview

The reset controller is controlled and operated by the following registers:

Table 9.4 Reset Controller Registers Overview

Register Name	Symbol	Address
Reset flag registers		
Reset factor register	RESF	FFF8 0760$_H$
Reset factor clear register	RESFC	FFF8 0768$_H$
Redundant reset factor register	RESFR	FFF8 0860$_H$
Redundant reset factor clear register	RESFCR	FFF8 0868$_H$
Software reset control register		
Software reset register	SWRESA	FFF8 0A04$_H$

NOTES

1. For the LVI related, RAM store related, and CVM related registers, see **Section 10, Supply Voltage Monitor**.
2. As for the protection registers, see **Section 5, Write-Protected Registers**.

9.3.2 Details of Reset Flag Registers

9.3.2.1 RESF — Reset Factor Register

This register contains information about which type of resets occurred after the last power-on clear reset. This register is initialized by a power-up reset PURES.

Each reset condition sets the corresponding flag in the register.

For example, if a clock monitor reset $\overline{\text{CLMA0RES}}$ occurs after a watchdog timer reset WDTA0RES, RESF reads 0000 000A$_H$.

Access: This register is a read-only register that can be read 32-bit units.

Address: FFF8 0760$_H$

Value after reset: 0000 0200$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RESF10	RESF9	RESF8	RESF7	RESF6	RESF5	RESF4	RESF3	RESF2	RESF1	RESF0
Value after reset	0	0	0	0	0	0	1	1/0*1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. For details, see **Figure 9.4, When $\overline{\text{RESET}}$ is Released before Execution of Flash Sequence**.

Table 9.5 RESF Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned.
10	RESF10	Reset flag by DeepSTOP mode 0: No reset occurred 1: Reset has occurred
9	RESF9	Power-up reset flag 0: No reset occurred 1: Reset has occurred
8	RESF8	External reset flag 0: No reset occurred 1: Reset has occurred
7	RESF7	CVM reset flag 0: No reset occurred 1: Reset has occurred
6	RESF6	LVI reset flag 0: No reset occurred 1: Reset has occurred
5	RESF5	CLMA2 reset flag 0: No reset occurred 1: Reset has occurred
4	RESF4	CLMA1 reset flag 0: No reset occurred 1: Reset has occurred
3	RESF3	CLMA0 reset flag 0: No reset occurred 1: Reset has occurred

Table 9.5 RESF Register Contents (2/2)

Bit Position	Bit Name	Function
2	RESF2	WDTA1 reset flag 0: No reset occurred 1: Reset has occurred
1	RESF1	WDTA0 reset flag 0: No reset occurred 1: Reset has occurred
0	RESF0	Software reset flag 0: No reset occurred 1: Reset has occurred

9.3.2.2 RESFC — Reset Factor Clear Register

This register clears the reset flags of the RESF register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFF8 0768$_H$

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RESFC10	RESFC9	RESFC8	RESFC7	RESFC6	RESFC5	RESFC4	RESFC3	RESFC2	RESFC1	RESFC0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	W	W	W	W	W	W	W	W	W	W	W

Table 9.6 RESFC Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When writing, write “0”.
10	RESFC10	Reset flag clear by DeepSTOP mode 0: Do not clear flag 1: Clear flag
9	RESFC9	Power-up reset flag clear 0: Do not clear flag 1: Clear flag
8	RESFC8	External reset flag clear 0: Do not clear flag 1: Clear flag
7	RESFC7	CVM reset flag clear 0: Do not clear flag 1: Clear flag
6	RESFC6	LVI reset flag clear 0: Do not clear flag 1: Clear flag
5	RESFC5	CLMA2 reset flag clear 0: Do not clear flag 1: Clear flag
4	RESFC4	CLMA1 reset flag clear 0: Do not clear flag 1: Clear flag
3	RESFC3	CLMA0 reset flag clear 0: Do not clear flag 1: Clear flag
2	RESFC2	WDTA1 reset flag clear 0: Do not clear flag 1: Clear flag
1	RESFC1	WDTA0 reset flag clear 0: Do not clear flag 1: Clear flag
0	RESFC0	Software reset flag clear 0: Do not clear flag 1: Clear flag

9.3.2.3 RESFR — Redundant Reset Factor Register

This register is a duplication of the reset factor register. This register is initialized by a power-up reset PURES.

In accordance with the setting conditions for each bit in the reset factor register, the same bits are set in this register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 0860$_H$

Value after reset: 0000 0200$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RESFR10	RESFR9	RESFR8	RESFR7	RESFR6	RESFR5	RESFR4	RESFR3	RESFR2	RESFR1	RESFR0
Value after reset	0	0	0	0	0	0	1	1/0*1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. For details, see **Figure 9.4, When $\overline{\text{RESET}}$ is Released before Execution of Flash Sequence**.

Table 9.7 RESFR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned.
10	RESFR10	Reset flag by DeepSTOP mode 0: No reset occurred 1: Reset has occurred
9	RESFR9	Power-up reset flag 0: No reset occurred 1: Reset has occurred
8	RESFR8	External reset flag 0: No reset occurred 1: Reset has occurred
7	RESFR7	CVM reset flag 0: No reset occurred 1: Reset has occurred
6	RESFR6	LVI reset flag 0: No reset occurred 1: Reset has occurred
5	RESFR5	CLMA2 reset flag 0: No reset occurred 1: Reset has occurred
4	RESFR4	CLMA1 reset flag 0: No reset occurred 1: Reset has occurred
3	RESFR3	CLMA0 reset flag 0: No reset occurred 1: Reset has occurred
2	RESFR2	WDTA1 reset flag 0: No reset occurred 1: Reset has occurred
1	RESFR1	WDTA0 reset flag 0: No reset occurred 1: Reset has occurred

Table 9.7 RESFR Register Contents (2/2)

Bit Position	Bit Name	Function
0	RESFR0	Software reset flag 0: No reset occurred 1: Reset has occurred

9.3.2.4 RESFCR — Redundant Reset Factor Clear Register

This register clears the reset flags of the RESFR register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFF8 0868$_H$

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	RESFCR10	RESFCR9	RESFCR8	RESFCR7	RESFCR6	RESFCR5	RESFCR4	RESFCR3	RESFCR2	RESFCR1	RESFCR0
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	W	W	W	W	W	W	W	W	W	W	W

Table 9.8 RESFCR Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When writing, write “0”.
10	RESFCR10	Reset flag clear by DeepSTOP mode 0: Do not clear flag 1: Clear flag
9	RESFCR9	Power-up reset flag clear 0: Do not clear flag 1: Clear flag
8	RESFCR8	External reset flag clear 0: Do not clear flag 1: Clear flag
7	RESFCR7	CVM reset flag clear 0: Do not clear flag 1: Clear flag
6	RESFCR6	LVI reset flag clear 0: Do not clear flag 1: Clear flag
5	RESFCR5	CLMA2 reset flag clear 0: Do not clear flag 1: Clear flag
4	RESFCR4	CLMA1 reset flag clear 0: Do not clear flag 1: Clear flag
3	RESFCR3	CLMA0 reset flag clear 0: Do not clear flag 1: Clear flag
2	RESFCR2	WDTA1 reset flag clear 0: Do not clear flag 1: Clear flag
1	RESFCR1	WDTA0 reset flag clear 0: Do not clear flag 1: Clear flag
0	RESFCR0	Software reset flag clear 0: Do not clear flag 1: Clear flag

9.3.3 Details of Software Reset Control Registers

9.3.3.1 SWRESA — Software Reset Register

This register is used to generate a software reset SWRES. The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFF8 0A04$_H$

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SWRESA
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9.9 SWRESA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write “0”.
0	SWRESA	Software reset trigger 0: No Software reset trigger is generated. 1: Software reset trigger is generated.

9.4 Functional Description

9.4.1 Reset Flags

The reset factor register (RESF) and the redundant reset factor register (RESFR) provide reset flags for each reset source.

If a reset has occurred, the corresponding flag is set. According to this, the source of the reset is evaluated.

RESF and RESFR are initialized by a power-up reset PURES (POCRES or $\overline{\text{DBRES}}$) (though bits RESF9 and RESFR9 are set to 1 after initialization). In addition, flags in RESF and RESFR can be cleared by the reset factor clear register (RESFC) and the redundant reset factor clear register (RESFRC).

Each reset source can set the corresponding flag independently from other reset sources.

9.4.2 Power-On-Clear (POC) Reset

The power-on-clear circuit (POC) constantly compares the power supply voltage REGVCC with the internal reference voltage V_{POC}. It ensures that the microcontroller only operates as long as the power supply exceeds a certain level.

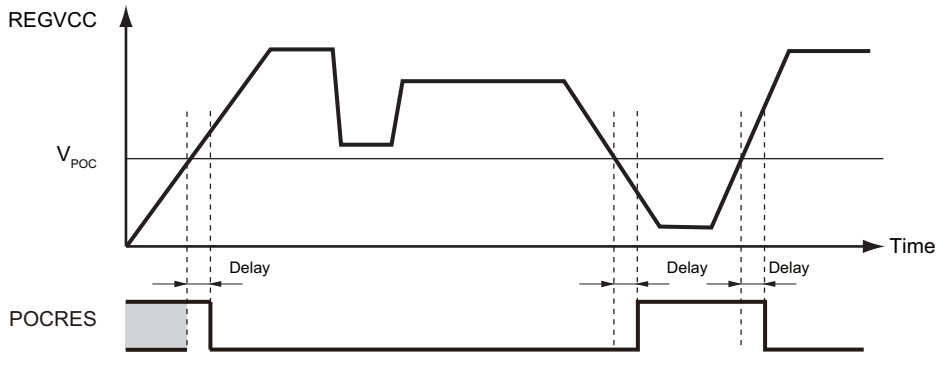
If REGVCC falls below the internal reference voltage (REGVCC < V_{POC}), the internal reset signal POCRES and a power-up reset PURES are asserted.

For details on the specifications of the internal voltage reference level V_{POC}, see the Data Sheet.

The reset factor register (RESF) and the redundant reset factor register (RESFR) are cleared by the power-on clear reset. RESF9 and RESFR9 are set to 1 after initialization.

The power-on-clear function holds the microcontroller in reset state as long as the power supply voltage does not exceed the threshold level V_{POC}.

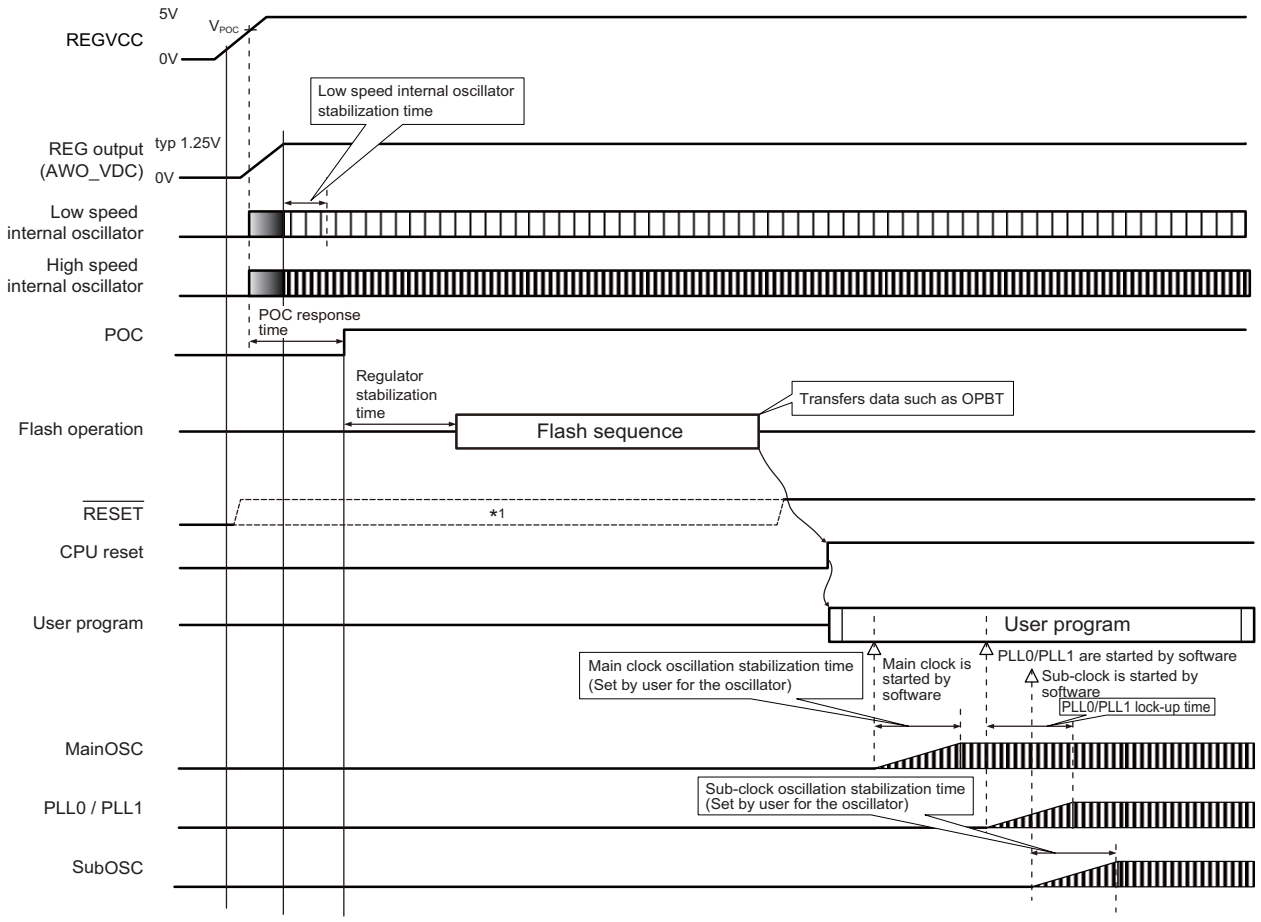
The following figure illustrates the timing of a POCRES.



Note: Delay:
When the REGVCC exceeds the V_{POC}, the POCRES is released after certain delay. When the REGVCC falls below the V_{POC}, the POCRES is asserted after certain delay.
See the Data Sheet for the delay time.

Figure 9.3 POC Reset Timing

(1) Overview of CPU System Startup after Power-On-Clear



Note 1. After a power-on clear reset, the on-chip pull-down resistors (min. 20 kΩ, typ. 40 kΩ, max. 100 kΩ) are enabled until the flash sequence is completed. The timing of releasing the $\overline{\text{RESET}}$ pin depends on the external circuit of the microcontroller.
If the $\overline{\text{RESET}}$ pin is low level, the initial values of both RESF.RESF8 and RESFR.RESFR8 are set to 1. If the $\overline{\text{RESET}}$ pin is high level, the initial values of both RESF.RESF8 and RESFR.RESFR8 are set to 0.

Figure 9.4 When $\overline{\text{RESET}}$ is Released before the Flash Sequence is Completed

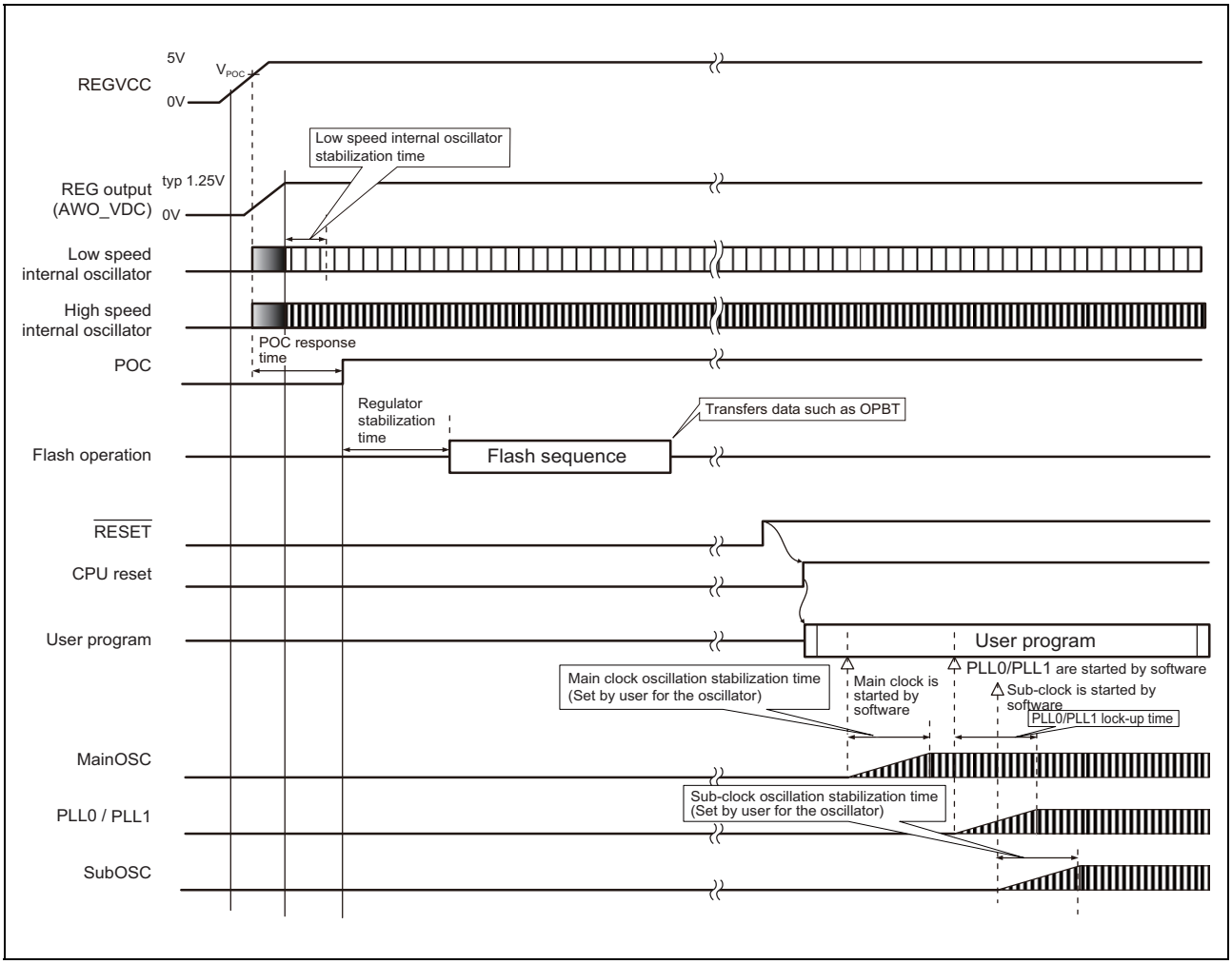


Figure 9.5 When $\overline{\text{RESET}}$ is Released after the Flash Sequence is Completed

9.4.3 Low-Voltage Indicator (LVI) Reset

The low-voltage indicator (LVI) permanently compares the power supply voltage REGVCC with the LVI internal reference voltage V_{LVI}.

When setting the LVI detection voltage and releasing the LVIRESMK, if REGVCC falls below the internal reference voltage (REGVCC < V_{LVI}), the internal reset signal $\overline{\text{LVIRES}}$ is generated.

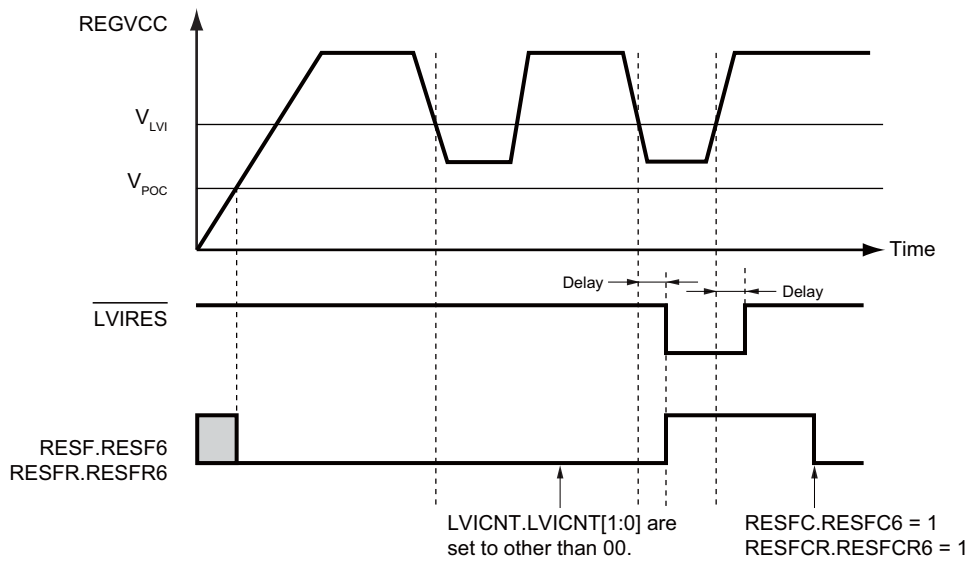
Additionally, the $\overline{\text{LVIRES}}$ flags (bits RESF.RESF6 and RESFR.RESFR6) are set.

After that, even if REGVCC exceeds V_{LVI}, bits RESF.RESF6 and RESFR.RESFR6 are not cleared automatically. They are cleared as described below.

- Setting the RESFC.RESFC6 bit to 1 clears the RESF.RESF6 bit.
  Setting the RESFCR.RESFCR6 bit to 1 clears the RESFR.RESFR6 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

For details on the LVI functions, see **Section 10, Supply Voltage Monitor**.

The following figure illustrates the timing of a $\overline{\text{LVIRES}}$ and bits RESF.RESF6 and RESFR.RESFR6.



Note: Delay:
When the REGVCC falls below the V_{LVI}, the $\overline{\text{LVIRES}}$ is asserted and LVI reset flags (RESF.RESF6 and RESFR.RESFR6) are set after certain delay. After that, when the REGVCC exceeds the V_{LVI}, the $\overline{\text{LVIRES}}$ is released after certain delay.
See the Data Sheet for the delay time.

Figure 9.6 LVI Reset Timing

9.4.4 Core Voltage Monitor (CVM) Reset

Core voltage monitor is used to monitor the core voltage inside the microcontroller.

The reset $\overline{\text{CVMRES}}$ is generated if the core voltage is not in the specified voltage range while CVM is enabled. Moreover, the $\overline{\text{CVMRES}}$ flags (RESF.RESF7 and RESFR.RESFR7) are set.

After that, the RESF.RESF7 and RESFR.RESFR7 bits are not automatically cleared even if the core voltage returns to the specified voltage range. The RESF.RESF7 and RESFR.RESFR7 bits are cleared as described below.

- Setting the RESFC.RESFC7 bit to 1 clears the RESF.RESF7 bit.
  Setting the RESFCR.RESFCR7 bit to 1 clears the RESFR.RESFR7 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

If the CVM detects an abnormal high voltage, the power supply to the ISO area is switched off.

Once the $\overline{\text{CVMRES}}$ by the high voltage detection is generated, the microcontroller stays in the reset state.

To cancel this state, it is mandatory to use the external reset ($\overline{\text{RESET}}$) input.

Release the external reset ($\overline{\text{RESET}}$) after the voltage level becomes lower than the high detection voltage.

For details on the CVM function, see **Section 10, Supply Voltage Monitor**.

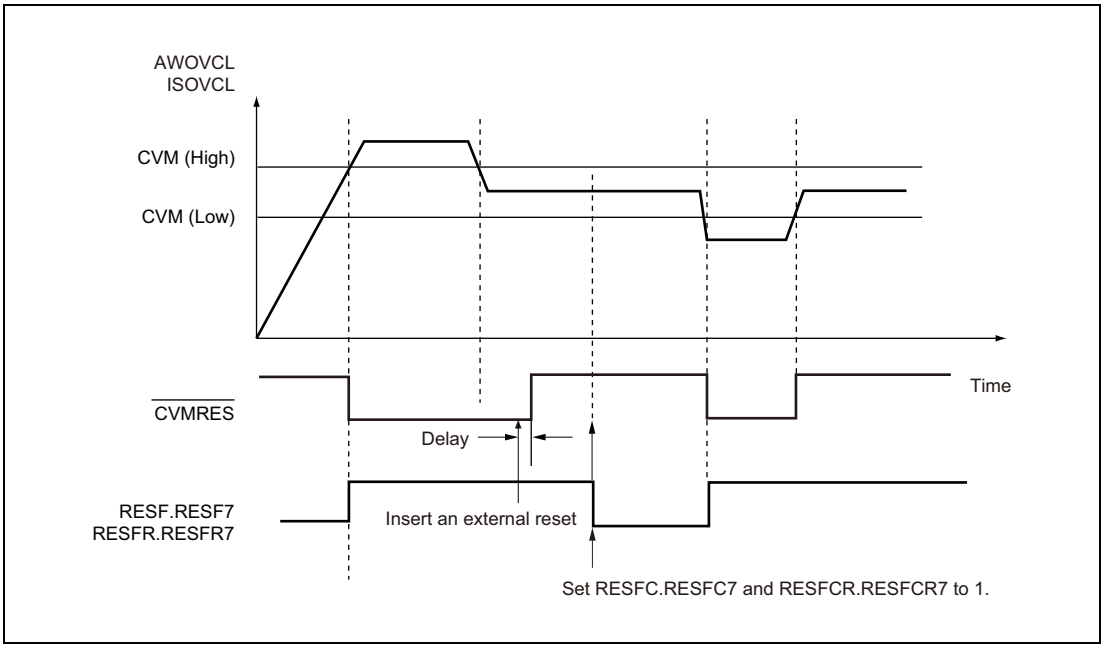


Figure 9.7 CVM Reset Timing

9.4.5 External Reset ($\overline{\text{RESET}}$)

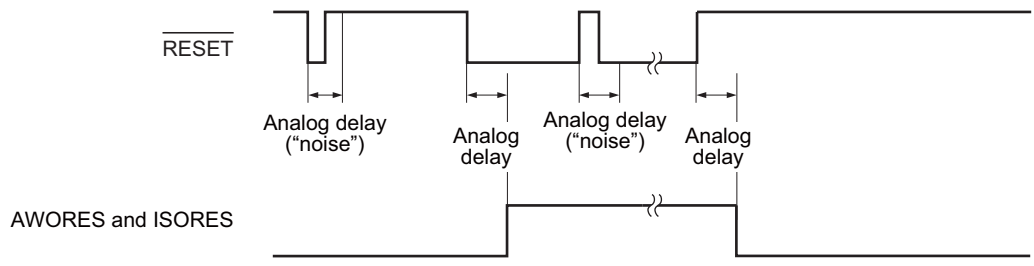
When a low-level input is applied to the $\overline{\text{RESET}}$ pin, a reset is asserted and the RESF.RESF8 and RESFR.RESFR8 bits are set.

After that, bits RESF.RESF8 and RESFR.RESFR8 are not cleared automatically, even if the low level input to the $\overline{\text{RESET}}$ pin is released. Bits RESF.RESF8 and RESFR.RESFR8 are cleared as described below.

- Setting the RESFC.RESFC8 bit to 1 clears the RESF.RESF8 bit.
  Setting the RESFCR.RESFCR8 bit to 1 clears the RESFR.RESFR8 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

The $\overline{\text{RESET}}$ pin includes an analog noise filter to prevent erroneous resets due to noise.

The following figure shows the timing when AWORES and ISORES are generated by an external reset. This figure also shows the effect of the noise filter.



Note: The analog delay is caused by the analog filter. The filter regards pulses up to a certain width as noise and suppresses them.
For the minimum $\overline{\text{RESET}}$ pulse width, see the Data Sheet.

Figure 9.8 External Reset ($\overline{\text{RESET}}$)

9.4.6 Watchdog Timer (WDTA) Reset

The watchdog timers can be configured to generate a reset if the overflow time is exceeded. After a watchdog timer reset is asserted, the corresponding watchdog timer reset flags (the RESF.RESF1 and RESFR.RESFR1 bits for WDTA0RES, and the RESF.RESF2 and RESFR.RESFR2 bits for WDTA1RES) are set.

After that, bits RESF.RESF1 and RESFR.RESFR1 (or bits RESF.RESF2 and RESFR.RESFR2) are not cleared automatically, even if WDTA0RES (or WDTA1RES) is released.

Bits RESF.RESF1 and RESFR.RESFR1, and bits RESF.RESF2 and RESFR.RESFR2 are cleared as described below.

- WDTA0RES:
  Setting the RESFC.RESFC1 bit to 1 clears the RESF.RESF1 bit.
  Setting the RESFCR.RESFCR1 bit to 1 clears the RESFR.RESFR1 bit.
- WDTA1RES:
  Setting the RESFC.RESFC2 bit to 1 clears the RESF.RESF2 bit.
  Setting the RESFCR.RESFCR2 bit to 1 clears the RESFR.RESFR2 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9.4.7 Software Reset

The software reset SWRES can be asserted by setting SWRESA.SWRESA to 1.

SWRES sets the reset flag RESF.RESF0 and the RESFR.RESFR0 bit.

RESF.RESF0 and RESFR.RESFR0 are not cleared automatically. RESF.RESF0 and RESFR.RESFR0 are cleared as described below.

- Setting the RESFC.RESFC0 bit to 1 clears the RESF.RESF0 bit.
  Setting the RESFCR.RESFCR0 bit to 1 clears the RESFR.RESFR0 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9.4.8 Clock Monitor (CLMA) Reset

The clock monitors can generate the following resets:

- $\overline{\text{CLMA0RES}}$, if a frequency abnormality in HS IntOSC is detected
- $\overline{\text{CLMA1RES}}$, if a frequency abnormality in MainOSC is detected
- $\overline{\text{CLMA2RES}}$, if a frequency abnormality in PLL is detected

When the Clock Monitor detects frequency abnormality of the respective clocks, resets $\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, and $\overline{\text{CLMA2RES}}$ are generated.

In addition, flags $\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, and $\overline{\text{CLMA2RES}}$ (RESF.RESF3, RESFR.RESFR3, RESF.RESF4, RESFR.RESFR4, RESF.RESF5, and RESFR.RESFR5) are set.

These flags are not cleared automatically. They are cleared as described below.

- $\overline{\text{CLMA0RES}}$:
  Setting the RESFC.RESFC3 bit to 1 clears the RESF.RESF3 bit.
  Setting the RESFCR.RESFCR3 bit to 1 clears the RESFR.RESFR3 bit.
- $\overline{\text{CLMA1RES}}$:
  Setting the RESFC.RESFC4 bit to 1 clears the RESF.RESF4 bit.
  Setting the RESFCR.RESFCR4 bit to 1 clears the RESFR.RESFR4 bit.
- $\overline{\text{CLMA2RES}}$:
  Setting the RESFC.RESFC5 bit to 1 clears the RESF.RESF5 bit.
  Setting the RESFCR.RESFCR5 bit to 1 clears the RESFR.RESFR5 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9.4.9 Debugger Reset

Debugger reset ($\overline{\text{DBRES}}$) is generated via a debugger command. $\overline{\text{DBRES}}$ activates PURES, and therefore operates in the same way as the power-on-clear reset POCRES:

- The clock generators are reset and stops operating. The clock generators should be restarted after release from the reset state.
- The reset factor register RESF and the redundant reset factor register RESFR are cleared (Bits RESF9 and RESFR9 are set to 1 after initialization.).

Section 10 Supply Voltage Monitor

This section explains in general about the supply voltage monitor.

The first part in this section describes the supply voltage monitor function, and the ensuing sections describe the registers.

This power supply monitor is for detecting and control of a power supply failure as early as possible when it occurs. However the voltage monitor does not detect all of the possible failures.

Therefore, power supply monitoring with an external device is needed for the following terminals in case that the customer's system which requires appropriate failure detection and control for safety.

- REGVCC
- EVCC
- BVCC
- A0VREF
- A1VREF
- AWOVCL
- ISOVCL

The required power supply specification for power supply monitoring with external device is shown in the data sheet.

10.1 Overview

10.1.1 Functional Overview

The supply voltage monitor continuously monitors multiple external and internal supply voltages in order to ensure that the device operates with a supply voltage within the specified range. If the voltage drops below the reference voltage or comparison voltage, an interrupt request signal or internal reset signal is generated. The following table lists the supply voltage monitor functions.

Table 10.1 Supply Voltage Monitor Functions

Function Name	Monitor Voltage	Signal Generated when Voltage Drops below Certain Level
Power-On Clear (POC)	REGVCC	Internal reset signal
Low-Voltage Indicator Circuit (LVI)	REGVCC	Internal reset signal, interrupt request signal
Core Voltage Monitor (CVM)	AWO, ISO area voltage	Internal reset signal
RAM Retention Voltage Indicator (VLVI)	REGVCC	—

Note: The RAM Retention Voltage Indicator sets the very-low voltage detection flag (VLVF) when the voltage drops below the RAM retention voltage.

10.1.2 Power-On Clear (POC)

The POC continuously monitors the external power supply voltage REGVCC. This ensures that the microcontroller only operates at or above power-on-clear detection voltage (V_{POC}).

If REGVCC falls below the POC detection voltage (REGVCC < V_{POC}), the internal reset signal (POCRES) is generated.

For details, see **Section 9.4.2, Power-On-Clear (POC) Reset**.

10.1.3 Low Voltage Indicator Circuit (LVI)

The LVI continuously compares the external power supply voltage REGVCC with the LVI reference voltage V_{LVI}.

If REGVCC falls below the reference voltage (REGVCC < V_{LVI}), an internal reset signal or interrupt request signal is generated.

10.1.3.1 LVI Reference Voltage

The LVI reference voltage V_{LVI} can be selected from three different levels by LVICNT.LVICNT[1:0].

If LVICNT.LVICNT[1:0] is set to 00_B, the LVI is disabled.

For the specifications of the reference voltage level V_{LVI}, see **Section 10.2.2.1, LVICNT — LVI Control Register**.

10.1.3.2 LVI Reset (LVIRES)

When the LVI detection voltage is set and LVIRESMK is cleared, if REGVCC falls below the reference voltage (REGVCC < V_{LVI}), the internal reset signal $\overline{\text{LVIRES}}$ is generated.

For the specifications of $\overline{\text{LVIRES}}$ generation, see **Section 9.4.3, Low-Voltage Indicator (LVI) Reset**.

10.1.3.3 LVI Interrupt (INTLVIL / INTLVIH)

After the LVI detection voltage is set to the LVICNT.LVICNT[1:0] and the LVICNT.LVIRESMK is set to 1, if REGVCC falls below the reference voltage (REGVCC (MIN) < V_{LVI}), the LVI interrupt INTLVIL is generated.

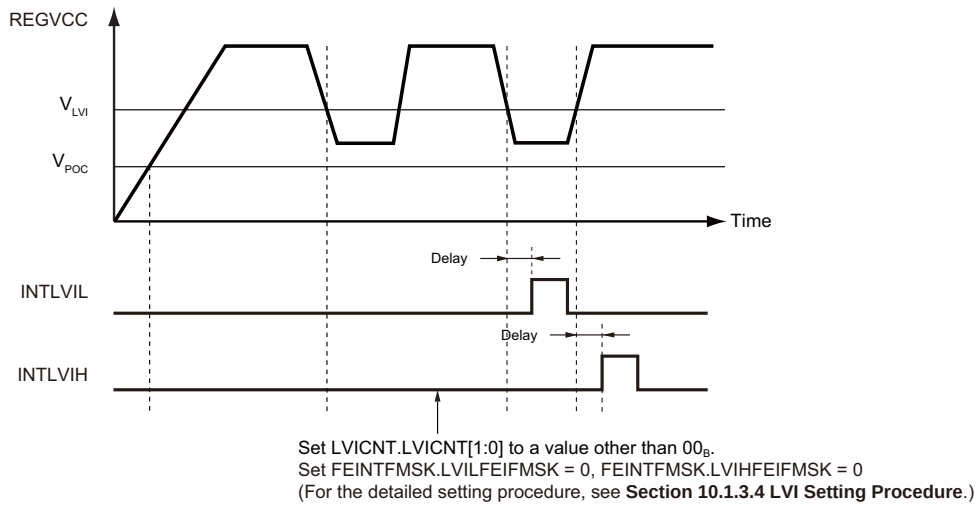
To use the LVI as an interrupt source, the INTLVIL interrupt must be unmasked.

INTLVIL interrupt can be used as wake-up source from all standby modes. For details, see **Section 12, Stand-By Controller (STBC)**.

After the LVI detection voltage is set to the LVICNT.LVICNT[1:0] and the LVICNT.LVIRESMK is set to 1, if REGVCC exceeds the reference voltage (REGVCC (MIN) > V_{LVI}), LVI interrupt INTLVIH is generated.

When LVI is used as an interrupt source, INTLVIH interrupt must be unmasked.

The following figure illustrates the timing of INTLVIL / INTLVIH.



Note: Delay:
When the REGVCC falls below the V_{LVI}, the INTLVIL is generated after certain delay. After that, when the REGVCC exceeds the V_{LVI}, the INTLVIH is generated after certain delay.
See the Data Sheet for delay time.

Figure 10.1 INTLVIL / INTLVIH Generation Timing

10.1.3.4 LVI Setting Procedure

The setting procedures for LVI are shown below.

(1) Using LVI as the reset source

(1) Mask LVI reset. (LVICNT.LVIRESMK = 1)*1

(2) Mask LVI interrupt. (FEINTFMSK.LVILFEIFMSK = 1, FEINTFMSK.LVIHFEIFMSK = 1)

(3) Set detection voltage and enable operation. (Set LVICNT.LVICNT[1:0])*1

(4) Insert ample wait time by software (see Data Sheet).

(5) Unmask LVI reset. (LVICNT.LVIRESMK = 0)*1

(2) Using LVI as the interrupt source (FEINT)

(1) Mask LVI reset. (LVICNT.LVIRESMK = 1)*1

(2) Mask LVI interrupt. (FEINTFMSK.LVILFEIFMSK = 1, FEINTFMSK.LVIHFEIFMSK = 1)

(3) Set detection voltage and enable operation. (Set LVICNT.LVICNT[1:0])*1

(4) Insert ample wait time by software (see Data Sheet).

(5) Unmask LVI interrupt. (FEINTFMSK.LVILFEIFMSK = 0, FEINTFMSK.LVIHFEIFMSK = 0)

Note 1. Follow the register protection sequence to set LVICNT register because it is a write-protected register. For details on the write-protected registers, see **Section 5, Write-Protected Registers**.

CAUTION

If REGVCC is not stable around the LVI detection level (V_{LVI}), correct judgment of whether INTLVIH or INTLVIL interrupt processing should proceed may not be possible.

For example, if multiple interrupts consisting of both INTLVIH and INTLVIL occur during INTLVIL interrupt processing due to REGVCC being unstable, the software cannot detect which type of interrupt was generated last.

Consequently, if the last interrupt generated was an INTLVIL interrupt, regardless of REGVCC (min.) being greater than V_{LVI}, the software erroneously judges that REGVCC (min.) < V_{LVI}.

Accordingly, take measures such as programming the software so that LVI detection interrupt processing is completed before the next LVI interrupt. Also, consider control of REGVCC.

10.1.3.5 Clock Supply to the LVI

The clock supply to the LVI is shown in the following table.

Table 10.2 Clock Supply to the LVI

Unit Name	Unit Clock Name	Supply Clock Name
LVI	Register access clock	EMCLK

10.1.4 Core Voltage Monitor (CVM)

The core voltage monitor (CVM) monitors the AWO area and ISO area voltages (referred to as “core voltage” below) in the microcontroller.

If the regulator output voltage is outside the specified range, the internal reset signal ($\overline{\text{CVMRES}}$) is generated.

If the CVM detects an abnormal high voltage, the power supply to the ISO area is switched off in addition to a reset being generated.

When operation shifts to diagnostic mode (DIAG mode), the CVM enters the abnormal core voltage detection state. An abnormal core voltage detected state can be intentionally created by using the DIAG mode so that the CVM abnormal voltage detected flag can be checked for failures.

CAUTION

The CVM cannot detect drifts in the voltage being supplied to the on-chip voltage regulator, AWO area, and ISO area, or the increase or decrease of voltage.

10.1.4.1 CVM Reset ($\overline{\text{CVMRES}}$)

If the core voltage exceeds the specified level while high-voltage monitor is enabled (CVMDE.H_D_E = 1), then $\overline{\text{CVMRES}}$ is generated and the power supply to the isolated area is stopped.

If the core voltage falls below the specified level while low voltage monitor is enabled (CVMDE.L_D_E = 1), $\overline{\text{CVMRES}}$ is generated.

For the specifications of $\overline{\text{CVMRES}}$ generation, see **Section 9.4.4, Core Voltage Monitor (CVM) Reset**.

10.1.4.2 CVM Setting

Use the option byte to enable the high-voltage monitor and the low-voltage monitor. For details, see **Section 36.9, Option Bytes**.

10.1.4.3 Diagnostic (DIAG) Mode

This product supports diagnostic mode.

In diagnostic (DIAG) mode, whether the CVM abnormal voltage detection flag is set to 1 can be checked.

In diagnostic mode, $\overline{\text{CVMRES}}$ is not output.

The setting procedure for diagnostic mode is described below.

Set the registers according to this procedure. Otherwise the operation is not guaranteed.

(1) Set CVMDIAG.CVM_DIAG_MASK.*1

(2) Set CVMDIAG.CVM_DIAG.*1

(3) Wait for 12 µs.*2

(4) Read the CVMF register to confirm that the H_V_F and L_V_F bits are set to 1 (if these bits are 0, the CVM does not operate normally, requiring error handling).

(5) Clear CVMDIAG.CVM_DIAG.*1

(6) Clear the CVMF register.*1

(7) Read the CVMF register to confirm that the H_V_F and L_V_F bits are set to 0 (if these bits are 1, go back to step 5 again).

(8) Clear CVMDIAG.CVM_DIAG_MASK.*1

Note 1. Follow the register protection sequence to set CVMF and CVMDIAG registers because these are write-protected registers. For details, see **Section 5, Write-Protected Registers**.

Note 2. At least 50 µs must elapse after the following conditions are fulfilled before step (4) is started.

- Release from HALT mode
- Release from STOP mode
- Release from the reset state, when a reset other than a reset due to the CVM is generated in RUN/HALT mode
- The CPU clock is switched
- Operation of the MainOSC is started or stopped
- Operation of the PLL is started or stopped

10.1.4.4 Clock Supply to the CVM

The clock supply to the CVM is shown in the following table.

Table 10.3 Clock Supply to the CVM

Unit Name	Unit Clock Name	Supply Clock Name
CVM	Register access clock	CPUCLK2

10.1.5 RAM Retention Voltage Indicator (Very-Low-Voltage Detection Circuit:VLVI)

The very-low-voltage detection circuit (VLVI) is used to detect the RAM retention voltage, and continuously compares the power supply voltage REGVCC with the RAM retention voltage V_{VLVI}.

See the Data Sheet for the specifications of the RAM retention voltage level V_{VLVI}.

10.1.5.1 Clock Supply to the VLVI

The clock supply to the VLVI is shown in the following table.

Table 10.4 Clock Supply to the VLVI

Unit Name	Unit Clock Name	Supply Clock Name
VLVI	Register access clock	EMCLK

10.1.5.2 Retention RAM Content Retention

If the power supply voltage REGVCC does not fall below V_{VLVI}, the content of the Retention RAM (RRAM) is retained.

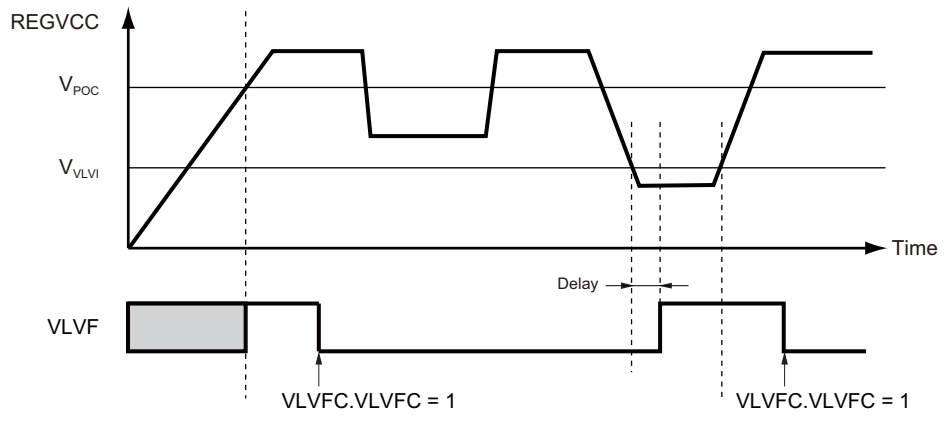
If REGVCC falls below V_{VLVI}, the RRAM content cannot be guaranteed. Thus the entire RRAM must be restored before continuing operation.

If REGVCC falls below the RAM retention voltage (REGVCC < V_{VLVI}), the VLVF.VLVF bit is set.

After that, even if REGVCC exceeds V_{VLVI}, the VLVF.VLVF bit is not cleared automatically. It is cleared by

- setting VLVFC.VLVFC bit to 1.

The following figure illustrates the timing of VLVF.



Note: Delay:
When the REGVCC falls below the V_{VLVI}, the VLVF bit is set to 1 after certain delay. See the Data Sheet for delay time.

Figure 10.2 VLVF Operation Timing

10.1.6 Block Diagram

The block diagram of the supply voltage monitor is shown below.

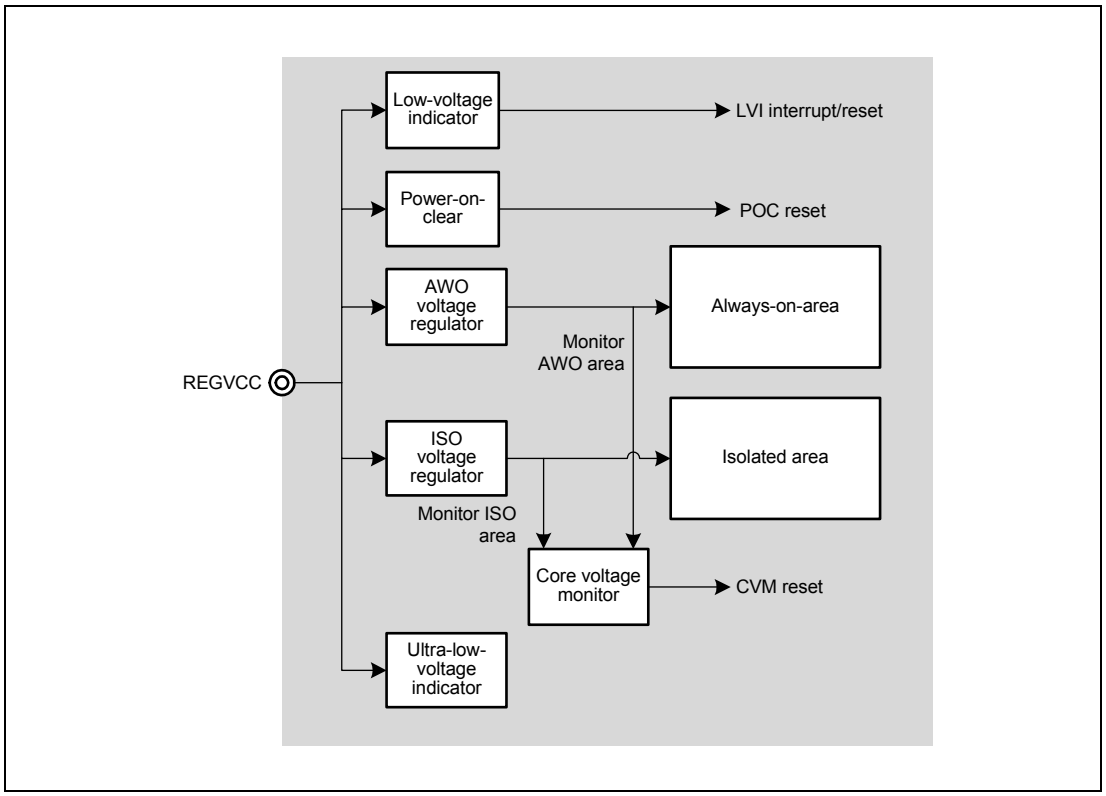


Figure 10.3 Supply Voltage Monitor

10.2 Registers

10.2.1 List of Registers

The following table lists the supply voltage monitor registers.

Table 10.5 Power Supply Voltage Monitor Registers

Register Name	Symbol	Address
Low-voltage indicator reset control register		
LVI control register	LVICNT	FFF8 0A00$_H$
Core voltage monitor control register		
CVM factor register	CVMF	FFF8 3100$_H$
CVM detection enable register	CVMDE	FFF8 3104$_H$
CVM diagnostic mode setting register	CVMDIAG	FFF8 3114$_H$
Very-low-voltage detection control register		
Very-low-voltage detection register	VLVF	FFF8 0980$_H$
Very-low-voltage detection clear register	VLVFC	FFF8 0988$_H$

10.2.2 Low-Voltage Indicator Reset Control Registers

10.2.2.1 LVICNT — LVI Control Register

This register is used to control the Low-Voltage Indicator and to select the LVI detection level.

This register is initialized by a power-up reset PURES.

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFF8 0A00$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	LVIRESMK	LVICNT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 10.6 LVICNT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	LVIRESMK	Mask LVI Reset 0: LVI reset is not masked 1: LVI reset is masked
1, 0	LVICNT[1:0]	Detection Level 0 0: LVI is disabled 0 1: 4.0+/-0.1 V (drop), 4.0+/-0.13 V (rise) 1 0: 3.7+/-0.1 V (drop), 3.7+/-0.13 V (rise) 1 1: 3.5+/-0.1 V (drop), 3.5+/-0.13 V (rise)

NOTE

To use an LVI interrupt, LVI reset must be masked (LVIRESMK = 1) by LVIRESMK.

10.2.3 Core Voltage Monitor Control Registers

10.2.3.1 CVMF — CVM Factor Register

This register records the core voltage failure state generated after the last POC reset.

If the L_V_F or H_V_F bit of this register is set to 1, that bit is not updated until it is initialized by a power-on-clear or by writing 0 to the CVMF.L_V_F or CVMF.H_V_F bit. However, it continuously monitors an error signal from the core voltage monitoring circuit in diagnostic mode.

The correct write sequence using the PROTCMDCVM register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFF8 3100_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H_V_F	L_V_F
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.7 CVMF Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	H_V_F	High-Voltage Failure Detection of the Core Voltage by the CVM 0: No high-voltage failure state is detected 1: High-voltage failure state is detected
0	L_V_F	Low-Voltage Failure Detection of the Core Voltage by the CVM 0: No low-voltage failure state is detected 1: Low-voltage failure state is detected

10.2.3.2 CVMDE — CVM Detection Enable Register

This register is used to indicate the voltage detection enabled or disabled state.

This register is initialized only by the Power-On-Clear.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 3104_H

Value after reset: The value after reset depends on the option byte setting.

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H_D_E	L_D_E
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1*1	0/1*2
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The setting of the option byte OPBT0.CVM_HD_EN is reflected.

Note 2. The setting of the option byte OPBT0.CVM_LD_EN is reflected.

For details on the option byte, see **Section 36.9, Option Bytes**.

Table 10.8 CVMDE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	H_D_E	High-Voltage Monitor Enable 0: High-voltage detection is disabled. 1: High-voltage detection is enabled.
0	L_D_E	Low-Voltage Monitor Enable 0: Low-voltage detection is disabled. 1: Low-voltage detection is enabled.

10.2.3.3 CVMDIAG — CVM Diagnostic Mode Setting Register

This register sets the CVM diagnostic mode.

This register is initialized only by the Power-On-Clear.

For details on the register settings in diagnostic mode, see **Section 10.1.4.3, Diagnostic (DIAG) Mode**.

The correct write sequence using the PROTCMDCVM register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFF8 3114$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CVM_DIAG_MASK	CVM_DIAG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 10.9 CVMDIAG Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CVM_DIAG_MASK	$\overline{\text{CVMRES}}$ Mask Control 0: $\overline{\text{CVMRES}}$ is not masked. 1: $\overline{\text{CVMRES}}$ output is masked.
0	CVM_DIAG	CVM Diagnostic Mode Setting 0: Normal mode 1: Diagnostic mode

10.2.4 Very-Low-Voltage Detection Control Registers

10.2.4.1 VLVF — Very-Low-Voltage Detection Register

The very-low-voltage detection register (VLVF) shows the state of the RAM retention voltage detection.

This register is set upon detection of a voltage below the RAM retention voltage (V_{VLVI}).

If VLVF is set, the Retention RAM content cannot be guaranteed.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 0980_H

Value after power on: $0000\ 0001_H$ This register is not initialized by any reset source.

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 10.10 VLVF Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	VLVF	Very-Low-Voltage Detection Flag 0: Very-low-voltage is not detected. 1: Very-low-voltage is detected. **Note:** Very-low-voltage is the voltage status of REGVCC < RAM retention voltage (V_{VLVI}). For details, see **10.1.5.2, Retention RAM Content Retention**

10.2.4.2 VLVFC — Very-Low-Voltage Detection Clear Register

This register clears the VLVF.VLVF bit.

Access: This register is a write-only register that can be written in 32-bit units.

Address: FFF8 0988$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLVFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 10.11 VLVFC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	VLVFC	Clear VLVF.VLVF bit. 0: Do not clear 1: Clear

Section 11 Clock Controller

This section explains in general about the clock controller.

The first part in this section describes the specific features of the clock controller of the RH850/F1M microcontrollers. The ensuing sections describe the clock oscillators, clock output function, and clock monitor function that make up the clock controller.

11.1 Features of Clock Controller of RH850/F1M

The clock controller of the RH850/F1M microcontrollers has the following features.

- Six on-chip clock oscillators
  - Main Oscillator (MainOSC) with an oscillation frequency of 8 to 24 MHz
  - Sub Oscillator (SubOSC) with an oscillation frequency of 32.768 kHz
  - High Speed Internal Oscillator (HS IntOSC) with a nominal frequency of 8 MHz (Typ.)
  - Low Speed Internal Oscillator (LS IntOSC) with a nominal frequency of 240 kHz (Typ.)
  - PLL (PLL0 and PLL1)
- Spread spectrum PLL circuit (PLL0) that allows adjustment of frequency modulation by parameters as well as unmodulated clock PLL circuit (PLL1) are included.
- Fine management of clock supply to peripheral modules through clock domains
- On-chip clock monitor that detects CPU clock anomalies when the Main Oscillator, High Speed Internal Oscillator, or PLL are in use
- CPU internal clock output (FOUT)

Figure 11.1 shows the schematic diagram of the clock controller.

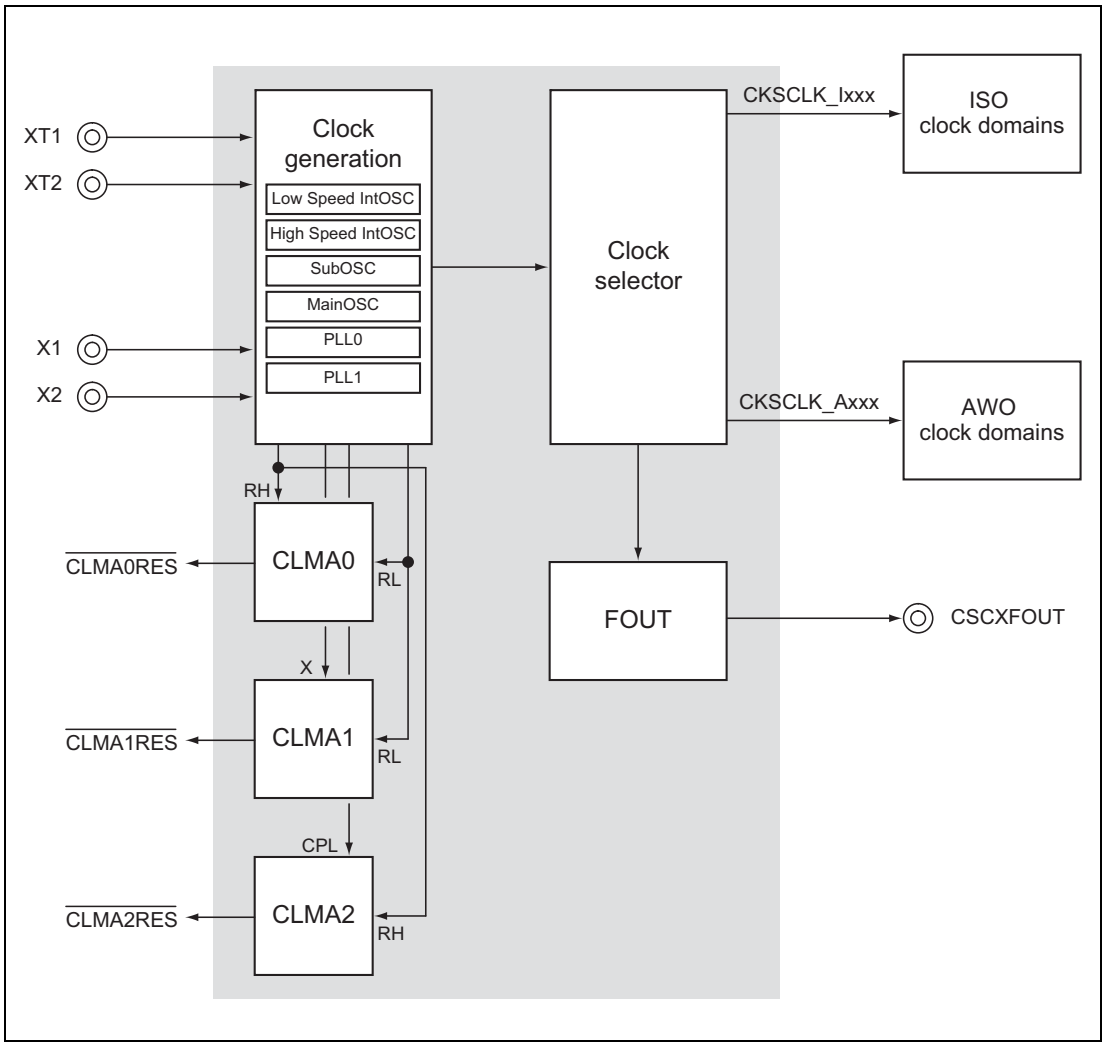


Figure 11.1 Clock Controller Overview

11.2 Configuration of Clock Controller

This section describes the configuration of the clock controller.

The clock controller is composed of clock oscillators and clock generation circuits that generate the clocks for the CPU and the peripheral modules, a clock selector for selecting the optimum clock, and clock domains for the CPU and the peripheral modules.

Figure 11.2 shows the configuration of the clock controller.

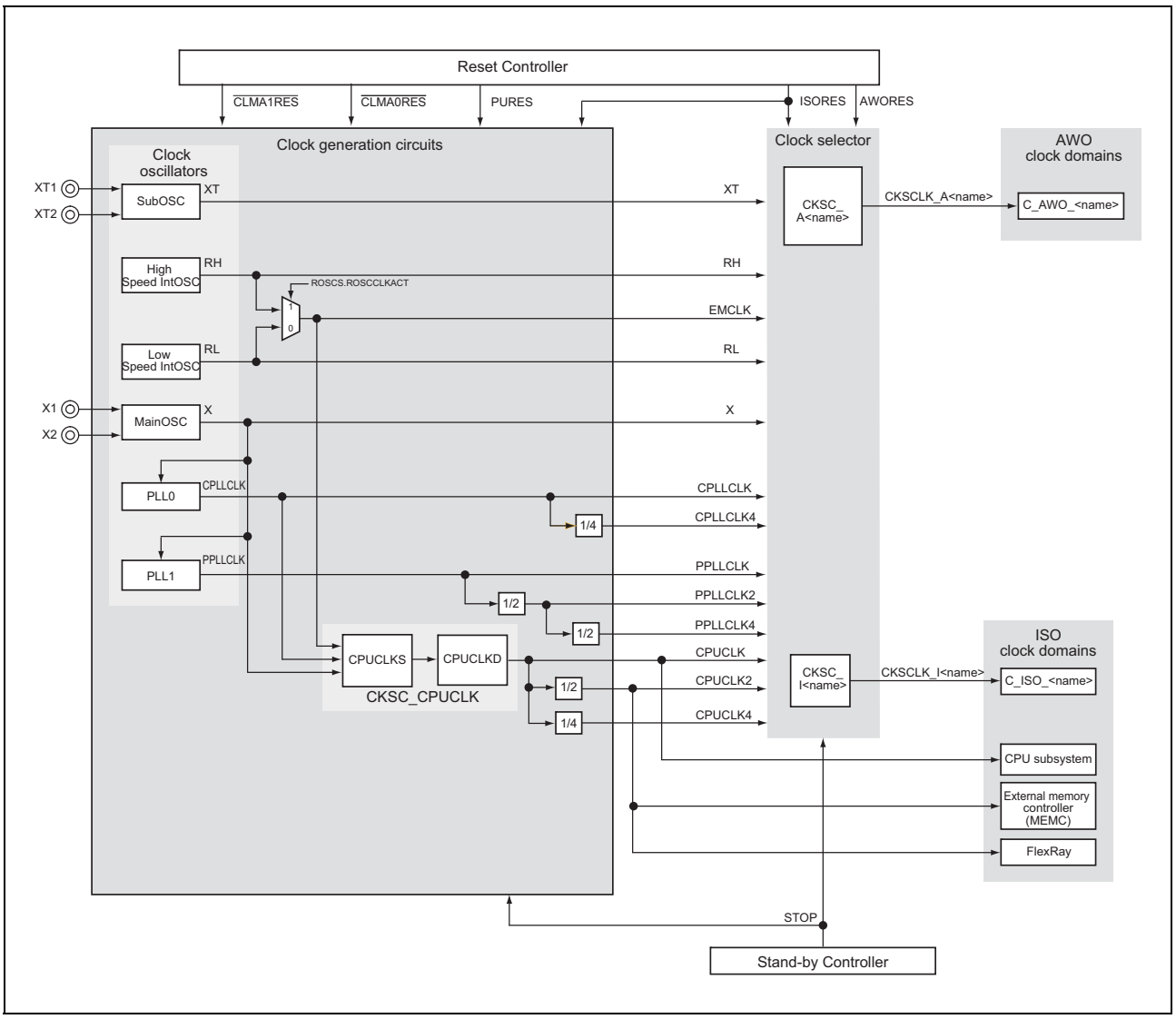


Figure 11.2 Clock Controller Configuration

NOTE

Clock domain and clock control register naming conventions

The clock signals and their control registers, etc., described in this section are named according to the following naming conventions to reflect the power domain or clock domain to which they belong. The placeholder “<name>” is used to identify the target module in the clock domain:

- Clock domain names:
  - C_AWO_<name>: Always-On-Area*1 clock domain
  - C_ISO_<name>: Isolated-Area*1 clock domain
- Domain clock names:
  - CKSCLK_A<name>: Always-On-Area domain clock
  - CKSCLK_I<name>: Isolated-Area domain clock
- Clock selector names:
  - CKSC_A<name>: Always-On-Area clock selector
  - CKSC_I<name>: Isolated-Area clock selector
- Clock selector register names:
  - CKSC_A<name>S_CTL: Always-On-Area source clock selector register
  - CKSC_A<name>D_CTL: Always-On-Area source clock divider register
  - CKSC_I<name>S_CTL: Isolated-Area source clock selector register
  - CKSC_I<name>D_CTL: Isolated-Area source clock divider register

Example

The clock signal CKSCLK_AADCA (placeholder <name> = ADCA) is the clock supplied to the clock domain C_AWO_ADCA in the Always-On-Area. This clock is selected by the clock selector register CKSC_AADCAS_CTL.

Note 1. Always-On-Area and Isolated-Area refer to the power supply domains. Always-On-Area (AWO) is an always-on power supply, and the Isolated_Area (ISO) is a power supply that is switched on or off according to the operation mode. For details, see **Section 39, Power Supply and Power Domains**.

11.2.1 Clock Generation Circuits

Six clock oscillators are provided:

Four clock oscillators (MainOSC, SubOSC, HS IntOSC and LS IntOSC) are located on the Always-On-Area (AWO) and PLL0 and PLL1 are located on the Isolated-Area (ISO).

Main Oscillator (MainOSC)

The MainOSC generates main clock X.
Generation of main clock X requires the connection of an external resonator to X1 and X2.
The clock X is used as the reference clock for PLL0 and PLL1.

Sub Oscillator (SubOSC)

The SubOSC generates the sub-clock XT, which runs at a frequency of 32.768 kHz (Typ.). Generation of the sub clock XT requires the connection of an external resonator to XT1 and XT2.
This clock is mainly used for real-time clock applications.

High Speed Internal Oscillator (HS IntOSC)

The HS IntOSC generates a clock RH, which runs at a frequency of 8 MHz (Typ.).

Low Speed Internal Oscillator (LS IntOSC)

The LS IntOSC generates the clock RL, which runs at a frequency of 240 kHz (Typ.). It starts operation after power up and can not be stopped, hence it is always operating.

PLL0 and PLL1

The PLL0 and the PLL1 generate high speed operation clocks CPLLCLK and PPLLCLK required for normal operation of the microcontroller.

The clocks supplied by the clock oscillators (X, XT, RH, RL, CPLLCLK, PPLLCLK) and their divided clocks (CPLLCLK4, PPLLCLK2, and PPLLCLK4) are all generated in the clock generation circuits.

11.2.2 Clock Selection

The clocks, generated by the clock oscillators are input to the clock selectors CKSC_A<name>/ CKSC_I<name>.

Domain clocks CKSCLK_A<name>/CKSCLK_I<name> are selected by dedicated clock selectors from clocks directly input from the oscillators, or in some cases from clocks that have been divided by clock dividers.

- CKSC_I<name>S_CTL/CKSC_I<name>D_CTL registers: determine the clocks for the Isolated-Area clock domains.
- CKSC_A<name>S_CTL/CKSC_A<name>D_CTL registers: determine the clocks for the Always-On-Area clock domains.

Note that not all available clocks generated by the clock oscillators are input to each clock selector.

The following clocks are supplied to the CPU and related modules from the clock generation circuits.

Emergency Clock (EMCLK)

The emergency clock EMCLK is supplied by the

- HS IntOSC, if it is active
- LS IntOSC, if the HS IntOSC is inactive

The selection is done automatically after CLMA0 reset is occurred, so if the HS IntOSC becomes lower than the limit for any reason, vital modules of the microcontroller are still in operation, since the LS IntOSC does not stop.

CPU Subsystem Clock (CPUCLK)

The CPU Subsystem clock CPUCLK is derived from PLL0 clock CPLLCLK, MainOSC, and EMCLK. The CPU clock selector CKSC_CPUCLK incorporates the selector CPUCLKS, followed by the clock divider CPUCLKD.

The CPUCLK clock divider provides the frequency-divided CPUCLK2 clock signal and CPUCLK4 clock signal derived from CPUCLK.

External Memory Controller and FlexRay Clock

CPUCLK divided by two is supplied to the external memory controller*1 and FlexRay.

Note 1. The supplied clock is further divided by two in the external memory controller.

11.2.3 Clock Domains

The clock controller allows selection of the respective clocks for the CPU and peripheral modules. The clock control scope is called the clock domain. For the correspondence between the CPU and peripheral modules and clock domains, see **Section 11.5.3, Clock Domain Settings**.

11.2.4 Resetting Clock Oscillators

The clock oscillators on the Always-On-Area are reset by the PURES signal.

The HS IntOSC is reset when $\overline{\text{CLMA0RES}}$ is generated and the MainOSC is reset when $\overline{\text{CLMA1RES}}$ is generated.

The clock oscillator on the Isolated-Area is reset by the ISORES signal.

For further details on the clock oscillators, see **Section 11.3, Clock Oscillators**.

CAUTION

For the specifications of the frequencies, acceptable variation, and other parameters of the clock generators, see the Data Sheet.

11.3 Clock Oscillators

11.3.1 Main Oscillator (MainOSC)

The Main Oscillator generates the clock X. X is also used as PLLCLKIN, which is a clock input to the PLL0 and the PLL1.

Figure 11.3 shows the basic configuration and signals of the MainOSC.

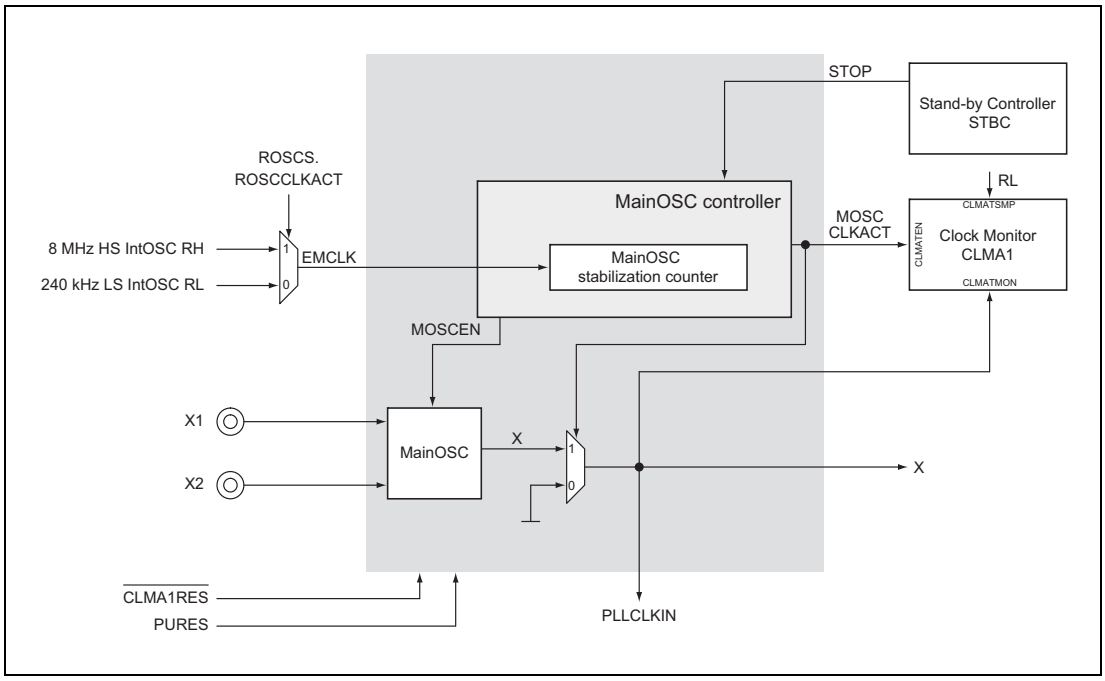


Figure 11.3 Main Oscillator (MainOSC)

MainOSC

The MainOSC stops operating after reset is released. To use the MainOSC, set the MainOSC enable trigger bit (MOSCE.MOSCENTRG) to 1 to start the MainOSC.

MainOSC stabilization

The MOSCST.MOSCCLKST[16:0] bits set the MainOSC oscillation stabilization time.
The MainOSC stabilization counter counts the oscillation stabilization time with EMCLK as the clock source for counting. The oscillation stabilization time can be set to up to 2^{17}-1 EMCLK cycles.
As long as the MainOSC is not stable, the MOSCCLKACT signal disables the X output.
When the MainOSC stabilization counter reaches the value specified in MOSCST.MOSCCLKST[16:0], X is assumed to be stable and MOSCCLKACT switches from 0 to 1 to enable output of X when a waveform is output from MainOSC.

Stable and active X clock is indicated by MOSCS.MOSCCLKACT = 1.

MainOSC amplification gain

By using MOSCC.MOSCAMPSEL[1:0], the MainOSC's input frequency, determined by the external resonator, can be selected in the range of 8 MHz to 24 MHz.

MainOSC STOP requests in stand-by mode

The STOP signal from the Stand-by Controller requests the MainOSC Controller to switch off the X clock in stand-by mode.

The stop request mask bit MOSCSTPM.MOSCSTPMSK controls whether the MainOSC is stopped during stand-by or continues operation:

- MOSCSTPM.MOSCSTPMSK = 0:
  The STOP request signal is not masked, so the MainOSC is stopped during stand-by.
  If the MainOSC was in operation before stand-by, it is automatically re-started after wake-up from stand-by, and the MainOSC stabilization timer counts the oscillation stabilization time.
- MOSCSTPM.MOSCSTPMSK = 1:
  The STOP request signal is masked, so the MainOSC continues to operate during stand-by.

Clock monitor control

The MainOSC activity signal MOSCCLKACT enables or disables supervision by the Clock Monitor CLMA1. In case the MainOSC is inactive (MOSCCLKACT = 0), supervision of its output clock X by CLMA1 is also disabled.

MainOSC enable/disable trigger

The MainOSC can be enabled and disabled by the enable and disable trigger control bits:

- Enable trigger MOSCE.MOSCENTRG = 1 starts the MainOSC.
  Note that setting the enable trigger is only effective if the MainOSC is inactive, i.e. if MOSCS.MOSCCLKACT = 0.
- Disable trigger MOSCE.MOSCDISTRG = 1 stops the MainOSC.
  Note that setting the disable trigger is only effective if the MainOSC is active (MOSCS.MOSCCLKACT = 1).

11.3.2 Sub Oscillator (SubOSC)

The Sub Oscillator generates the sub clock XT. XT typically has a frequency of 32.768 kHz and is used for the Real-time Clock.

Figure 11.4 shows the basic structure and signals of the SubOSC.

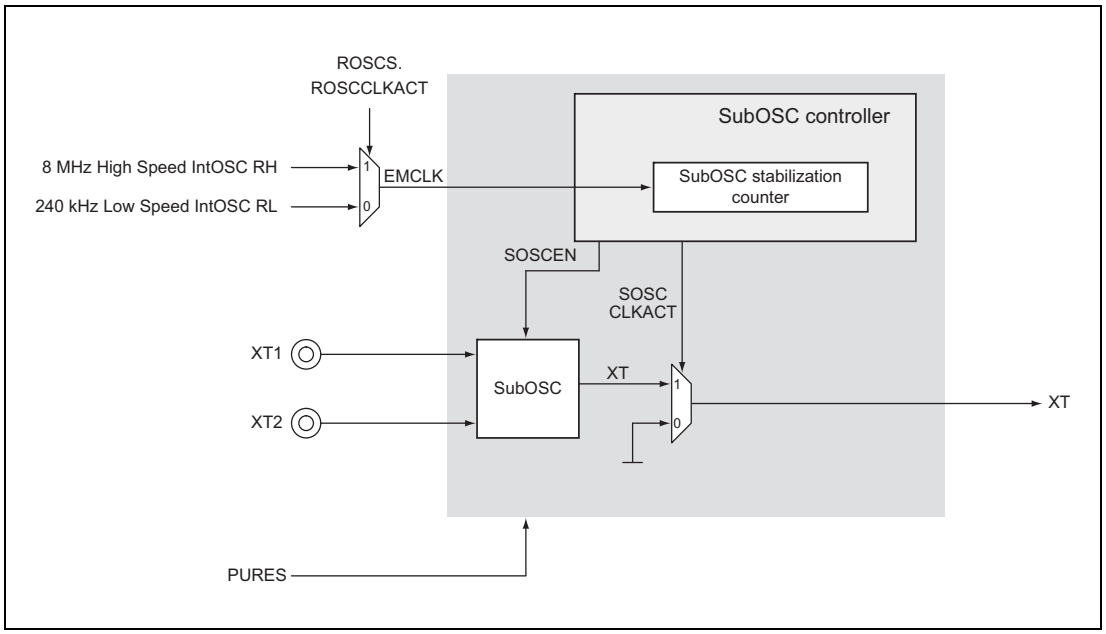


Figure 11.4 Sub Oscillator (SubOSC)

SubOSC enable

The SubOSC stops operating after reset is released. To use the SubOSC, set SubOSC enable trigger bit (SOSCE.SOSCENTRG) to 1 to start the SubOSC.

SubOSC stabilization

The SOSCST.SOSCCLKST[29:0] bits set the SubOSC oscillation stabilization time.
The SubOSC stabilization counter counts the oscillation stabilization time with EMCLK as the clock source for counting.
As long as the SubOSC is not stable, the SOSCCLKACT signal disables the XT output.
When the SubOSC stabilization counter reaches the value specified in SOSCST.SOSCCLKST[29:0], XT is assumed to be stable and SOSCCLKACT switches from 0 to 1 to enable output of XT.
Secure the stabilization time longer than 2 seconds.
Stable and active XT clock is indicated by SOSCS.SOSCCLKACT = 1.

SubOSC frequency

The SubOSC input frequency is 32.768 kHz (Typ.).

SubOSC enable trigger/disable trigger

SubOSC can be enabled or disabled by using enable/disable trigger control bit.

- Enable trigger SOSCE.SOSCENTRG = 1 starts the SubOSC.
  Note that setting the enable trigger is only effective if the SubOSC is inactive, i.e. if SOSCS.SOSCCLKACT = 0.

- Disable trigger SOSCE.SOSCDISTRG = 1 stops the SubOSC.
  Note that setting the disable trigger is only effective if the SubOSC is active; that is, if SOSCS.SOSCCLKACT = 1.

11.3.3 High Speed Internal Oscillator (HS IntOSC)

The High Speed Internal Oscillator generates the clock RH. RH has a nominal frequency of 8 MHz.

Figure 11.5 shows the basic structure and signals of the HS IntOSC.

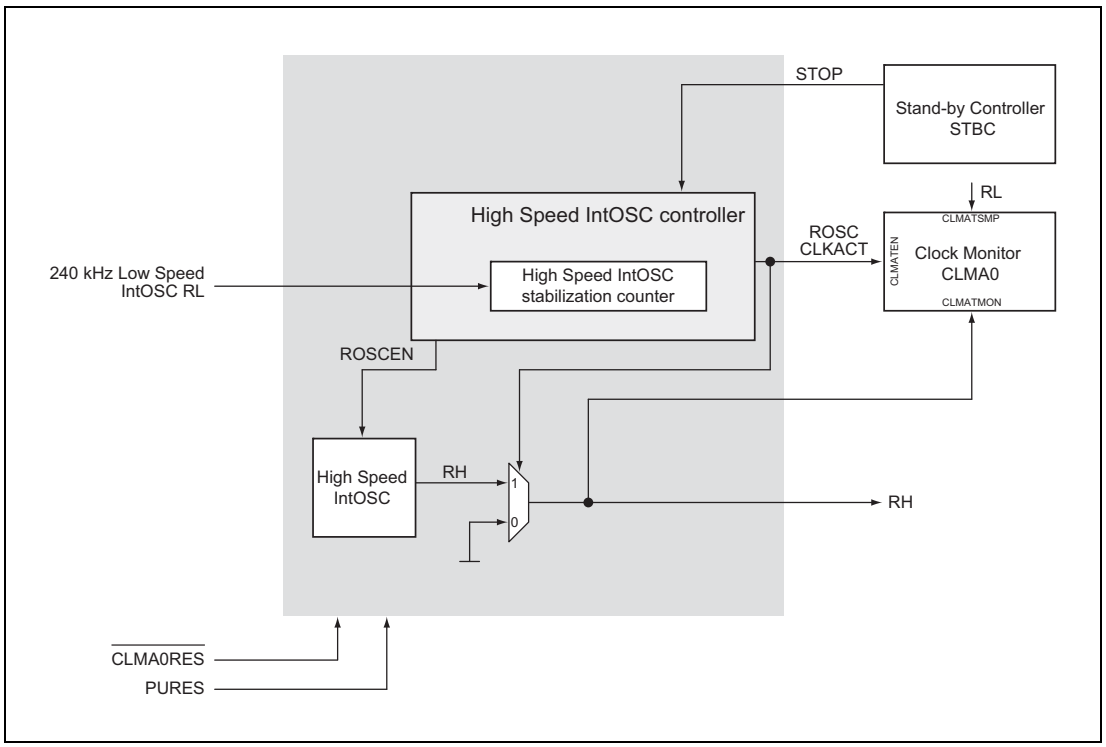


Figure 11.5 High Speed Internal Oscillator (HS IntOSC)

After reset release the HS IntOSC starts operation.

NOTE

The HS IntOSC can neither be stopped nor started by software. It can only be stopped in stand-by mode. On the other hand, when CLMA0 is reset, the HS IntOSC can be stopped by software.

HS IntOSC stabilization

HS IntOSC outputs RH when it is stabilized.
Stable and active RH clock is indicated by ROSCS.ROSCCLKACT = 1.

HS IntOSC STOP requests in stand-by mode

The STOP signal from the stand-by controller requests the HS IntOSC Controller to switch off the RH clock in stand-by mode.
The stop request mask bit ROSCSTPM.ROSCSTPMSK controls whether the HS IntOSC is stopped during stand-by or continues operation:

- ROSCSTPM.ROSCSTPMSK = 0:
  The STOP request signal is not masked, so the HS IntOSC is stopped during stand-by and automatically restarted after wake-up from stand-by.
  However, the STOP request is masked under the following conditions, even if ROSCSTPM.ROSCSTPMSK = 0. Therefore, the HS IntOSC will continue to operate even in standby mode.
  - If the stop mask is set (CKSC_xxxx_STPM = 0000_0003_H) for a clock domain for which the HS IntOSC is selected
  - If the low power sampler (LPS) is operating
- ROSCSTPM.ROSCSTPMSK = 1:
  The STOP request signal is masked, so the HS IntOSC continues to operate during stand-by.

Clock Monitor control

The HS IntOSC activity signal ROSCCLKACT enables or disables supervision by the Clock Monitor CLMA0. In case the HS IntOSC is inactive (ROSCCLKACT = 0), supervision of its output clock by CLMA0 is also disabled.

The HS IntOSC clock RH is used as the sampling clock for Clock Monitor CLMA2.

HS IntOSC disable trigger

The disable trigger, ROSCE.ROSCDISTRG = 1, stops the HS IntOSC.

11.3.4 Low Speed Internal Oscillator (LS IntOSC)

The Low Speed Internal Oscillator generates the clock RL. RL has a nominal frequency of 240 kHz.

Figure 11.6 shows the basic structure and signals of the LS IntOSC.

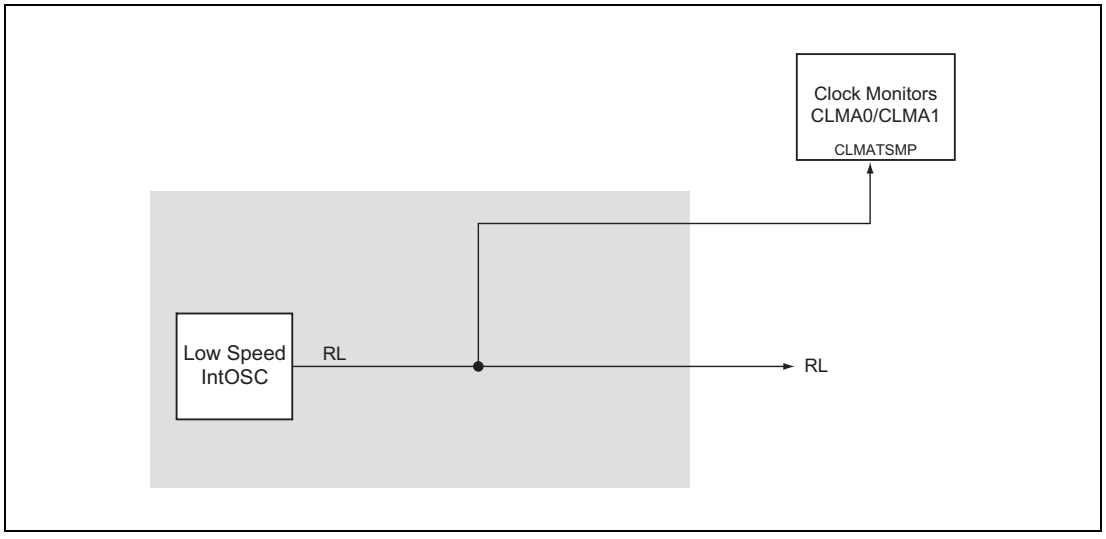


Figure 11.6 Low Speed Internal Oscillator (LS IntOSC)

After reset release the LS IntOSC starts operation. It can not be stopped.

The LS IntOSC clock RL is used as the sampling clock for the Clock Monitors CLMA0 and CLMA1.

11.3.5 PLL0/PLL1

The Main Oscillator clock X is input to the Phase-Locked Loops (PLL) clock oscillator. The PLL0 output clock CPLLCLK and the PLL1 output clock PPLLCLK are served as the main operation clocks for the microcontroller.

Figure 11.7 shows the basic structure and signals of the PLL0 and PLL1.

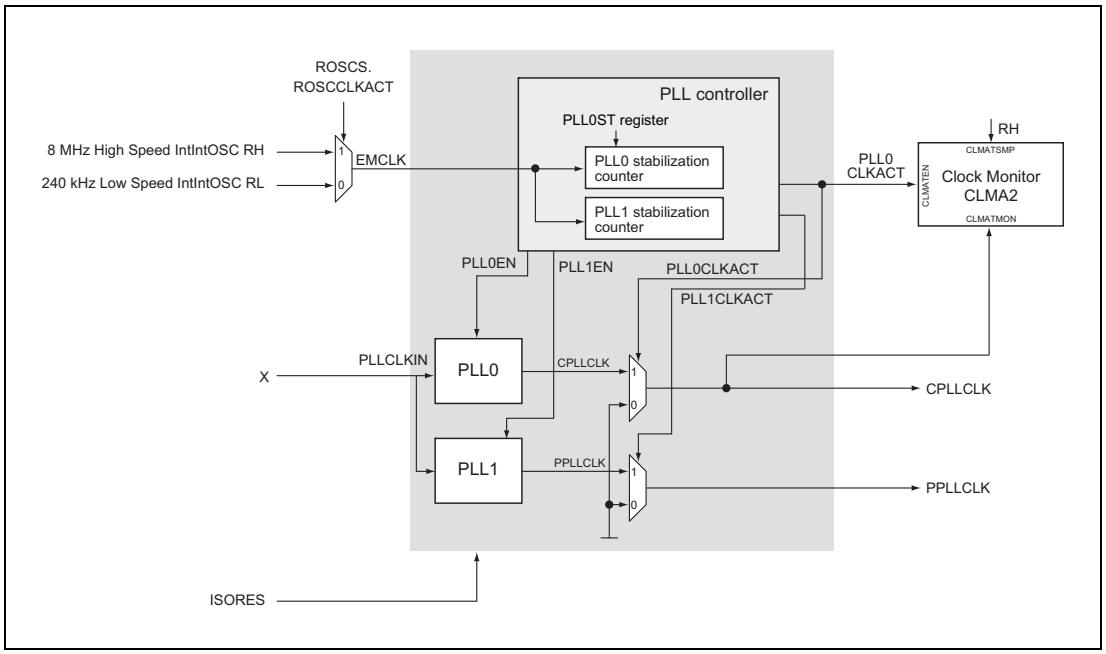


Figure 11.7 PLL

PLL0/PLL1 enable

The PLL0 stops operating after reset is deasserted. To use the PLL0, set the PLL0 enable trigger bit (PLL0E.PLL0ENTRG) to 1 to start the PLL0.

The PLL1 also has been stopped after reset is deasserted. To use the PLL1, set the PLL1 enable trigger bit (PLL1E.PLL1ENTRG) to 1 to start the PLL1.

PLL0/PLL1 stabilization

The PLL0 stabilization time is set in PLL0ST.PLL0CLKST[12:0]. The PLL0 stabilization counter starts counting the stabilization time by using EMCLK as the counting source. As long as the PLL0 is not stable, the PLL0CLKACT signal disables the CPLLCLK output. When the PLL0 stabilization counter reaches the predefined value in PLL0ST.PLL0CLKST[12:0], CPLLCLK is assumed to be stable and PLL0CLKACT switches from 0 to 1 to enable output of CPLLCLK.

Once the PLL1 starts operating, the PLL1 stabilization counter starts counting the stabilization time. As long as the PLL1 is unstable, the PLL1CLKACT signal disables the PPLLCLK output.

When the PLL1 stabilization counter has reached a predefined value, PPLLCLK is assumed to be stable and PLL1CLKACT switches from 0 to 1 to enable output of PPLLCLK.

The stable and active state of the CPLLCLK clock is indicated by PLL0S.PLL0CLKACT = 1 whereas the stable and active state of the PPLLCLK clock is indicated by PLL1S.PLL1CLKACT = 1.

PLL0/PLL1 in stand-by modes

In STOP mode, the PLL0 and PLL1 are automatically disabled and resume operation after wake-up from STOP mode if they were operating before entering STOP mode.

The PLL0 and PLL1 are also automatically disabled when transitioning to DeepSTOP mode. However, after restoring from DeepSTOP mode, the PLL0 and the PLL1 need to be reconfigured.

In Cyclic RUN and Cyclic STOP mode, the PLL0 and the PLL1 are not available. Do not enable the PLL0 and the PLL1 in Cyclic RUN mode.

Clock Monitor control

The PLL0 activity signal PLL0CLKACT enables or disables supervision by the Clock Monitor CLMA2. In case the PLL is inactive (PLL0CLKACT = 0), supervision of the output clocks CPLLCLK by CLMA2 is also disabled.

PLL0 and PLL1 Enable/Disable Trigger

The PLL0 and PLL1 can be enabled and disabled by the enable and disable trigger control bits:

- Enable trigger PLL0E.PLL0ENTRG to 1 starts the PLL0.
  Note that setting the PLL0 enable trigger is only effective if the PLL0 is inactive, i.e. if PLL0S.PLL0CLKACT = 0.
  Enable trigger PLL1E.PLL0ENTRG to 1 starts the PLL1.
  Note that setting the PLL1 enable trigger is only effective if the PLL1 is inactive, i.e. if PLL1S.PLL1CLKACT = 0.
- Disable trigger PLL0E.PLL0DISTRG to 1 stops the PLL0.
  Note that setting the PLL0 disable trigger is only effective if the PLL0 is active, i.e. if PLL0S.PLL0CLKACT = 1.
  Disable trigger PLL1E.PLL1DISTRG to 1 stops the PLL1.
  Note that setting the PLL1 disable trigger is only effective if the PLL1 is active, i.e. if PLL1S.PLL1CLKACT = 1.

11.3.5.1 PLL0 Parameters

PLL0 operation is configured by a set of parameters loaded from the control register PLL0C.

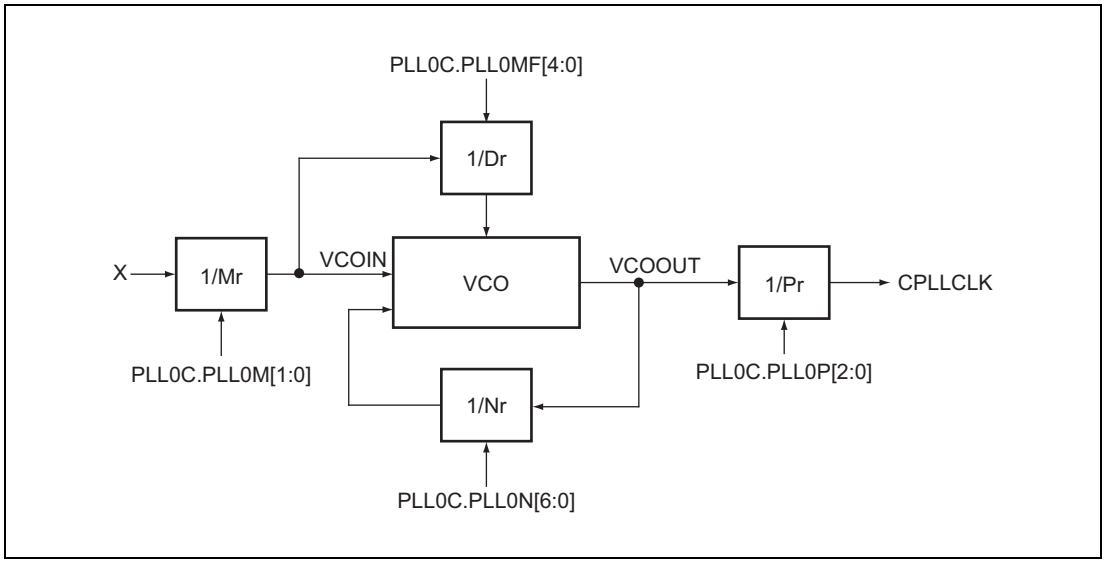


Figure 11.8 PLL0 Circuit Configuration

Operating modes

The PLL0 has two operating modes.

- When PLL0C.PLL0MD is 0: PLL mode with fixed output frequency
- When PLL0C.PLL0MD is 1: SSCG (Spread Spectrum Clock Generator) mode with modulated output frequency

CAUTION

Set the modulation mode to center spread modulation (PLL0C.PLL0SMD = 1) to use the PLL0 in the SSCG mode (PLL0C.PLL0MD = 1).

CPLLCLK

The CPLLCLK is a source clock of the CPU subsystem clock CPUCLK.

The clock frequency $f_{CPLLCLK}$ is integer fraction of the VCO output frequency f_{VCOOUT}.

$$f_{VCOOUT} = f_X \times (Nr / Mr)$$

$f_{CPLLCLK}$ is calculated as follows:

$$f_{CPLLCLK} = f_X \times (Nr / Mr) \times 1/Pr$$

The values Nr, Mr, and Pr are derived from the PLL0C register bits:

- Nr = PLL0C.PLL0N[6:0] + 1
- Mr = PLL0C.PLL0M[1:0] + 1
- Pr is determined by PLL0C.PLL0P[2:0] according to the table below.

PLL0C.PLL0P[2:0]	Pr
010_B	4
011_B	8
100_B	16

Frequency modulation

When frequency modulation is enabled (PLL0C.PLL0MD = 1), additional parameters must be set by the PLL0C register.

- PLL0C.PLL0MF[4:0] determines the frequency modulation cycle.
- PLL0C.PLL0ADJ[2:0] determines the frequency modulation range.

For how to calculate modulation frequency, see **Section 11.4.2.14, PLL0C — PLL0 Control Register**.

PLL0 output frequency

PLL0C.PLL0FVV[1:0] needs to be set according to the VCO output frequency.

For details of the PLLC0 register, see **Section 11.4.2.14, PLL0C — PLL0 Control Register**.

11.3.5.2 PLL1 Parameters

The PLL1 is configured by a set of parameters, loaded from the control register PLL1C.

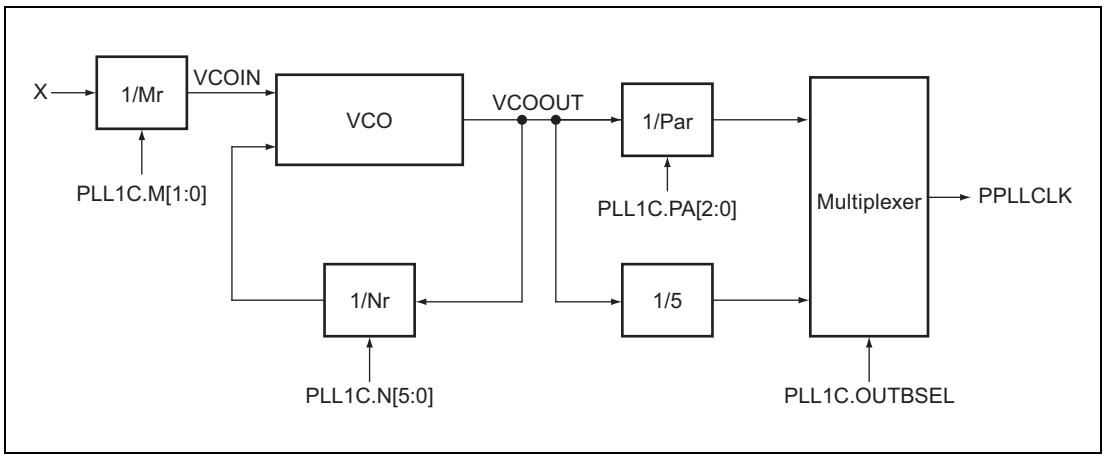


Figure 11.9 PLL1 Circuit Configuration

PPLLCLK

PPLLCLK is the source of the clock supply for several functional modules and its frequency must not exceed 80 MHz.

The frequency of $f_{PPLLCLK}$ is one divided by the integer that is the VCO output frequency f_{VCOOUT} divided by Par or by the 1/5 clock divider.

$$f_{VCOOUT} = f_X \times (Nr / Mr)$$

The frequency $f_{PPLLCLK}$ is calculated as follows:

$$f_{PPLLCLK} = f_X \times (Nr / Mr) \times 1/Par$$

or

$$f_{PPLLCLK} = f_X \times (Nr / Mr) \times 1/5$$

The values Nr, Mr, and Par are derived from the PLL1C register bits:

- Nr = PLL1C.N[5:0] + 1
- Mr = PLL1C.M[1:0] + 1
- Par is determined by PLL1C.PA[2:0] according to the table below:

PLL1C.PA[2:0]	Par
010_B	4
011_B	6
100_B	8
101_B	16

For details of the PLLC1 register, see **Section 11.4.2.18, PLL1C — PLL1 Control Register**.

11.4 Registers

11.4.1 List of Registers

The registers of the clock controller are listed below.

Table 11.1 List of Clock Controller Registers (1/2)

Register Name	Symbol	Address
Clock oscillator registers:		
MainOSC enable register	MOSCE	FFF8 1100$_H$
MainOSC status register	MOSCS	FFF8 1104$_H$
MainOSC control register	MOSCC	FFF8 1108$_H$
MainOSC stabilization time register	MOSCST	FFF8 110C$_H$
MainOSC stop mask register	MOSCSTPM	FFF8 1118$_H$
SubOSC enable register	SOSCE	FFF8 1200$_H$
SubOSC status register	SOSCS	FFF8 1204$_H$
SubOSC stabilization time register	SOSCST	FFF8 120C$_H$
HS IntOSC enable register	ROSCE	FFF8 1000$_H$
HS IntOSC status register	ROSCS	FFF8 1004$_H$
HS IntOSC stop mask register	ROSCSTPM	FFF8 1018$_H$
PLL0 enable register	PLL0E	FFF8 9000$_H$
PLL0 status register	PLL0S	FFF8 9004$_H$
PLL0 control register	PLL0C	FFF8 9008$_H$
PLL0 stabilization time register	PLL0ST	FFF8 900C$_H$
PLL1 enable register	PLL1E	FFF8 9100$_H$
PLL1 status register	PLL1S	FFF8 9104$_H$
PLL1 control register	PLL1C	FFF8 9108$_H$
Clock selector registers:		
C_AWO_WDTA clock divider register	CKSC_AWDTAD_CTL	FFF8 2000$_H$
C_AWO_WDTA clock divider active register	CKSC_AWDTAD_ACT	FFF8 2008$_H$
C_AWO_WDTA stop mask register	CKSC_AWDTAD_STPM	FFF8 2018$_H$
C_AWO_TAUJ source clock selection register	CKSC_ATAUJS_CTL	FFF8 2100$_H$
C_AWO_TAUJ source clock active register	CKSC_ATAUJS_ACT	FFF8 2108$_H$
C_AWO_TAUJ clock divider register	CKSC_ATAUJD_CTL	FFF8 2200$_H$
C_AWO_TAUJ clock divider active register	CKSC_ATAUJD_ACT	FFF8 2208$_H$
C_AWO_TAUJ stop mask register	CKSC_ATAUJD_STPM	FFF8 2218$_H$
C_AWO_RTCA source clock selection register	CKSC_ARTCAS_CTL	FFF8 2300$_H$
C_AWO_RTCA source clock active register	CKSC_ARTCAS_ACT	FFF8 2308$_H$
C_AWO_RTCA clock divider register	CKSC_ARTCAD_CTL	FFF8 2400$_H$
C_AWO_RTCA clock divider active register	CKSC_ARTCAD_ACT	FFF8 2408$_H$
C_AWO_RTCA stop mask register	CKSC_ARTCAD_STPM	FFF8 2418$_H$
C_AWO_ADCA source clock selection register	CKSC_AADCAS_CTL	FFF8 2500$_H$
C_AWO_ADCA source clock active register	CKSC_AADCAS_ACT	FFF8 2508$_H$
C_AWO_ADCA clock divider register	CKSC_AADCAD_CTL	FFF8 2600$_H$
C_AWO_ADCA clock divider active register	CKSC_AADCAD_ACT	FFF8 2608$_H$
C_AWO_ADCA stop mask register	CKSC_AADCAD_STPM	FFF8 2618$_H$
C_AWO_FOUT source clock selection register	CKSC_AFOUTS_CTL	FFF8 2700$_H$

Table 11.1 List of Clock Controller Registers (2/2)

Register Name	Symbol	Address
C_AWO_FOUT source clock active register	CKSC_AFOUTS_ACT	FFF8 2708$_H$
C_AWO_FOUT stop mask register	CKSC_AFOUTS_STPM	FFF8 2718$_H$
C_ISO_CPUCLK source clock selection register	CKSC_CPUCLKS_CTL	FFF8 A000$_H$
C_ISO_CPUCLK source clock active register	CKSC_CPUCLKS_ACT	FFF8 A008$_H$
C_ISO_CPUCLK clock divider register	CKSC_CPUCLKD_CTL	FFF8 A100$_H$
C_ISO_CPUCLK clock divider active register	CKSC_CPUCLKD_ACT	FFF8 A108$_H$
C_ISO_PERI1 source clock selection register	CKSC_IPERI1S_CTL	FFF8 A200$_H$
C_ISO_PERI1 source clock active register	CKSC_IPERI1S_ACT	FFF8 A208$_H$
C_ISO_PERI2 source clock selection register	CKSC_IPERI2S_CTL	FFF8 A300$_H$
C_ISO_PERI2 source clock active register	CKSC_IPERI2S_ACT	FFF8 A308$_H$
C_ISO_LIN source clock selection register	CKSC_ILINS_CTL	FFF8 A400$_H$
C_ISO_LIN source clock active register	CKSC_ILINS_ACT	FFF8 A408$_H$
C_ISO_ADCA source clock selection register	CKSC_IADCAS_CTL	FFF8 A500$_H$
C_ISO_ADCA source clock active register	CKSC_IADCAS_ACT	FFF8 A508$_H$
C_ISO_ADCA clock divider register	CKSC_IADCAD_CTL	FFF8 A600$_H$
C_ISO_ADCA clock divider active register	CKSC_IADCAD_ACT	FFF8 A608$_H$
C_ISO_LIN clock divider register	CKSC_ILIND_CTL	FFF8 A800$_H$
C_ISO_LIN clock divider active register	CKSC_ILIND_ACT	FFF8 A808$_H$
C_ISO_LIN stop mask register	CKSC_ILIND_STPM	FFF8 A818$_H$
C_ISO_CAN source clock selection register	CKSC_ICANS_CTL	FFF8 A900$_H$
C_ISO_CAN source clock active register	CKSC_ICANS_ACT	FFF8 A908$_H$
C_ISO_CAN stop mask register	CKSC_ICANS_STPM	FFF8 A918$_H$
C_ISO_CANOSC clock divider register	CKSC_ICANOSCD_CTL	FFF8 AA00$_H$
C_ISO_CANOSC clock divider active register	CKSC_ICANOSCD_ACT	FFF8 AA08$_H$
C_ISO_CANOSC stop mask register	CKSC_ICANOSCD_STPM	FFF8 AA18$_H$
C_ISO_CSI source clock selection register	CKSC_ICSIS_CTL	FFF8 AB00$_H$
C_ISO_CSI source clock active register	CKSC_ICSIS_ACT	FFF8 AB08$_H$
C_ISO_IIC source clock selection register	CKSC_IIICS_CTL	FFF8 AC00$_H$
C_ISO_IIC source clock active register	CKSC_IIICS_ACT	FFF8 AC08$_H$

11.4.2 Clock Oscillator Registers

11.4.2.1 MOSCE — MainOSC Enable Register

This register is used to start and stop the MainOSC.

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the power-up reset signals PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1100_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCDISTRG	MOSCENTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.2 MOSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	MOSCDISTRG	MainOSC Disable Trigger*1 0: No function 1: Stops MainOSC
0	MOSCENTRG	MainOSC Enable Trigger 0: No function 1: Starts MainOSC

Note 1. <Recommended procedure for stopping MainOSC using MOSCDISTRG>
1. Check that there is no clock domain for which MainOSC is selected.
If MainOSC is selected for a clock domain, disable the setting or select a clock source other than MainOSC.
2. Stop the MainOSC (MOSCE.MOSCDISTRG = 1).
3. Confirm that the MainOSC has been stopped (MOSCS.MOSCCLKACT = 0).

11.4.2.2 MOSCS — MainOSC Status Register

This register provides active status information about the MainOSC.

This register is initialized by the power-up reset signals PURES and $\overline{\text{CLMA1RES}}$.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 1104_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1*1	0*1
	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCCLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 11.3 MOSCS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	MOSCCLKACT	MainOSC Active Status 0: MainOSC is inactive 1: MainOSC is active
1, 0	Reserved	When read, an undefined value is returned.

11.4.2.3 MOSCC — MainOSC Control Register

This register is used to specify amplification gain of the MainOSC.

This register is initialized by the power-up reset signals PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1108_H

Value after reset: 0000 0004_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCAMPSEL [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.4 MOSCC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	MOSCAMPSEL [1:0]	MainOSC Frequency Selection

MOSCAMPSEL [1:0]	Amplification Gain	Typical Frequency Range for External Resonator (Depends on Oscillator Type)
00_B	High	20 MHz < f_X ≤ 24 MHz
01_B	Mid-high	16 MHz < f_X ≤ 20 MHz
10_B	Mid-low	8 MHz < f_X ≤ 16 MHz
11_B	Low	8 MHz

CAUTION

Set this register when MainOSC is stopped.

11.4.2.4 MOSCST — MainOSC Stabilization Time Register

This register determines the MainOSC stabilization time.

This register is initialized by the power-up reset signals PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 110C$_H$

Value after reset: 0000 44C0$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSC CLKST 16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOSCCLKST[15:0]															
Value after reset	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.5 MOSCST Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16 to 0	MOSC CLKST[16:0]	The MOSCCLKST[16:0] bits specify the count value for the MainOSC stabilization counter. • If the HS IntOSC is active (ROSCS.ROSCCLKACT = 1): Stabilization time = MOSCCLKST[16:0] / f_{RH} • If the HS IntOSC is inactive (ROSCS.ROSCCLKACT = 0): Stabilization time = MOSCCLKST[16:0] / f_{RL}

NOTE

See the Data Sheet for information about the MainOSC stabilization time.

CAUTION

Set this register when MainOSC is stopped.

11.4.2.5 MOSCSTPM — MainOSC Stop Mask Register

This register is initialized by the power-up reset signals PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1118$_H$

Value after reset: 0000 0002$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 11.6 MOSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	MOSCSTPMSK	MainOSC Stop Request Mask in Stand-by Mode 0: MainOSC stops operation in stand-by mode. 1: MainOSC continues operation in stand-by mode. Note that the MainOSC can be stopped in the case the MainOSC disable trigger MOSCE.MOSCDISTRG is set to 1 regardless of the setting of this bit.

11.4.2.6 SOSCE — SubOSC Enable Register

This register is used to start and stop the SubOSC.

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the power-up reset signal PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1200_H

Value after reset: $0000\ 0000_H$.

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOSCDISTRG	SOSCENTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.7 SOSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SOSCDISTRG	SubOSC Disable Trigger*1 0: No function 1: Stops SubOSC
0	SOSCENTRG	SubOSC Enable Trigger 0: No function 1: Starts SubOSC

Note 1. <Recommended procedure for stopping SubOSC using SOSCDISTRG>
1. Check that there is no clock domain for which SubOSC is selected.
If SubOSC is selected for a clock domain, disable the setting or select a clock source other than SubOSC.
2. Stop the SubOSC (SOSCE.SOSCDISTRG = 1).
3. Confirm that the SubOSC has been stopped (SOSCS.SOSCCLKACT = 0).

11.4.2.7 SOSCS — SubOSC Status Register

This register provides active status information about the SubOSC.

This register is initialized by the power-up reset signal PURES.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 1204_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1*1	0*1
	—	—	—	—	—	—	—	—	—	—	—	—	—	SOSCCLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 11.8 SOSCS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	SOSCCLKACT	SubOSC Active Status 0: SubOSC is inactive 1: SubOSC is active
1, 0	Reserved	When read, an undefined value is returned.

11.4.2.8 SOSCST — SubOSC Stabilization Time Register

This register determines the SubOSC stabilization time.

This register is initialized by the power-up reset signal PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 120C$_H$

Value after reset: 010C 8E00$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SOSCCLKST[29:16]													
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SOSCCLKST[15:0]															
Value after reset	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.9 SOSCST Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 0	SOSCCLKST [29:0]	The SOSCCLKST[29:0] bits specify the count value for the SubOSC stabilization time counter. • If the HS IntOSC is active (ROSCS.ROSCCLKACT = 1): Stabilization time = SOSCCLKST[29:0] / f_{RH} • If the HS IntOSC is inactive (ROSCS.ROSCCLKACT = 0): Stabilization time = SOSCCLKST[29:0] / f_{RL}

NOTE

See the Data Sheet for information about the SubOSC stabilization time.

CAUTION

Set this register when SubOSC is stopped.

11.4.2.9 ROSCE — HS IntOSC Enable Register

This register is used to stop the HS IntOSC operation.

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the power-up reset signals PURES and $\overline{\text{CLMA0RES}}$.

CAUTION

Set the ROSCE.ROSCDISTRG bit only when the $\overline{\text{CLMA0RES}}$ has occurred. In other cases, setting this bit is prohibited.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1000_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCDISTRG	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 11.10 ROSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ROSCDISTRG	HS IntOSC Disable Trigger 0: No function 1: Stops HS IntOSC
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

11.4.2.10 ROSCS — HS IntOSC Status Register

This register provides active status information about the HS IntOSC.

This register is initialized by the power-up reset signals PURES and $\overline{\text{CLMA0RES}}$.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 1004_H

Value after reset: 0000 0007_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1*1	0*1
	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCCLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 11.11 ROSCS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	ROSCCLKACT	HS IntOSC Active Status 0: HS IntOSC is inactive 1: HS IntOSC is active
1, 0	Reserved	When read, an undefined value is returned.

11.4.2.11 ROSCSTPM — HS IntOSC Stop Mask Register

This register is initialized by the power-up reset signals PURES and $\overline{\text{CLMA0RES}}$.

Access: This register can be read or written in 32-bit units.

Address: FFF8 1018_H

Value after reset: 0000 0002_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 11.12 ROSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ROSCSTPMSK	HS IntOSC Stop Request Mask in Stand-by Mode 0: HS IntOSC stops operation in stand-by mode. 1: HS IntOSC continues operation in stand-by mode. Note that the HS IntOSC can be stopped in the case the HS IntOSC disable trigger ROSCE.ROSCDISTRG is set to 1 regardless of the setting of this bit.

11.4.2.12 PLL0E — PLL0 Enable Register

This register is used to start and stop PLL0.

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the ISORES signal.

Access: This register can be read or written in 32-bit units.

Address: FFF8 9000_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0DISTRG	PLL0ENTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.13 PLL0E Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	PLL0DISTRG	PLL0 Disable Trigger*1 0: No function 1: Stops PLL0
0	PLL0ENTRG	PLL0 Enable Trigger*2 0: No function 1: Starts PLL0

Note 1. <Recommended procedure for stopping PLL0 using PLL0E.PLL0DISTRG>
1. Check that there is no clock domain for which PLL0 is selected.
If PLL0 is selected for a clock domain, disable the setting or select a clock source other than PLL0.
2. Stop the PLL0 (PLL0E.PLL0DISTRG = 1).
3. Confirm that the PLL0 has been stopped (PLL0S.PLL0CLKACT = 0).

Note 2. Before starting PLL0 using PLL0ENTRG, confirm that MainOSC is operating.

11.4.2.13 PLL0S — PLL0 Status Register

This register provides active status information about PLL0.

This register is initialized by the ISORES signal.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 9004_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1*1	0*1
	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0CLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 11.14 PLL0S Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	PLL0CLKACT	PLL0 Active Status 0: PLL0 is inactive 1: PLL0 is active
1, 0	Reserved	When read, an undefined value is returned.

11.4.2.14 PLL0C — PLL0 Control Register

This register is used to set the PLL0 output clock frequency $f_{CPLLCLK}$.

This register can only be written, if PLL0 is disabled.

This register is initialized by the ISORES signal.

Access: This register can be read or written in 32-bit units.

Address: FFF8 9008_H

Value after reset: 6000 $123B_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PLL0FVV[1:0]		PLL0MF[4:0]					—	PLL0ADJ[2:0]			—	—	—	—
Value after reset	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PLL0MD	PLL0SMD	PLL0M[1:0]		PLL0P[2:0]			—	PLL0N[6:0]						
Value after reset	0	0	0	1	0	0	1	0	0	0	1	1	1	0	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.15 PLL0C Register Contents (1/4)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30, 29	PLL0FVV[1:0]	VCO Output Frequency Range Setting

PLL0FVV1	PLL0FVV0	VCO Output Frequency Range
0	0	320 to 360 MHz
0	1	360 to 400 MHz
1	0	400 to 440 MHz
1	1	440 to 480 MHz

Table 11.15 PLL0C Register Contents (2/4)

Bit Position	Bit Name	Function
28 to 24	PLL0MF[4:0]	Frequency Modulation Cycle Setting The modulation frequency is calculated as: Modulation frequency = (f_X / Mr) / (MFD × 4) Example: When f_X = 8 MHz, Mr = 1, and MFD = 00011_B = 20, Modulation frequency = (8/1) / (20 × 4) = 100 [kHz] For the modulation frequency setting range, see the Data Sheet.

PLL0MF4	PLL0MF3	PLL0MF2	PLL0MF1	PLL0MF0	Modulation Frequency Division Ratio MFD
0	0	0	0	0	10
0	0	0	0	1	12
0	0	0	1	0	18
0	0	0	1	1	20
0	0	1	0	0	22
0	0	1	0	1	26
0	0	1	1	0	28
0	0	1	1	1	30
0	1	0	0	0	34
0	1	0	0	1	38
0	1	0	1	0	40
0	1	0	1	1	44
0	1	1	0	0	50
0	1	1	0	1	56
0	1	1	1	0	58
0	1	1	1	1	60
1	0	0	0	0	62
1	0	0	0	1	66
1	0	0	1	0	72
1	0	0	1	1	76
1	0	1	0	0	80
1	0	1	0	1	84
1	0	1	1	0	86
1	0	1	1	1	100
1	1	0	0	0	120
1	1	0	0	1	126
1	1	0	1	0	134
1	1	0	1	1	150
1	1	1	0	0	166
1	1	1	0	1	200
1	1	1	1	0	250
1	1	1	1	1	300

23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 11.15 PLL0C Register Contents (3/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>22 to 20</td><td>PLL0ADJ[2:0]</td><td>Frequency Modulation Range Setting
<table>
<tr><th>PLL0ADJ2</th><th>PLL0ADJ1</th><th>PLL0ADJ0</th><th>Frequency Modulation Range</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1%</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>2%</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>3%</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>4%</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>5%</td></tr>
</table></td></tr>
<tr><td>19 to 15</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>14</td><td>PLL0MD</td><td>Operating Mode Setting
0: PLL mode (fixed frequency)
1: SSCG mode (modulation frequency)</td></tr>
<tr><td>13</td><td>PLL0SMD</td><td>Modulation Mode Setting in SSCG Mode
0: Setting prohibited
1: Center spread modulation</td></tr>
<tr><td>12, 11</td><td>PLL0M[1:0]</td><td>Mr-Value Setting
<table>
<tr><th>PLL0M1</th><th>PLL0M0</th><th>Mr-Value</th><th>MainOSC Frequency f_X</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td>8 MHz ≤ f_X ≤ 24 MHz</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>16 MHz ≤ f_X ≤ 24 MHz</td></tr>
<tr><td>1</td><td>0</td><td>3</td><td>f_X = 24 MHz</td></tr>
<tr><td>1</td><td>1</td><td colspan="2">Setting prohibited</td></tr>
</table></td></tr>
<tr><td>10 to 8</td><td>PLL0P[2:0]</td><td>Pr-Value Setting
<table>
<tr><th>PLL0P2</th><th>PLL0P1</th><th>PLL0P0</th><th>Pr-Value</th><th>Output Frequency Range</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td colspan="2">Setting prohibited</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td colspan="2">Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>4</td><td>80 MHz to 120 MHz</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>8</td><td>40 MHz to 80 MHz</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>16</td><td>25 MHz to 40 MHz</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td colspan="2">Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td colspan="2">Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td colspan="2">Setting prohibited</td></tr>
</table></td></tr>
<tr><td>7</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 11.15 PLL0C Register Contents (4/4)

Bit Position	Bit Name	Function
6 to 0	PLL0N[6:0]	Nr-Value Setting The Nr-value is calculated as: Nr-value = PLL0N[6:0] + 1

PLL0 N6	PLL0 N5	PLL0 N4	PLL0 N3	PLL0 N2	PLL0 N1	PLL0 N0	Nr-Value
0	0	0	1	0	1	0	12
0	0	0	1	1	0	0	13
:							:
1	0	0	1	1	1	0	79
1	0	0	1	1	1	1	80
Other than above							Setting prohibited

CAUTION

Set this register when PLL0 is stopped.

Table 11.16 PLL0 Output Table

MainOSC = 8 MHz (1/2)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency (MHz)
1	16	0	1	1	0	0	0	1	50	25.00
1	16	0	1	1	0	0	1	0	51	25.50
1	16	0	1	1	0	0	1	1	52	26.00
1	16	0	1	1	0	1	0	0	53	26.50
1	16	0	1	1	0	1	0	1	54	27.00
1	16	0	1	1	0	1	1	0	55	27.50
1	16	0	1	1	0	1	1	1	56	28.00
1	16	0	1	1	1	0	0	0	57	28.50
1	16	0	1	1	1	0	0	1	58	29.00
1	16	0	1	1	1	0	1	0	59	29.50
1	16	0	1	1	1	0	1	1	60	30.00
1	8	0	1	0	0	1	1	1	40	40.00
1	8	0	1	0	1	0	0	0	41	41.00
1	8	0	1	0	1	0	0	1	42	42.00
1	8	0	1	0	1	0	1	0	43	43.00
1	8	0	1	0	1	0	1	1	44	44.00
1	8	0	1	0	1	1	0	0	45	45.00
1	8	0	1	0	1	1	0	1	46	46.00
1	8	0	1	0	1	1	1	0	47	47.00
1	8	0	1	0	1	1	1	1	48	48.00
1	8	0	1	1	0	0	0	0	49	49.00
1	8	0	1	1	0	0	0	1	50	50.00
1	8	0	1	1	0	0	1	0	51	51.00
1	8	0	1	1	0	0	1	1	52	52.00
1	8	0	1	1	0	1	0	0	53	53.00

MainOSC = 8 MHz (2/2)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency (MHz)
1	8	0	1	1	0	1	0	1	54	54.00
1	8	0	1	1	0	1	1	0	55	55.00
1	8	0	1	1	0	1	1	1	56	56.00
1	8	0	1	1	1	0	0	0	57	57.00
1	8	0	1	1	1	0	0	1	58	58.00
1	8	0	1	1	1	0	1	0	59	59.00
1	8	0	1	1	1	0	1	1	60	60.00
1	4	0	1	0	0	1	1	1	40	80.00
1	4	0	1	0	1	0	0	0	41	82.00
1	4	0	1	0	1	0	0	1	42	84.00
1	4	0	1	0	1	0	1	0	43	86.00
1	4	0	1	0	1	0	1	1	44	88.00
1	4	0	1	0	1	1	0	0	45	90.00
1	4	0	1	0	1	1	0	1	46	92.00
1	4	0	1	0	1	1	1	0	47	94.00
1	4	0	1	0	1	1	1	1	48	96.00
1	4	0	1	1	0	0	0	0	49	98.00
1	4	0	1	1	0	0	0	1	50	100.00
1	4	0	1	1	0	0	1	0	51	102.00
1	4	0	1	1	0	0	1	1	52	104.00
1	4	0	1	1	0	1	0	0	53	106.00
1	4	0	1	1	0	1	0	1	54	108.00
1	4	0	1	1	0	1	1	0	55	110.00
1	4	0	1	1	0	1	1	1	56	112.00
1	4	0	1	1	1	0	0	0	57	114.00
1	4	0	1	1	1	0	0	1	58	116.00
1	4	0	1	1	1	0	1	0	59	118.00
1	4	0	1	1	1	0	1	1	60	120.00

MainOSC = 12 MHz (1/2)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency(MHz)
1	16	0	1	0	0	0	0	1	34	25.50
1	16	0	1	0	0	0	1	0	35	26.25
1	16	0	1	0	0	0	1	1	36	27.00
1	16	0	1	0	0	1	0	0	37	27.75
1	16	0	1	0	0	1	0	1	38	28.50
1	16	0	1	0	0	1	1	0	39	29.25
1	16	0	1	0	0	1	1	1	40	30.00
1	8	0	0	1	1	0	1	0	27	40.50
1	8	0	0	1	1	0	1	1	28	42.00
1	8	0	0	1	1	1	0	0	29	43.50
1	8	0	0	1	1	1	0	1	30	45.00

MainOSC = 12 MHz (2/2)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency(MHz)
1	8	0	0	1	1	1	1	0	31	46.50
1	8	0	0	1	1	1	1	1	32	48.00
1	8	0	1	0	0	0	0	0	33	49.50
1	8	0	1	0	0	0	0	1	34	51.00
1	8	0	1	0	0	0	1	0	35	52.50
1	8	0	1	0	0	0	1	1	36	54.00
1	8	0	1	0	0	1	0	0	37	55.50
1	8	0	1	0	0	1	0	1	38	57.00
1	8	0	1	0	0	1	1	0	39	58.50
1	8	0	1	0	0	1	1	1	40	60.00
1	4	0	0	1	1	0	1	0	27	81.00
1	4	0	0	1	1	0	1	1	28	84.00
1	4	0	0	1	1	1	0	0	29	87.00
1	4	0	0	1	1	1	0	1	30	90.00
1	4	0	0	1	1	1	1	0	31	93.00
1	4	0	0	1	1	1	1	1	32	96.00
1	4	0	1	0	0	0	0	0	33	99.00
1	4	0	1	0	0	0	0	1	34	102.00
1	4	0	1	0	0	0	1	0	35	105.00
1	4	0	1	0	0	0	1	1	36	108.00
1	4	0	1	0	0	1	0	0	37	111.00
1	4	0	1	0	0	1	0	1	38	114.00
1	4	0	1	0	0	1	1	0	39	117.00
1	4	0	1	0	0	1	1	1	40	120.00

MainOSC = 16 MHz (1/3)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency (MHz)
1	16	0	0	1	1	0	0	0	25	25.00
1	16	0	0	1	1	0	0	1	26	26.00
1	16	0	0	1	1	0	1	0	27	27.00
1	16	0	0	1	1	0	1	1	28	28.00
1	16	0	0	1	1	1	0	0	29	29.00
1	16	0	0	1	1	1	0	1	30	30.00
1	8	0	0	1	0	0	1	1	20	40.00
1	8	0	0	1	0	1	0	0	21	42.00
1	8	0	0	1	0	1	0	1	22	44.00
1	8	0	0	1	0	1	1	0	23	46.00
1	8	0	0	1	0	1	1	1	24	48.00
1	8	0	0	1	1	0	0	0	25	50.00
1	8	0	0	1	1	0	0	1	26	52.00
1	8	0	0	1	1	0	1	0	27	54.00
1	8	0	0	1	1	0	1	1	28	56.00

MainOSC = 16 MHz (2/3)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency (MHz)
1	8	0	0	1	1	1	0	0	29	58.00
1	8	0	0	1	1	1	0	1	30	60.00
1	4	0	0	1	0	0	1	1	20	80.00
1	4	0	0	1	0	1	0	0	21	84.00
1	4	0	0	1	0	1	0	1	22	88.00
1	4	0	0	1	0	1	1	0	23	92.00
1	4	0	0	1	0	1	1	1	24	96.00
1	4	0	0	1	1	0	0	0	25	100.00
1	4	0	0	1	1	0	0	1	26	104.00
1	4	0	0	1	1	0	1	0	27	108.00
1	4	0	0	1	1	0	1	1	28	112.00
1	4	0	0	1	1	1	0	0	29	116.00
1	4	0	0	1	1	1	0	1	30	120.00
2	16	0	1	1	0	0	0	1	50	25.00
2	16	0	1	1	0	0	1	0	51	25.50
2	16	0	1	1	0	0	1	1	52	26.00
2	16	0	1	1	0	1	0	0	53	26.50
2	16	0	1	1	0	1	0	1	54	27.00
2	16	0	1	1	0	1	1	0	55	27.50
2	16	0	1	1	0	1	1	1	56	28.00
2	16	0	1	1	1	0	0	0	57	28.50
2	16	0	1	1	1	0	0	1	58	29.00
2	16	0	1	1	1	0	1	0	59	29.50
2	16	0	1	1	1	0	1	1	60	30.00
2	8	0	1	0	0	1	1	1	40	40.00
2	8	0	1	0	1	0	0	0	41	41.00
2	8	0	1	0	1	0	0	1	42	42.00
2	8	0	1	0	1	0	1	0	43	43.00
2	8	0	1	0	1	0	1	1	44	44.00
2	8	0	1	0	1	1	0	0	45	45.00
2	8	0	1	0	1	1	0	1	46	46.00
2	8	0	1	0	1	1	1	0	47	47.00
2	8	0	1	0	1	1	1	1	48	48.00
2	8	0	1	1	0	0	0	0	49	49.00
2	8	0	1	1	0	0	0	1	50	50.00
2	8	0	1	1	0	0	1	0	51	51.00
2	8	0	1	1	0	0	1	1	52	52.00
2	8	0	1	1	0	1	0	0	53	53.00
2	8	0	1	1	0	1	0	1	54	54.00
2	8	0	1	1	0	1	1	0	55	55.00
2	8	0	1	1	0	1	1	1	56	56.00
2	8	0	1	1	1	0	0	0	57	57.00
2	8	0	1	1	1	0	0	1	58	58.00

MainOSC = 16 MHz (3/3)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency (MHz)
2	8	0	1	1	1	0	1	0	59	59.00
2	8	0	1	1	1	0	1	1	60	60.00
2	4	0	1	0	0	1	1	1	40	80.00
2	4	0	1	0	1	0	0	0	41	82.00
2	4	0	1	0	1	0	0	1	42	84.00
2	4	0	1	0	1	0	1	0	43	86.00
2	4	0	1	0	1	0	1	1	44	88.00
2	4	0	1	0	1	1	0	0	45	90.00
2	4	0	1	0	1	1	0	1	46	92.00
2	4	0	1	0	1	1	1	0	47	94.00
2	4	0	1	0	1	1	1	1	48	96.00
2	4	0	1	1	0	0	0	0	49	98.00
2	4	0	1	1	0	0	0	1	50	100.00
2	4	0	1	1	0	0	1	0	51	102.00
2	4	0	1	1	0	0	1	1	52	104.00
2	4	0	1	1	0	1	0	0	53	106.00
2	4	0	1	1	0	1	0	1	54	108.00
2	4	0	1	1	0	1	1	0	55	110.00
2	4	0	1	1	0	1	1	1	56	112.00
2	4	0	1	1	1	0	0	0	57	114.00
2	4	0	1	1	1	0	0	1	58	116.00
2	4	0	1	1	1	0	1	0	59	118.00
2	4	0	1	1	1	0	1	1	60	120.00

MainOSC = 20 MHz (1/2)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency (MHz)
1	16	0	0	1	0	1	1	1	24	30.00
1	8	0	0	0	1	1	1	1	16	40.00
1	8	0	0	1	0	0	0	0	17	42.50
1	8	0	0	1	0	0	0	1	18	45.00
1	8	0	0	1	0	0	1	0	19	47.50
1	8	0	0	1	0	0	1	1	20	50.00
1	8	0	0	1	0	1	0	0	21	52.50
1	8	0	0	1	0	1	0	1	22	55.00
1	8	0	0	1	0	1	1	0	23	57.50
1	8	0	0	1	0	1	1	1	24	60.00
1	4	0	0	0	1	1	1	1	16	80.00
1	4	0	0	1	0	0	0	0	17	85.00
1	4	0	0	1	0	0	0	1	18	90.00
1	4	0	0	1	0	0	1	0	19	95.00
1	4	0	0	1	0	0	1	1	20	100.00

MainOSC = 20 MHz (2/2)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency (MHz)
1	4	0	0	1	0	1	0	0	21	105.00
1	4	0	0	1	0	1	0	1	22	110.00
1	4	0	0	1	0	1	1	0	23	115.00
1	4	0	0	1	0	1	1	1	24	120.00
2	16	0	1	0	1	1	1	1	48	30.00
2	8	0	0	1	1	1	1	1	32	40.00
2	8	0	1	0	0	0	0	0	33	41.25
2	8	0	1	0	0	0	0	1	34	42.50
2	8	0	1	0	0	0	1	0	35	43.75
2	8	0	1	0	0	0	1	1	36	45.00
2	8	0	1	0	0	1	0	0	37	46.25
2	8	0	1	0	0	1	0	1	38	47.50
2	8	0	1	0	0	1	1	0	39	48.75
2	8	0	1	0	0	1	1	1	40	50.00
2	8	0	1	0	1	0	0	0	41	51.25
2	8	0	1	0	1	0	0	1	42	52.50
2	8	0	1	0	1	0	1	0	43	53.75
2	8	0	1	0	1	0	1	1	44	55.00
2	8	0	1	0	1	1	0	0	45	56.25
2	8	0	1	0	1	1	0	1	46	57.50
2	8	0	1	0	1	1	1	0	47	58.75
2	8	0	1	0	1	1	1	1	48	60.00
2	4	0	0	1	1	1	1	1	32	80.00
2	4	0	1	0	0	0	0	0	33	82.50
2	4	0	1	0	0	0	0	1	34	85.00
2	4	0	1	0	0	0	1	0	35	87.50
2	4	0	1	0	0	0	1	1	36	90.00
2	4	0	1	0	0	1	0	0	37	92.50
2	4	0	1	0	0	1	0	1	38	95.00
2	4	0	1	0	0	1	1	0	39	97.50
2	4	0	1	0	0	1	1	1	40	100.00
2	4	0	1	0	1	0	0	0	41	102.50
2	4	0	1	0	1	0	0	1	42	105.00
2	4	0	1	0	1	0	1	0	43	107.50
2	4	0	1	0	1	0	1	1	44	110.00
2	4	0	1	0	1	1	0	0	45	112.50
2	4	0	1	0	1	1	0	1	46	115.00
2	4	0	1	0	1	1	1	0	47	117.50
2	4	0	1	0	1	1	1	1	48	120.00

MainOSC = 24 MHz (1/3)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency (MHz)
1	8	0	0	0	1	1	0	1	14	42.00
1	8	0	0	0	1	1	1	0	15	45.00
1	8	0	0	0	1	1	1	1	16	48.00
1	8	0	0	1	0	0	0	0	17	51.00
1	8	0	0	1	0	0	0	1	18	54.00
1	8	0	0	1	0	0	1	0	19	57.00
1	8	0	0	1	0	0	1	1	20	60.00
1	4	0	0	0	1	1	0	1	14	84.00
1	4	0	0	0	1	1	1	0	15	90.00
1	4	0	0	0	1	1	1	1	16	96.00
1	4	0	0	1	0	0	0	0	17	102.00
1	4	0	0	1	0	0	0	1	18	108.00
1	4	0	0	1	0	0	1	0	19	114.00
1	4	0	0	1	0	0	1	1	20	120.00
2	8	0	0	1	1	0	1	0	27	40.50
2	8	0	0	1	1	0	1	1	28	42.00
2	8	0	0	1	1	1	0	0	29	43.50
2	8	0	0	1	1	1	0	1	30	45.00
2	8	0	0	1	1	1	1	0	31	46.50
2	8	0	0	1	1	1	1	1	32	48.00
2	8	0	1	0	0	0	0	0	33	49.50
2	8	0	1	0	0	0	0	1	34	51.00
2	8	0	1	0	0	0	1	0	35	52.50
2	8	0	1	0	0	0	1	1	36	54.00
2	8	0	1	0	0	1	0	0	37	55.50
2	8	0	1	0	0	1	0	1	38	57.00
2	8	0	1	0	0	1	1	0	39	58.50
2	8	0	1	0	0	1	1	1	40	60.00
2	4	0	0	1	1	0	1	0	27	81.00
2	4	0	0	1	1	0	1	1	28	84.00
2	4	0	0	1	1	1	0	0	29	87.00
2	4	0	0	1	1	1	0	1	30	90.00
2	4	0	0	1	1	1	1	0	31	93.00
2	4	0	0	1	1	1	1	1	32	96.00
2	4	0	1	0	0	0	0	0	33	99.00
2	4	0	1	0	0	0	0	1	34	102.00
2	4	0	1	0	0	0	1	0	35	105.00
2	4	0	1	0	0	0	1	1	36	108.00
2	4	0	1	0	0	1	0	0	37	111.00
2	4	0	1	0	0	1	0	1	38	114.00
2	4	0	1	0	0	1	1	0	39	117.00
2	4	0	1	0	0	1	1	1	40	120.00

MainOSC = 24 MHz (2/3)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency (MHz)
3	16	0	1	1	0	0	0	1	50	25.00
3	16	0	1	1	0	0	1	0	51	25.50
3	16	0	1	1	0	0	1	1	52	26.00
3	16	0	1	1	0	1	0	0	53	26.50
3	16	0	1	1	0	1	0	1	54	27.00
3	16	0	1	1	0	1	1	0	55	27.50
3	16	0	1	1	0	1	1	1	56	28.00
3	16	0	1	1	1	0	0	0	57	28.50
3	16	0	1	1	1	0	0	1	58	29.00
3	16	0	1	1	1	0	1	0	59	29.50
3	16	0	1	1	1	0	1	1	60	30.00
3	8	0	1	0	0	1	1	1	40	40.00
3	8	0	1	0	1	0	0	0	41	41.00
3	8	0	1	0	1	0	0	1	42	42.00
3	8	0	1	0	1	0	1	0	43	43.00
3	8	0	1	0	1	0	1	1	44	44.00
3	8	0	1	0	1	1	0	0	45	45.00
3	8	0	1	0	1	1	0	1	46	46.00
3	8	0	1	0	1	1	1	0	47	47.00
3	8	0	1	0	1	1	1	1	48	48.00
3	8	0	1	1	0	0	0	0	49	49.00
3	8	0	1	1	0	0	0	1	50	50.00
3	8	0	1	1	0	0	1	0	51	51.00
3	8	0	1	1	0	0	1	1	52	52.00
3	8	0	1	1	0	1	0	0	53	53.00
3	8	0	1	1	0	1	0	1	54	54.00
3	8	0	1	1	0	1	1	0	55	55.00
3	8	0	1	1	0	1	1	1	56	56.00
3	8	0	1	1	1	0	0	0	57	57.00
3	8	0	1	1	1	0	0	1	58	58.00
3	8	0	1	1	1	0	1	0	59	59.00
3	8	0	1	1	1	0	1	1	60	60.00
3	4	0	1	0	0	1	1	1	40	80.00
3	4	0	1	0	1	0	0	0	41	82.00
3	4	0	1	0	1	0	0	1	42	84.00
3	4	0	1	0	1	0	1	0	43	86.00
3	4	0	1	0	1	0	1	1	44	88.00
3	4	0	1	0	1	1	0	0	45	90.00
3	4	0	1	0	1	1	0	1	46	92.00
3	4	0	1	0	1	1	1	0	47	94.00
3	4	0	1	0	1	1	1	1	48	96.00
3	4	0	1	1	0	0	0	0	49	98.00
3	4	0	1	1	0	0	0	1	50	100.00

MainOSC = 24 MHz (3/3)

Mr	Pr	NI6	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{CPLLCLK}$ Frequency (MHz)
3	4	0	1	1	0	0	1	0	51	102.00
3	4	0	1	1	0	0	1	1	52	104.00
3	4	0	1	1	0	1	0	0	53	106.00
3	4	0	1	1	0	1	0	1	54	108.00
3	4	0	1	1	0	1	1	0	55	110.00
3	4	0	1	1	0	1	1	1	56	112.00
3	4	0	1	1	1	0	0	0	57	114.00
3	4	0	1	1	1	0	0	1	58	116.00
3	4	0	1	1	1	0	1	0	59	118.00
3	4	0	1	1	1	0	1	1	60	120.00

11.4.2.15 PLL0ST — PLL0 Stabilization Time Register

This register specifies stabilization time of the PLL0 and is initialized by the ISORES signal.

Access: This register can be read or written in 32-bit units.

Address: FFF8 900C$_H$

Value after reset: 0000 1B80$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PLL0CLKST[12:0]												
Value after reset	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.17 PLL0ST Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is read. When writing, write the value after reset.
12 to 0	PLL0CLKST [12:0]	PLL0 stabilization time setting The PLL0 stabilization counter operates using the HS IntOSC. Set the following value based on the PLL0 operating mode settings.

Mode	PLL0CLKST[12:0]
PLL mode	0AA0$_H$
SSCG mode	1B80$_H$

NOTE

For details of the PLL0 stabilization time, see the Data Sheet.

CAUTION

Set this register when PLL0 is stopped.

11.4.2.16 PLL1E — PLL1 Enable Register

This register is used to start and stop PLL1.

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by the ISORES signal.

Access: This register can be read or written in 32-bit units.

Address: FFF8 9100_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1DISTRG	PLL1ENTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.18 PLL1E Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	PLL1DISTRG	PLL1 Disable Trigger*1 0: No function 1: Stops PLL1
0	PLL1ENTRG	PLL1 Enable Trigger*2 0: No function 1: Starts PLL1

Note 1. <Recommended procedure for stopping PLL1 using PLL1E.PLL1DISTRG>
1. Check that there is no clock domain for which PLL1 is selected.
If PLL1 is selected for a clock domain, disable the setting or select a clock source other than PLL1.
2. Stop the PLL1 (PLL1E.PLL1DISTRG = 1).
3. Confirm that the PLL1 has been stopped (PLL1S.PLL1CLKACT = 0).

Note 2. Before starting PLL1 using PLL1ENTRG, confirm that MainOSC is operating.

11.4.2.17 PLL1S — PLL1 Status Register

This register provides active status information about the PLL1.

This register is initialized by the ISORES signal.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 9104_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1*1	0*1
	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1CLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 11.19 PLL1S Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	PLL1CLKACT	PLL1 Active Status 0: PLL1 is inactive 1: PLL1 is active
1, 0	Reserved	When read, an undefined value is returned.

11.4.2.18 PLL1C — PLL1 Control Register

This register is used to set the PLL1 output clock frequencies $f_{PPLLCLK}$, shown in **Section 11.3.5.2, PLL1 Parameters**.

This register can only be written, if the PLL1 is disabled.

This register is initialized by the ISORES signal.

Access: This register can be read or written in 32-bit units.

Address: FFF8 9108$_H$

Value after reset: 0000 133B$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OUTBSEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	M[1:0]		PA[2:0]			—	—	N[5:0]					
Value after reset	0	0	0	1	0	0	1	1	0	0	1	1	1	0	1	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.20 PLL1C Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	OUTBSEL	$f_{PPLLCLK}$ selection bit 0: Select the clock dividing f_{VCOOUT} by PA[2:0] divider 1: Select the clock dividing f_{VCOOUT} by 5
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12, 11	M[1:0]	
10 to 8	PA[2:0]	P Divider Selection
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

M1	M0	Mr-Value	MainOSC Frequency f_X
0	0	1	8 MHz ≤ f_X ≤ 24 MHz
0	1	2	16 MHz ≤ f_X ≤ 24 MHz
1	0	3	f_X = 24 MHz
1	1	Setting prohibited	

PA[2:0]	Par-Value	PLL Output Frequency Range
000_B	Setting prohibited	—
001_B	Setting prohibited	—
010_B	4	60 MHz to 80 MHz
011_B	6	40 MHz to 80 MHz
100_B	8	30 MHz to 60 MHz
101_B	16	25 MHz to 30 MHz
110_B	Setting prohibited	—
111_B	Setting prohibited	—

Table 11.20 PLL1C Register Contents (2/2)

Bit Position	Bit Name	Function
5 to 0	N[5:0]	Division ratio Nr is set. For N[5:0] example settings, see **Table 11.21, PLL1 Output Table**.

CAUTION

Set this register when PLL1 is stopped.

Table 11.21 PLL1 Output Table

MainOSC = 8 MHz (1/2)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
0	1	16	1	1	0	0	0	1	50	25.00
0	1	16	1	1	0	0	1	0	51	25.50
0	1	16	1	1	0	0	1	1	52	26.00
0	1	16	1	1	0	1	0	0	53	26.50
0	1	16	1	1	0	1	0	1	54	27.00
0	1	16	1	1	0	1	1	0	55	27.50
0	1	16	1	1	0	1	1	1	56	28.00
0	1	16	1	1	1	0	0	0	57	28.50
0	1	16	1	1	1	0	0	1	58	29.00
0	1	16	1	1	1	0	1	0	59	29.50
0	1	16	1	1	1	0	1	1	60	30.00
0	1	8	0	1	1	1	0	1	30	30.00
0	1	8	0	1	1	1	1	0	31	31.00
0	1	8	0	1	1	1	1	1	32	32.00
0	1	8	1	0	0	0	0	0	33	33.00
0	1	8	1	0	0	0	0	1	34	34.00
0	1	8	1	0	0	0	1	0	35	35.00
0	1	8	1	0	0	0	1	1	36	36.00
0	1	8	1	0	0	1	0	0	37	37.00
0	1	8	1	0	0	1	0	1	38	38.00
0	1	8	1	0	0	1	1	0	39	39.00
0	1	8	1	0	0	1	1	1	40	40.00
0	1	8	1	0	1	0	0	0	41	41.00
0	1	8	1	0	1	0	0	1	42	42.00
0	1	8	1	0	1	0	1	0	43	43.00
0	1	8	1	0	1	0	1	1	44	44.00
0	1	8	1	0	1	1	0	0	45	45.00
0	1	8	1	0	1	1	0	1	46	46.00
0	1	8	1	0	1	1	1	0	47	47.00
0	1	8	1	0	1	1	1	1	48	48.00
0	1	8	1	1	0	0	0	0	49	49.00
0	1	8	1	1	0	0	0	1	50	50.00
0	1	8	1	1	0	0	1	0	51	51.00
0	1	8	1	1	0	0	1	1	52	52.00

MainOSC = 8 MHz (2/2)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
0	1	8	1	1	0	1	0	0	53	53.00
0	1	8	1	1	0	1	0	1	54	54.00
0	1	8	1	1	0	1	1	0	55	55.00
0	1	8	1	1	0	1	1	1	56	56.00
0	1	8	1	1	1	0	0	0	57	57.00
0	1	8	1	1	1	0	0	1	58	58.00
0	1	8	1	1	1	0	1	0	59	59.00
0	1	8	1	1	1	0	1	1	60	60.00
0	1	4	0	1	1	1	0	1	30	60.00
0	1	4	0	1	1	1	1	0	31	62.00
0	1	4	0	1	1	1	1	1	32	64.00
0	1	4	1	0	0	0	0	0	33	66.00
0	1	4	1	0	0	0	0	1	34	68.00
0	1	4	1	0	0	0	1	0	35	70.00
0	1	4	1	0	0	0	1	1	36	72.00
0	1	4	1	0	0	1	0	0	37	74.00
0	1	4	1	0	0	1	0	1	38	76.00
0	1	4	1	0	0	1	1	0	39	78.00
0	1	4	1	0	0	1	1	1	40	80.00
1	1	*1	0	1	1	1	0	1	30	48.00
1	1	*1	0	1	1	1	1	0	31	49.60
1	1	*1	0	1	1	1	1	1	32	51.20
1	1	*1	1	0	0	0	0	0	33	52.80
1	1	*1	1	0	0	0	0	1	34	54.40
1	1	*1	1	0	0	0	1	0	35	56.00
1	1	*1	1	0	0	0	1	1	36	57.60
1	1	*1	1	0	0	1	0	0	37	59.20
1	1	*1	1	0	0	1	0	1	38	60.80
1	1	*1	1	0	0	1	1	0	39	62.40
1	1	*1	1	0	0	1	1	1	40	64.00
1	1	*1	1	0	1	0	0	0	41	65.60
1	1	*1	1	0	1	0	0	1	42	67.20
1	1	*1	1	0	1	0	1	0	43	68.80
1	1	*1	1	0	1	0	1	1	44	70.40
1	1	*1	1	0	1	1	0	0	45	72.00
1	1	*1	1	0	1	1	0	1	46	73.60
1	1	*1	1	0	1	1	1	0	47	75.20
1	1	*1	1	0	1	1	1	1	48	76.80
1	1	*1	1	1	0	0	0	0	49	78.40
1	1	*1	1	1	0	0	0	1	50	80.00

Note 1. A clock of frequency output VCO (f_{VCOOUT}) divided by 5 is selected regardless of the value of Par.

MainOSC = 12 MHz (1/2)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
0	1	16	1	0	0	0	0	1	34	25.50
0	1	16	1	0	0	0	1	0	35	26.25
0	1	16	1	0	0	0	1	1	36	27.00
0	1	16	1	0	0	1	0	0	37	27.75
0	1	16	1	0	0	1	0	1	38	28.50
0	1	16	1	0	0	1	1	0	39	29.25
0	1	16	1	0	0	1	1	1	40	30.00
0	1	8	0	1	0	0	1	1	20	30.00
0	1	8	0	1	0	1	0	0	21	31.50
0	1	8	0	1	0	1	0	1	22	33.00
0	1	8	0	1	0	1	1	0	23	34.50
0	1	8	0	1	0	1	1	1	24	36.00
0	1	8	0	1	1	0	0	0	25	37.50
0	1	8	0	1	1	0	0	1	26	39.00
0	1	8	0	1	1	0	1	0	27	40.50
0	1	8	0	1	1	0	1	1	28	42.00
0	1	8	0	1	1	1	0	0	29	43.50
0	1	8	0	1	1	1	0	1	30	45.00
0	1	8	0	1	1	1	1	0	31	46.50
0	1	8	0	1	1	1	1	1	32	48.00
0	1	8	1	0	0	0	0	0	33	49.50
0	1	8	1	0	0	0	0	1	34	51.00
0	1	8	1	0	0	0	1	0	35	52.50
0	1	8	1	0	0	0	1	1	36	54.00
0	1	8	1	0	0	1	0	0	37	55.50
0	1	8	1	0	0	1	0	1	38	57.00
0	1	8	1	0	0	1	1	0	39	58.50
0	1	8	1	0	0	1	1	1	40	60.00
0	1	4	0	1	0	0	1	1	20	60.00
0	1	4	0	1	0	1	0	0	21	63.00
0	1	4	0	1	0	1	0	1	22	66.00
0	1	4	0	1	0	1	1	0	23	69.00
0	1	4	0	1	0	1	1	1	24	72.00
0	1	4	0	1	1	0	0	0	25	75.00
0	1	4	0	1	1	0	0	1	26	78.00
1	1	*1	0	1	0	0	1	1	20	48.00
1	1	*1	0	1	0	1	0	0	21	50.40
1	1	*1	0	1	0	1	0	1	22	52.80
1	1	*1	0	1	0	1	1	0	23	55.20
1	1	*1	0	1	0	1	1	1	24	57.60
1	1	*1	0	1	1	0	0	0	25	60.00
1	1	*1	0	1	1	0	0	1	26	62.40
1	1	*1	0	1	1	0	1	0	27	64.80

MainOSC = 12 MHz (2/2)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
1	1	*1	0	1	1	0	1	1	28	67.20
1	1	*1	0	1	1	1	0	0	29	69.60
1	1	*1	0	1	1	1	0	1	30	72.00
1	1	*1	0	1	1	1	1	0	31	74.40
1	1	*1	0	1	1	1	1	1	32	76.80
1	1	*1	1	0	0	0	0	0	33	79.20

Note 1. A clock of frequency output VCO (f_{VCOOUT}) divided by 5 is selected regardless of the value of Par.

MainOSC = 16 MHz (1/3)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
0	1	16	0	1	1	0	0	0	25	25.00
0	1	16	0	1	1	0	0	1	26	26.00
0	1	16	0	1	1	0	1	0	27	27.00
0	1	16	0	1	1	0	1	1	28	28.00
0	1	16	0	1	1	1	0	0	29	29.00
0	1	16	0	1	1	1	0	1	30	30.00
0	1	8	0	1	0	0	1	1	20	40.00
0	1	8	0	1	0	1	0	0	21	42.00
0	1	8	0	1	0	1	0	1	22	44.00
0	1	8	0	1	0	1	1	0	23	46.00
0	1	8	0	1	0	1	1	1	24	48.00
0	1	8	0	1	1	0	0	0	25	50.00
0	1	8	0	1	1	0	0	1	26	52.00
0	1	8	0	1	1	0	1	0	27	54.00
0	1	8	0	1	1	0	1	1	28	56.00
0	1	8	0	1	1	1	0	0	29	58.00
0	1	8	0	1	1	1	0	1	30	60.00
0	2	16	1	1	0	0	0	1	50	25.00
0	2	16	1	1	0	0	1	0	51	25.50
0	2	16	1	1	0	0	1	1	52	26.00
0	2	16	1	1	0	1	0	0	53	26.50
0	2	16	1	1	0	1	0	1	54	27.00
0	2	16	1	1	0	1	1	0	55	27.50
0	2	16	1	1	0	1	1	1	56	28.00
0	2	16	1	1	1	0	0	0	57	28.50
0	2	16	1	1	1	0	0	1	58	29.00
0	2	16	1	1	1	0	1	0	59	29.50
0	2	16	1	1	1	0	1	1	60	30.00
0	2	8	0	1	1	1	0	1	30	30.00
0	2	8	0	1	1	1	1	0	31	31.00
0	2	8	0	1	1	1	1	1	32	32.00
0	2	8	1	0	0	0	0	0	33	33.00

MainOSC = 16 MHz (2/3)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
0	2	8	1	0	0	0	0	1	34	34.00
0	2	8	1	0	0	0	1	0	35	35.00
0	2	8	1	0	0	0	1	1	36	36.00
0	2	8	1	0	0	1	0	0	37	37.00
0	2	8	1	0	0	1	0	1	38	38.00
0	2	8	1	0	0	1	1	0	39	39.00
0	2	8	1	0	0	1	1	1	40	40.00
0	2	8	1	0	1	0	0	0	41	41.00
0	2	8	1	0	1	0	0	1	42	42.00
0	2	8	1	0	1	0	1	0	43	43.00
0	2	8	1	0	1	0	1	1	44	44.00
0	2	8	1	0	1	1	0	0	45	45.00
0	2	8	1	0	1	1	0	1	46	46.00
0	2	8	1	0	1	1	1	0	47	47.00
0	2	8	1	0	1	1	1	1	48	48.00
0	2	8	1	1	0	0	0	0	49	49.00
0	2	8	1	1	0	0	0	1	50	50.00
0	2	8	1	1	0	0	1	0	51	51.00
0	2	8	1	1	0	0	1	1	52	52.00
0	2	8	1	1	0	1	0	0	53	53.00
0	2	8	1	1	0	1	0	1	54	54.00
0	2	8	1	1	0	1	1	0	55	55.00
0	2	8	1	1	0	1	1	1	56	56.00
0	2	8	1	1	1	0	0	0	57	57.00
0	2	8	1	1	1	0	0	1	58	58.00
0	2	8	1	1	1	0	1	0	59	59.00
0	2	8	1	1	1	0	1	1	60	60.00
0	2	4	0	1	1	1	0	1	30	60.00
0	2	4	0	1	1	1	1	0	31	62.00
0	2	4	0	1	1	1	1	1	32	64.00
0	2	4	1	0	0	0	0	0	33	66.00
0	2	4	1	0	0	0	0	1	34	68.00
0	2	4	1	0	0	0	1	0	35	70.00
0	2	4	1	0	0	0	1	1	36	72.00
0	2	4	1	0	0	1	0	0	37	74.00
0	2	4	1	0	0	1	0	1	38	76.00
0	2	4	1	0	0	1	1	0	39	78.00
0	2	4	1	0	0	1	1	1	40	80.00
1	1	*1	0	1	0	0	1	1	20	64.00
1	1	*1	0	1	0	1	0	0	21	67.20
1	1	*1	0	1	0	1	0	1	22	70.40
1	1	*1	0	1	0	1	1	0	23	73.60
1	1	*1	0	1	0	1	1	1	24	76.80

MainOSC = 16 MHz (3/3)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
1	1	*1	0	1	1	0	0	0	25	80.00
1	2	*1	0	1	1	1	0	1	30	48.00
1	2	*1	0	1	1	1	1	0	31	49.60
1	2	*1	0	1	1	1	1	1	32	51.20
1	2	*1	1	0	0	0	0	0	33	52.80
1	2	*1	1	0	0	0	0	1	34	54.40
1	2	*1	1	0	0	0	1	0	35	56.00
1	2	*1	1	0	0	0	1	1	36	57.60
1	2	*1	1	0	0	1	0	0	37	59.20
1	2	*1	1	0	0	1	0	1	38	60.80
1	2	*1	1	0	0	1	1	0	39	62.40
1	2	*1	1	0	0	1	1	1	40	64.00
1	2	*1	1	0	1	0	0	0	41	65.60
1	2	*1	1	0	1	0	0	1	42	67.20
1	2	*1	1	0	1	0	1	0	43	68.80
1	2	*1	1	0	1	0	1	1	44	70.40
1	2	*1	1	0	1	1	0	0	45	72.00
1	2	*1	1	0	1	1	0	1	46	73.60
1	2	*1	1	0	1	1	1	0	47	75.20
1	2	*1	1	0	1	1	1	1	48	76.80
1	2	*1	1	1	0	0	0	0	49	78.40
1	2	*1	1	1	0	0	0	1	50	80.00

Note 1. A clock of frequency output VCO (f_{VCOOUT}) divided by 5 is selected regardless of the value of Par.

MainOSC = 20 MHz (1/3)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
0	1	16	0	1	0	0	1	1	20	25.00
0	1	16	0	1	0	1	0	0	21	26.25
0	1	16	0	1	0	1	0	1	22	27.50
0	1	16	0	1	0	1	1	0	23	28.75
0	1	16	0	1	0	1	1	1	24	30.00
0	1	8	0	1	0	0	1	1	20	50.00
0	1	8	0	1	0	1	0	0	21	52.50
0	1	8	0	1	0	1	0	1	22	55.00
0	1	8	0	1	0	1	1	0	23	57.50
0	1	8	0	1	0	1	1	1	24	60.00
0	2	16	1	0	0	1	1	1	40	25.00
0	2	16	1	0	1	0	0	0	41	25.63
0	2	16	1	0	1	0	0	1	42	26.25
0	2	16	1	0	1	0	1	0	43	26.88
0	2	16	1	0	1	0	1	1	44	27.50
0	2	16	1	0	1	1	0	0	45	28.13

MainOSC = 20 MHz (2/3)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
0	2	16	1	0	1	1	0	1	46	28.75
0	2	16	1	0	1	1	1	0	47	29.38
0	2	16	1	0	1	1	1	1	48	30.00
0	2	8	0	1	0	1	1	1	24	30.00
0	2	8	0	1	1	0	0	0	25	31.25
0	2	8	0	1	1	0	0	1	26	32.50
0	2	8	0	1	1	0	1	0	27	33.75
0	2	8	0	1	1	0	1	1	28	35.00
0	2	8	0	1	1	1	0	0	29	36.25
0	2	8	0	1	1	1	0	1	30	37.50
0	2	8	0	1	1	1	1	0	31	38.75
0	2	8	0	1	1	1	1	1	32	40.00
0	2	8	1	0	0	0	0	0	33	41.25
0	2	8	1	0	0	0	0	1	34	42.50
0	2	8	1	0	0	0	1	0	35	43.75
0	2	8	1	0	0	0	1	1	36	45.00
0	2	8	1	0	0	1	0	0	37	46.25
0	2	8	1	0	0	1	0	1	38	47.50
0	2	8	1	0	0	1	1	0	39	48.75
0	2	8	1	0	0	1	1	1	40	50.00
0	2	8	1	0	1	0	0	0	41	51.25
0	2	8	1	0	1	0	0	1	42	52.50
0	2	8	1	0	1	0	1	0	43	53.75
0	2	8	1	0	1	0	1	1	44	55.00
0	2	8	1	0	1	1	0	0	45	56.25
0	2	8	1	0	1	1	0	1	46	57.50
0	2	8	1	0	1	1	1	0	47	58.75
0	2	8	1	0	1	1	1	1	48	60.00
0	2	4	0	1	0	1	1	1	24	60.00
0	2	4	0	1	1	0	0	0	25	62.50
0	2	4	0	1	1	0	0	1	26	65.00
0	2	4	0	1	1	0	1	0	27	67.50
0	2	4	0	1	1	0	1	1	28	70.00
0	2	4	0	1	1	1	0	0	29	72.50
0	2	4	0	1	1	1	0	1	30	75.00
0	2	4	0	1	1	1	1	0	31	77.50
0	2	4	0	1	1	1	1	1	32	80.00
1	1	*1	0	1	0	0	1	1	20	80.00
1	2	*1	0	1	0	1	1	1	24	48.00
1	2	*1	0	1	1	0	0	0	25	50.00
1	2	*1	0	1	1	0	0	1	26	52.00
1	2	*1	0	1	1	0	1	0	27	54.00
1	2	*1	0	1	1	0	1	1	28	56.00

MainOSC = 20 MHz (3/3)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
1	2	*1	0	1	1	1	0	0	29	58.00
1	2	*1	0	1	1	1	0	1	30	60.00
1	2	*1	0	1	1	1	1	0	31	62.00
1	2	*1	0	1	1	1	1	1	32	64.00
1	2	*1	1	0	0	0	0	0	33	66.00
1	2	*1	1	0	0	0	0	1	34	68.00
1	2	*1	1	0	0	0	1	0	35	70.00
1	2	*1	1	0	0	0	1	1	36	72.00
1	2	*1	1	0	0	1	0	0	37	74.00
1	2	*1	1	0	0	1	0	1	38	76.00
1	2	*1	1	0	0	1	1	0	39	78.00
1	2	*1	1	0	0	1	1	1	40	80.00

Note 1. A clock of frequency output VCO (f_{VCOOUT}) divided by 5 is selected regardless of the value of Par.

MainOSC = 24 MHz (1/3)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
0	1	16	0	1	0	0	1	1	20	30.00
0	1	8	0	1	0	0	1	1	20	60.00
0	2	16	1	0	0	1	1	1	40	30.00
0	2	8	0	1	0	0	1	1	20	30.00
0	2	8	0	1	0	1	0	0	21	31.50
0	2	8	0	1	0	1	0	1	22	33.00
0	2	8	0	1	0	1	1	0	23	34.50
0	2	8	0	1	0	1	1	1	24	36.00
0	2	8	0	1	1	0	0	0	25	37.50
0	2	8	0	1	1	0	0	1	26	39.00
0	2	8	0	1	1	0	1	0	27	40.50
0	2	8	0	1	1	0	1	1	28	42.00
0	2	8	0	1	1	1	0	0	29	43.50
0	2	8	0	1	1	1	0	1	30	45.00
0	2	8	0	1	1	1	1	0	31	46.50
0	2	8	0	1	1	1	1	1	32	48.00
0	2	8	1	0	0	0	0	0	33	49.50
0	2	8	1	0	0	0	0	1	34	51.00
0	2	8	1	0	0	0	1	0	35	52.50
0	2	8	1	0	0	0	1	1	36	54.00
0	2	8	1	0	0	1	0	0	37	55.50
0	2	8	1	0	0	1	0	1	38	57.00
0	2	8	1	0	0	1	1	0	39	58.50
0	2	8	1	0	0	1	1	1	40	60.00
0	2	4	0	1	0	0	1	1	20	60.00
0	2	4	0	1	0	1	0	0	21	63.00

MainOSC = 24 MHz (2/3)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
0	2	4	0	1	0	1	0	1	22	66.00
0	2	4	0	1	0	1	1	0	23	69.00
0	2	4	0	1	0	1	1	1	24	72.00
0	2	4	0	1	1	0	0	0	25	75.00
0	2	4	0	1	1	0	0	1	26	78.00
0	3	16	1	1	1	0	1	1	60	30.00
0	3	8	0	1	1	1	0	1	30	30.00
0	3	8	0	1	1	1	1	0	31	31.00
0	3	8	0	1	1	1	1	1	32	32.00
0	3	8	1	0	0	0	0	0	33	33.00
0	3	8	1	0	0	0	0	1	34	34.00
0	3	8	1	0	0	0	1	0	35	35.00
0	3	8	1	0	0	0	1	1	36	36.00
0	3	8	1	0	0	1	0	0	37	37.00
0	3	8	1	0	0	1	0	1	38	38.00
0	3	8	1	0	0	1	1	0	39	39.00
0	3	8	1	0	0	1	1	1	40	40.00
0	3	8	1	0	1	0	0	0	41	41.00
0	3	8	1	0	1	0	0	1	42	42.00
0	3	8	1	0	1	0	1	0	43	43.00
0	3	8	1	0	1	0	1	1	44	44.00
0	3	8	1	0	1	1	0	0	45	45.00
0	3	8	1	0	1	1	0	1	46	46.00
0	3	8	1	0	1	1	1	0	47	47.00
0	3	8	1	0	1	1	1	1	48	48.00
0	3	8	1	1	0	0	0	0	49	49.00
0	3	8	1	1	0	0	0	1	50	50.00
0	3	8	1	1	0	0	1	0	51	51.00
0	3	8	1	1	0	0	1	1	52	52.00
0	3	8	1	1	0	1	0	0	53	53.00
0	3	8	1	1	0	1	0	1	54	54.00
0	3	8	1	1	0	1	1	0	55	55.00
0	3	8	1	1	0	1	1	1	56	56.00
0	3	8	1	1	1	0	0	0	57	57.00
0	3	8	1	1	1	0	0	1	58	58.00
0	3	8	1	1	1	0	1	0	59	59.00
0	3	8	1	1	1	0	1	1	60	60.00
0	3	4	0	1	1	1	0	1	30	60.00
0	3	4	0	1	1	1	1	0	31	62.00
0	3	4	0	1	1	1	1	1	32	64.00
0	3	4	1	0	0	0	0	0	33	66.00
0	3	4	1	0	0	0	0	1	34	68.00
0	3	4	1	0	0	0	1	0	35	70.00

MainOSC = 24 MHz (3/3)

OUTBSEL	Mr	Par	NI5	NI4	NI3	NI2	NI1	NI0	Nr	$f_{PPLLCLK}$ Frequency (MHz)
0	3	4	1	0	0	0	1	1	36	72.00
0	3	4	1	0	0	1	0	0	37	74.00
0	3	4	1	0	0	1	0	1	38	76.00
0	3	4	1	0	0	1	1	0	39	78.00
0	3	4	1	0	0	1	1	1	40	80.00
1	2	*1	0	1	0	0	1	1	20	48.00
1	2	*1	0	1	0	1	0	0	21	50.40
1	2	*1	0	1	0	1	0	1	22	52.80
1	2	*1	0	1	0	1	1	0	23	55.20
1	2	*1	0	1	0	1	1	1	24	57.60
1	2	*1	0	1	1	0	0	0	25	60.00
1	2	*1	0	1	1	0	0	1	26	62.40
1	2	*1	0	1	1	0	1	0	27	64.80
1	2	*1	0	1	1	0	1	1	28	67.20
1	2	*1	0	1	1	1	0	0	29	69.60
1	2	*1	0	1	1	1	0	1	30	72.00
1	2	*1	0	1	1	1	1	0	31	74.40
1	2	*1	0	1	1	1	1	1	32	76.80
1	2	*1	1	0	0	0	0	0	33	79.20
1	3	*1	0	1	1	1	0	1	30	48.00
1	3	*1	0	1	1	1	1	0	31	49.60
1	3	*1	0	1	1	1	1	1	32	51.20
1	3	*1	1	0	0	0	0	0	33	52.80
1	3	*1	1	0	0	0	0	1	34	54.40
1	3	*1	1	0	0	0	1	0	35	56.00
1	3	*1	1	0	0	0	1	1	36	57.60
1	3	*1	1	0	0	1	0	0	37	59.20
1	3	*1	1	0	0	1	0	1	38	60.80
1	3	*1	1	0	0	1	1	0	39	62.40
1	3	*1	1	0	0	1	1	1	40	64.00
1	3	*1	1	0	1	0	0	0	41	65.60
1	3	*1	1	0	1	0	0	1	42	67.20
1	3	*1	1	0	1	0	1	0	43	68.80
1	3	*1	1	0	1	0	1	1	44	70.40
1	3	*1	1	0	1	1	0	0	45	72.00
1	3	*1	1	0	1	1	0	1	46	73.60
1	3	*1	1	0	1	1	1	0	47	75.20
1	3	*1	1	0	1	1	1	1	48	76.80
1	3	*1	1	1	0	0	0	0	49	78.40
1	3	*1	1	1	0	0	0	1	50	80.00

Note 1. A clock of frequency output VCO (f_{VCOOUT}) divided by 5 is selected regardless of the value of Par.

11.4.3 Clock Selector Control Register

11.4.3.1 WDTA0 Clock Domain C_AWO_WDTA

(1) CKSC_AWDTAD_CTL — C_AWO_WDTA Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2000$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTADCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.22 CKSC_AWDTAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AWDTADCSID [1:0]	Clock Divider Setting for C_AWO_WDTA 00_B: Setting prohibited 01_B: LS IntOSC / 128 (default) 10_B: LS IntOSC / 1 11_B: Setting prohibited

CAUTION

Confirm that CKSC_AWDTAD_CTL is CKSC_AWDTAD_ACT before setting the CKSC_AWDTAD_CTL register.

(2) CKSC_AWDTAD_ACT — C_AWO_WDTA Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2008$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTADACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*1	0*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The value in the inactive state. The value becomes 01$_B$ after entering the active state.

Table 11.23 CKSC_AWDTAD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AWDTADACT [1:0]	Clock divider for currently active C_AWO_WDTA

(3) CKSC_AWDTAD_STPM — C_AWO_WDTA Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2018$_H$

Value after reset: 0000 0002$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWDTADSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 11.24 CKSC_AWDTAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AWDTAD STPMSK	0: Clock domain C_AWO_WDTA is stopped in stand-by mode. 1: Clock domain C_AWO_WDTA is not stopped in stand-by mode.

11.4.3.2 TAUJ Clock Domain C_AWO_TAUJ

(1) CKSC_ATAUJS_CTL — C_AWO_TAUJ Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2100_H

Value after reset: 0000 0001_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJSCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 11.25 CKSC_ATAUJS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ATAUJSCSID [2:0]	Source Clock Setting for C_AWO_TAUJ*1 000_B: Disabled 001_B: HS IntOSC (default) 010_B: MainOSC 011_B: LS IntOSC 100_B: PPLLCLK2 Other than above: Setting prohibited

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

(2) CKSC_ATAUJS_ACT — C_AWO_TAUJ Source Clock Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2108$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJSACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.26 CKSC_ATAUJS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ATAUJSACT [2:0]	Source clock for currently active C_AWO_TAUJ

(3) CKSC_ATAUJD_CTL — C_AWO_TAUJ Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2200$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJDCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 11.27 CKSC_ATAUJD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ATAUJDCSID [2:0]	Clock Divider Setting for C_AWO_TAUJ 000_B: Setting prohibited 001_B: CKSC_ATAUJS_CTL selection /1 (default) 010_B: CKSC_ATAUJS_CTL selection /2 011_B: CKSC_ATAUJS_CTL selection /4 100_B: CKSC_ATAUJS_CTL selection /8 Other than above: Setting prohibited

(4) CKSC_ATAUJD_ACT — C_AWO_TAUJ Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2208$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJDACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.28 CKSC_ATAUJD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ATAUJDACT [2:0]	Clock divider for currently active C_AWO_TAUJ

(5) CKSC_ATAUJD_STPM — C_AWO_TAUJ Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2218$_H$

Value after reset: 0000 0002$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ATAUJ DSTP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 11.29 CKSC_ATAUJD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ATAUJD STPMSK	0: Clock domain C_AWO_TAUJ is stopped in stand-by mode. Clock domain C_AWO_TAUJ is stopped by disabling the C_AWO_TAUJ source clock selection register (CKSC_ATAUJS_CTL = 0000 0000$_H$). 1: Clock domain C_AWO_TAUJ is not stopped in stand-by mode. Clock domain C_AWO_TAUJ is not stopped by disabling the C_AWO_TAUJ source clock selection register (CKSC_ATAUJS_CTL = 0000 0000$_H$).

11.4.3.3 RTCA Clock Domain C_AWO_RTCA

(1) CKSC_ARTCAS_CTL — C_AWO_RTCA Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by a power-up reset PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 2300$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCASCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.30 CKSC_ARTCAS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ARTCASCSID [1:0]	Source Clock Setting for C_AWO_RTCA 00_B: Disable (default) 01_B: SubOSC 10_B: MainOSC*1 11_B: LS IntOSC

Note 1. To avoid supplying a clock signal at a frequency of 4 MHz or higher to the C_AWO_RTCA clock domain due to the CKSC_ARTCAS_CTL register setting, check that CKSC_ARTCAD_ACT = 0000 0000$_H$ (disabled) when the setting of CKSC_ARTCAS_CTL is 10_B (MainOSC).

(2) CKSC_ARTCAS_ACT — C_AWO_RTCA Source Clock Active Register

This register is initialized by a power-up reset PURES.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2308$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCASACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.31 CKSC_ARTCAS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ARTCASACT [1:0]	Source clock for currently active C_AWO_RTCA

(3) CKSC_ARTCAD_CTL — C_AWO_RTCA Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by a power-up reset PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 2400$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCADCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 11.32 CKSC_ARTCAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	ARTCADCSID [2:0]	Clock Divider Setting for C_AWO_RTCA 000_B: Disabled (default) 001_B: CKSC_ARTCAS_CTL selection /1 010_B: CKSC_ARTCAS_CTL selection /2 011_B: CKSC_ARTCAS_CTL selection /4 100_B: CKSC_ARTCAS_CTL selection /8 Other than above: Setting prohibited

(4) CKSC_ARTCAD_ACT — C_AWO_RTCA Clock Divider Active Register

This register is initialized by a power-up reset PURES.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2408$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCADACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.33 CKSC_ARTCAD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	ARTCADACT [2:0]	Clock divider for currently active C_AWO_RTCA

(5) CKSC_ARTCAD_STPM — C_AWO_RTCA Stop Mask Register

This register is initialized by a power-up reset PURES.

Access: This register can be read or written in 32-bit units.

Address: FFF8 2418$_H$

Value after reset: 0000 0002$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARTCADSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 11.34 CKSC_ARTCAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ARTCAD STPMSK	0: Clock domain C_AWO_RTCA is stopped in stand-by mode. Clock domain C_AWO_RTCA is stopped by disabling the C_AWO_RTCA source clock selection register (CKSC_ARTCAS_CTL = 0000 0000$_H$). 1: Clock domain C_AWO_RTCA is not stopped in stand-by mode. Clock domain C_AWO_RTCA is not stopped by disabling the C_AWO_RTCA source clock selection register (CKSC_ARTCAS_CTL = 0000 0000$_H$).

11.4.3.4 ADCA0 Clock Domain C_AWO_ADCA

(1) CKSC_AADCAS_CTL — C_AWO_ADCA Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2500_H

Value after reset: 0000 0001_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCASCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.35 CKSC_AADCAS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AADCASCSID [1:0]	Source Clock Setting for C_AWO_ADCA*1 00_B: Disabled 01_B: HS IntOSC (default) 10_B: MainOSC 11_B: PPLLCLK2

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

(2) CKSC_AADCAS_ACT — C_AWO_ADCA Source Clock Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2508$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCASACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.36 CKSC_AADCAS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AADCASACT [1:0]	Source clock for currently active C_AWO_ADCA

(3) CKSC_AADCAD_CTL — C_AWO_ADCA Clock Divider Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2600$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCADCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.37 CKSC_AADCAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	AADCADCSID [1:0]	Clock Divider Setting for C_AWO_ADCA 00_B: Setting prohibited 01_B: CKSC_AADCAS_CTL selection /1 (default) 10_B: CKSC_AADCAS_CTL selection /2*1 11_B: Setting prohibited

Note 1. Make sure that the frequency of CKSC_AADCA is no less than 8 MHz after division by 2.

(4) CKSC_AADCAD_ACT — C_AWO_ADCA Clock Divider Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2608$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCADACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.38 CKSC_AADCAD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	AADCADACT [1:0]	Clock divider for currently active C_AWO_ADCA

(5) CKSC_AADCAD_STPM — C_AWO_ADCA Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2618$_H$

Value after reset: 0000 0002$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCA DSTP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 11.39 CKSC_AADCAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AADCAD STPMSK	0: Clock domain C_AWO_ADCA is stopped in stand-by mode. Clock domain C_AWO_ADCA is stopped by disabling the C_AWO_ADCA source clock selection register (CKSC_AADCAS_CTL = 0000 0000$_H$). 1: Clock domain C_AWO_ADCA is not stopped in stand-by mode. Clock domain C_AWO_ADCA is not stopped by disabling the C_AWO_ADCA source clock selection register (CKSC_AADCAS_CTL = 0000 0000$_H$).

11.4.3.5 FOUT Clock Domain C_AWO_FOUT

(1) CKSC_AFOUTS_CTL — C_AWO_FOUT Source Clock Selection Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2700_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUTSCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 11.40 CKSC_AFOUTS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	AFOUTSCSID [2:0]	Source Clock Setting for C_AWO_FOUT*1 000_B: Disabled (default) 001_B: MainOSC 010_B: HS IntOSC 011_B: LS IntOSC 100_B: SubOSC 101_B: CPLLCLK4*2 110_B: PPLLCLK4*2 111_B: Setting prohibited

Note 1. Before transitioning to stand-by mode, select a source clock other than CPLLCLK4 and PPLLCLK4. However, the frequency output function (FOUT) cannot be used in DeepSTOP mode.

Note 2. Confirm that PLL0S.PLL0CLKACT=PLL1S.PLL1CLKACT=1 (both PLL0 and PLL1 are operating) before selecting CPLLCLK4 or PPLLCLK4.

(2) CKSC_AFOUTS_ACT — C_AWO_FOUT Source Clock Active Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2708$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUTSACT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.41 CKSC_AFOUTS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	AFOUTSACT [2:0]	Source clock for currently active C_AWO_FOUT

(3) CKSC_AFOUTS_STPM — C_AWO_FOUT Stop Mask Register

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2718$_H$

Value after reset: 0000 0002$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AFOUTSSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 11.42 CKSC_AFOUTS_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	AFOUTS STPMSK	0: Clock domain C_AWO_FOUT is stopped in stand-by mode. 1: Clock domain C_AWO_FOUT is not stopped in stand-by mode.

11.4.3.6 CPU Clock Domain C_ISO_CPUCLK

(1) CKSC_CPUCLKS_CTL — C_ISO_CPUCLK Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A000$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.43 CKSC_CPUCLKS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	CPUCLKSCSID [1:0]	Source Clock Setting for C_ISO_CPUCLK 00_B: Setting prohibited 01_B: EMCLK (default) 10_B: MainOSC 11_B: CPLLCLK

CAUTION

The clock source selected for the C_ISO_CPUCLK clock domain should not be stopped by software.

(2) CKSC_CPUCLKS_ACT — C_ISO_CPUCLK Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A008$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.44 CKSC_CPUCLKS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ACPUCLKSACT [1:0]	Source clock for currently active C_ISO_CPUCLK

(3) CKSC_CPUCLKD_CTL — C_ISO_CPUCLK Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A100$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKDCSID[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 11.45 CKSC_CPUCLKD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	CPUCLKDCSID [2:0]	Clock Divider Setting for C_ISO_CPUCLK 000_B: Setting prohibited 001_B: CKSC_CPUCLKS_CTL selection /1 (Default) 010_B: CKSC_CPUCLKS_CTL selection /2 011_B: CKSC_CPUCLKS_CTL selection /4 100_B: CKSC_CPUCLKS_CTL selection /8 Other than above: Setting prohibited

(4) CKSC_CPUCLKD_ACT — C_ISO_CPUCLK Clock Divider Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A108$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKDACT [2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.46 CKSC_CPUCLKD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	CPUCLKDACT [2:0]	Clock divider for currently active C_ISO_CPUCLK

11.4.3.7 Peripheral Clock Domains C_ISO_PERI1 and C_ISO_PERI2

(1) CKSC_IPERI1S_CTL — C_ISO_PERI1 Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 $A200_H$

Value after reset: 0000 0001_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI1SCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.47 CKSC_IPERI1S_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IPERI1SCSID [1:0]	Source Clock Setting for C_ISO_PERI1 00_B: Disabled 01_B: CPUCLK2 (default) 10_B: PPLLCLK 11_B: Setting prohibited

(2) CKSC_IPERI1S_ACT — C_ISO_PERI1 Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A208$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI1SACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.48 CKSC_IPERI1S_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IPERI1SACT [1:0]	Source clock for currently active C_ISO_PERI1

(3) CKSC_IPERI2S_CTL — C_ISO_PERI2 Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A300$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI2SCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.49 CKSC_IPERI2S_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IPERI2SCSID [1:0]	Source Clock Setting for C_ISO_PERI2 00_B: Disabled 01_B: CPUCLK2 (default) 10_B: PPLLCLK2 11_B: Setting prohibited

(4) CKSC_IPERI2S_ACT — C_ISO_PERI2 Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A308$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPERI2SACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.50 CKSC_IPERI2S_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IPERI2SACT [1:0]	Source clock for currently active C_ISO_PERI2

11.4.3.8 RLIN Clock Domains C_ISO_LIN

(1) CKSC_ILINS_CTL — C_ISO_LIN Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A400$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINSCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.51 CKSC_ILINS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ILINSCSID[1:0]	Source Clock Setting for C_ISO_LIN*1 00_B: Disabled 01_B: CPUCLK2 (default) 10_B: MainOSC 11_B: PPLLCLK2

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK2.

(2) CKSC_ILINS_ACT — C_ISO_LIN Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A408$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINSACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.52 CKSC_ILINS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ILINSACT[1:0]	Source clock for currently active C_ISO_LIN

(3) CKSC_ILIND_CTL — C_ISO_LIN Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A800$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINDCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.53 CKSC_ILIND_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ILINDCSID[1:0]	Clock Divider Setting for C_ISO_LIN 00_B: Setting prohibited 01_B: CKSC_ILINS_CTL selection /1 (default) 10_B: CKSC_ILINS_CTL selection /4 11_B: CKSC_ILINS_CTL selection /8

NOTE

The setting of this register is only applicable to RLIN30.
The settings 10_B (CKSC_ILINS_CTL selection /4) and 11_B (CKSC_ILINS_CTL selection /8) are only available in UART mode.

(4) CKSC_ILIND_ACT — C_ISO_LIN Clock Divider Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A808$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINDACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.54 CKSC_ILIND_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ILINDACT[1:0]	Clock divider for currently active C_ISO_LIN

(5) CKSC_ILIND_STPM — C_ISO_LIN Stop Mask Register

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A818$_H$

Value after reset: 0000 0002$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILIND STP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 11.55 CKSC_ILIND_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ILINDSTPMSK	0: Clock domain C_ISO_LIN is stopped in stand-by mode. Clock domain C_ISO_LIN is stopped by disabling the C_ISO_LIN source clock selection register (CKSC_ILINS_CTL = 0000 0000$_H$). 1: Clock domain C_ISO_LIN is not stopped in stand-by mode. Clock domain C_ISO_LIN is not stopped by disabling the C_ISO_LIN source clock selection register (CKSC_ILINS_CTL = 0000 0000$_H$).

11.4.3.9 ADCA1 Clock Domain C_ISO_ADCA

(1) CKSC_IADCAS_CTL — C_ISO_ADCA Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 $A500_H$

Value after reset: 0000 0001_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCASCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.56 CKSC_IADCAS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IADCASCSID [1:0]	Source Clock Setting for C_ISO_ADCA 00_B: Disabled 01_B: HS IntOSC (default) 10_B: MainOSC 11_B: PPLLCLK2

CAUTIONS

The CKSC_IADCAS_CTL register and the CKSC_IADCAD_CTL register must be set so that the relationship between frequency (1) and (2) is retained within the range of "(1) / (2) = 2 to 4.8".

(1) Frequency [MHz] specified by the C_ISO_CPUCLK source clock selection register (CKSC_CPUCLKS_CTL) and C_ISO_CPUCLK clock divider selection register (CKSC_CPUCLKD_CTL)

(2) Frequency [MHz] specified by the C_ISO_ADCA source clock selection register (CKSC_IADCAS_CTL) and the C_ISO_ADCA clock divider selection register (CKSC_IADCAD_CTL)

(2) CKSC_IADCAS_ACT — C_ISO_ADCA Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A508$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCASACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.57 CKSC_IADCAS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IADCASACT [1:0]	Source clock for currently active C_ISO_ADCA

(3) CKSC_IADCAD_CTL — C_ISO_ADCA Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A600$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCADCSID [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.58 CKSC_IADCAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IADCADCSID [1:0]	Clock Divider Setting for C_ISO_ADCA 00_B: Setting prohibited 01_B: CKSC_IADCAS_CTL selection /1 (default) 10_B: CKSC_IADCAS_CTL selection /2*1 11_B: Setting prohibited

Note 1. Make sure that the frequency of CKSC_IADCA is no less than 8 MHz after division by 2.

CAUTIONS

The CKSC_IADCAS_CTL register and the CKSC_IADCAD_CTL register must be set so that the relationship between frequency (1) and (2) is retained within the range of “(1) / (2) = 2 to 4.8”.

(1) Frequency [MHz] specified by the C_ISO_CPUCLK source clock selection register (CKSC_CPUCLKS_CTL) and C_ISO_CPUCLK clock divider selection register (CKSC_CPUCLKD_CTL)

(2) Frequency [MHz] specified by the C_ISO_ADCA source clock selection register (CKSC_IADCAS_CTL) and the C_ISO_ADCA clock divider selection register (CKSC_IADCAD_CTL)

(4) CKSC_IADCAD_ACT — C_ISO_ADCA Clock Divider Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A608$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IADCADACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.59 CKSC_IADCAD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IADCADACT [1:0]	Clock divider for currently active C_ISO_ADCA

11.4.3.10 RS-CAN Clock Domains C_ISO_CAN and C_ISO_CANOSC

(1) CKSC_ICANS_CTL — C_ISO_CAN Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A900$_H$

Value after reset: 0000 0003$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANSCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.60 CKSC_ICANS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ICANS CSID[1:0]	Source Clock Setting for C_ISO_CAN*1, *2 00_B: Disabled 01_B: MainOSC 10_B: PPLLCLK 11_B: CPUCLK2 (default)

Note 1. Before transitioning to stand-by mode, select a source clock other than PPLLCLK.

Note 2. Before shifting to STOP mode, select MainOSC.

(2) CKSC_ICANS_ACT — C_ISO_CAN Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 A908$_H$

Value after reset: 0000 0003$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANSACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.61 CKSC_ICANS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICANSACT[1:0]	Source clock for currently active C_ISO_CAN

(3) CKSC_ICANS_STPM — C_ISO_CAN Stop Mask Register

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 A918$_H$

Value after reset: 0000 0002$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANS STP MSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 11.62 CKSC_ICANS_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICANS STPMSK	0: Clock domain C_ISO_CAN is stopped in stand-by mode. 1: Clock domain C_ISO_CAN is not stopped in stand-by mode.

(4) CKSC_ICANOSCD_CTL — C_ISO_CANOSC Clock Divider Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 AA00$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANOSCD CSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.63 CKSC_ICANOSCD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ICANOSCD CSID[1:0]	Clock Divider Setting for C_ISO_CANOSC*1 00_B: Disabled (default) 01_B: MainOSC/1 10_B: MainOSC/2 11_B: Setting prohibited

Note 1. Before shifting to STOP mode, select MainOSC/2.

(5) CKSC_ICANOSCD_ACT — C_ISO_CANOSC Clock Divider Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 AA08$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANOSCDACT [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.64 CKSC_ICANOSCD_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICANOSCDACT [1:0]	Clock divider for currently active C_ISO_CANOSC

(6) CKSC_ICANOSCD_STPM — C_ISO_CANOSC Stop Mask Register

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 AA18$_H$

Value after reset: 0000 0002$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICANOSCDSTPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

Do not change the value after reset of bit 1 from “1”.

Table 11.65 CKSC_ICANOSCD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICANOSCD STPMSK	0: Clock domain C_ISO_CANOSC is stopped in stand-by mode. 1: Clock domain C_ISO_CANOSC is not stopped in stand-by mode.

11.4.3.11 CSI Clock Domain C_ISO_CSI

(1) CKSC_ICSIS_CTL — C_ISO_CSI Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 AB00$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICSISCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.66 CKSC_ICSIS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	ICSISCSID[1:0]	Source Clock Setting for C_ISO_CSI 00_B: Disabled 01_B: CPUCLK2 (default) 10_B: PPLLCLK 11_B: Setting prohibited

(2) CKSC_ICSIS_ACT — C_ISO_CSI Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 AB08$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICSISACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.67 CKSC_ICSIS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICSISACT[1:0]	Source clock for currently active C_ISO_CSI

11.4.3.12 IIC Clock Domain C_ISO_IIC

(1) CKSC_IIICS_CTL — C_ISO_IIC Source Clock Selection Register

The correct write sequence using the PROTCMD1 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources (ISORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 AC00$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IIICSCSID[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 11.68 CKSC_IIICS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	IIICSCSID[1:0]	Source Clock Setting for C_ISO_IIC 00_B: Disabled 01_B: CPUCLK4 (default) 10_B: PPLLCLK2 11_B: Setting prohibited

(2) CKSC_IIICS_ACT — C_ISO_IIC Source Clock Active Register

This register is initialized by all reset sources (ISORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 AC08$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IIICSACT[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.69 CKSC_IIICS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	IIICSACT[1:0]	Source clock for currently active C_ISO_IIC

11.5 Clock Domain Setting Method

11.5.1 Clock Domain Setting

11.5.1.1 Overview of Clock Selector Register

The clock selector for a clock domain C_AWO_<name>/C_ISO_<name> can be controlled by the following registers:

- Source clock selection registers
  These registers select the clock to be used as the domain clock from the available source clocks.
  - AWO source clock selection: CKSC_A<name>S_CTL
  - ISO source clock selection: CKSC_I<name>S_CTL
- Clock divider selection registers
  These registers specify the clock division ratio for the selected source clock.
  - AWO clock division ratio: CKSC_A<name>D_CTL
  - ISO clock division ratio: CKSC_I<name>D_CTL
- Source clock active registers and clock divider active registers
  These registers return the currently active source clock selection and division ratio, respectively.
  - AWO source clock active register/clock divider active register: CKSC_A<name>S_ACT/ CKSC_A<name>D_ACT
  - ISO source clock active register/clock divider active register: CKSC_I<name>S_ACT/ CKSC_I<name>D_ACT

NOTE

- Not all clock selectors provide all of the control functions described above.
- The symbols “I”, which indicates the power domain, are not added to the names of registers in clock domain C_ISO_CPUCLK (CPUCLK, CPUCLK2, and CPUCLK4).

11.5.1.2 Setting Procedure for Clock Domain

Procedure of setting up clock domain is described as below:

1. Set up a source clock.
   - Select a source clock. (CKSC_A<name>S_CTL, CKSC_I<name>S_CTL)
   - Confirm completion of selection. (CKSC _A<name>S_ ACT, CKSC _I<name>S_ ACT)*1
2. Set up a clock divider
   - Select a clock divider. (CKSC _A<name>D_ CTL, CKSC _I<name>D_ CTL)
   - Confirm completion of selection. (CKSC _A<name>D_ ACT, CKSC _I<name>D_ ACT)*2

Note 1. Continue processing after CKSC_A<name>S_ACT and CKSC_I<name>S_ACT are updated with the new values written to CKSC_A<name>S_CTL and CKSC_I<name>S_CTL.

Note 2. Continue processing after CKSC_A<name>D_ACT and CKSC_I<name>D_ACT are updated with the new values written to CKSC_A<name>D_CTL and CKSC_I<name>D_CTL.

CAUTION

The source clock to be selected must be operating before performing these settings.

The behavior and performance are not guaranteed when setup is made while the source clock is stopped.

Access to a peripheral module is prohibited while the clock is not supplied to the module.

11.5.2 Stopping the Clock in Stand-By Mode

In stand-by mode, clock domain C_AWO_<name>/C_ISO_<name> can be configured to stop or continue its clock CKSCLK_A<name>/CKSCLK_I<name> in response to clock stop requests from the stand-by controller.

The clock stop mask registers are used to determine the operation status of the clock in stand-by mode:

- CKSC_A<name>_STPM.A<name>STPMSK/CKSC_I<name>_STPM.I<name>STPMSK = 0:
  The STOP request signal is not masked, so the domain clock CKSCLK_A<name>/ CKSCLK_I<name> is stopped during stand-by mode.
  If the domain clock was in operation before transition to stand-by mode, it is automatically re-started after wake-up from stand-by mode.
  If there is another clock domain which selects same source clock and its stop mask setting is set to 1 (CKSC_A<name>_STPM.A<name>STPMSK/CKSC_I<name>_STPM.I<name>STPMSK = 1), the source clock will continue operation in stand-by mode. The CPU clock domain C_ISO_CPUCLK is always stopped in stand-by mode.
- CKSC_A<name>_STPM.A<name>STPMSK/CKSC_I<name>_STPM.I<name>STPMSK = 1:
  The STOP request signal is masked, so CKSCLK_A<name>/CKSCLK_I<name> continues to operate during stand-by.
  The source clock selected for the target clock domain will also continue to operate in standby mode.
  Supply of clock signals to the clock domains in the ISO area will be stopped in DeepSTOP mode.

11.5.3 Clock Domain Settings

The following table shows the selectable source clocks, frequency division ratios, and registers to be used for each clock domain.

Table 11.70 List of Selectable Clocks (1/2)

Clock Domain	Domain Clock	Source Clock Selection Register		Clock Divider Selection Register		Target Unit
C_AWO_WDTA	CKSCLK_AWDTA	—	LS IntOSC	CKSC_AWDTAD_CTL	1/1	WDTA0
					1/128	
C_AWO_TAUJ	CKSCLK_ATAUJ	CKSC_ATAUJS_CTL	MainOSC	CKSC_ATAUJD_CTL	**1/1**	TAUJ0
			HS IntOSC		1/2	
			LS IntOSC		1/4	
			PPLLCLK2		1/8	
			Disable		—	
C_AWO_RTCA	CKSCLK_ARTCA	CKSC_ARTCAS_CTL	MainOSC	CKSC_ARTCAD_CTL	1/1	RTCA0
			LS IntOSC		1/2	
			SubOSC		1/4	
			Disable		1/8	
			—		**Disable**	
C_AWO_ADCA	CKSCLK_AADCA	CKSC_AADCAS_CTL	MainOSC	CKSC_AADCAD_CTL	**1/1**	ADCA0
			HS IntOSC		1/2	
			PPLLCLK2		—	
			Disable			
C_AWO_FOUT	CKSCLK_AFOUT	CKSC_AFOUTS_CTL	MainOSC	—	**1/1**	FOUT
			HS IntOSC			
			LS IntOSC			
			SubOSC			
			CPLLCLK4			
			PPLLCLK4			
			Disable			
C_ISO_CPUCLK	CKSCLK_CPUCLK	CKSC_CPUCLKS_CTL	MainOSC	CKSC_CPUCLKD_CTL	**1/1**	CPU subsystem MEMC FLXAn
			CPLLCLK		1/2	
			EMCLK		1/4	
			—		1/8	
C_ISO_PERI1	CKSCLK_IPERI1	CKSC_IPERI1S_CTL	PPLLCLK	—	**1/1**	TAUD0 TAUJ1 ENCA0 TAPA0 PIC0
			CPUCLK2			
			Disable			
C_ISO_PERI2	CKSCLK_IPERI2	CKSC_IPERI2S_CTL	**CPUCLK2**	—	**1/1**	TAUBn PWBAn PWGAn PWSAn
			PPLLCLK2			
			Disable			
C_ISO_LIN	CKSCLK_ILIN	CKSC_ILINS_CTL	MainOSC	CKSC_ILIND_CTL*1	**1/1**	RLIN2m RLIN3n
			CPUCLK2		1/4	
			PPLLCLK2		1/8	
			Disable		—	
C_ISO_ADCA	CKSCLK_IADCA	CKSC_IADCAS_CTL	MainOSC	CKSC_IADCAD_CTL	**1/1**	ADCA1
			HS IntOSC		1/2	
			PPLLCLK2		—	
			Disable			

Table 11.70 List of Selectable Clocks (2/2)

Clock Domain	Domain Clock	Source Clock Selection Register		Clock Divider Selection Register		Target Unit
C_ISO_CAN	CKSCLK_ICAN	CKSC_ICANS_CTL	MainOSC	—	**1/1**	RSCANn
			PPLLCLK			
			CPUCLK2			
			Disable			
C_ISO_CANOSC	CKSCLK_ICANOSC	—	MainOSC	CKSC_ICANOSCD_CTL	1/1	RSCANn
					1/2	
					Disable	
C_ISO_CSI	CKSCLK_ICSI	CKSC_ICSIS_CTL	PPLLCLK	—	**1/1**	CSIGn CSIHn
			CPUCLK2			
			Disable			
C_ISO_IIC	CKSCLK_IIIC	CKSC_IIICS_CTL	**CPUCLK4**	—	**1/1**	RIICn
			PPLLCLK2			
			Disable			

Note: The items written in **bold** are the default clocks for each register.

Note 1. The setting of this register is only applicable to RLIN30. The settings 1/4 and 1/8 are only available in UART mode.

CAUTION

To stop the source clock selected for the clock domain before transitioning to STOP/ DeepSTOP mode, select “Disable” for that clock domain in advance. Do not stop the source clock of a clock domain for which "Disable" cannot be selected while functions are operating on that clock domain. To stop the clock source selected for the domain by transitioning to STOP/DeepSTOP mode, "Disable" does not need to be selected. Instead of the setting, select stop of clock domain during stand-by mode by the stop mask register.

11.6 Frequency Output Function (FOUT)

The frequency output function (FOUT) allows to output the clock as the external signal. Furthermore, the frequency can be divided by the clock divider before it is output.

11.6.1 Functional Overview

Figure 11.10 shows the configuration of the frequency output function.

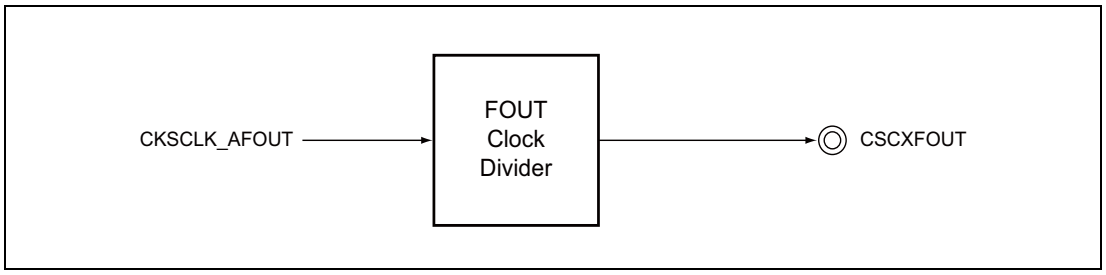


Figure 11.10 Frequency Output Function

The clock output function outputs the CKSCLK_AFOUT clock divided by 1 to 63 through the clock divider from CSCXFOUT. Division ratio N is set to the FOUTDIV[5:0] bits in the FOUTDIV register. Clock output frequency $f_{CSCXFOUT}$ is expressed by the following equation:

$$f_{CSCXFOUT} = (\text{CKSCLK\_AFOUT clock frequency}) / N$$

Clock output starts when, after CKSCLK_AFOUT is set and the clock output for the pin function is selected, division ratio N is set to the FOUTDIV[5:0] bits in the FOUTDIV register.

When a new division ratio is written to the FOUTDIV.FOUTDIV[5:0] bits, it becomes effective in synchronization with the CSCXFOUT output clock. Accordingly, the division ratio can be changed even while the CSCXFOUT clock is operating. The clock output is stopped by writing 000_H to the FOUTDIV[5:0] bits.

CAUTION

In DeepSTOP mode, the frequency output function (FOUT) cannot be used.

11.6.2 Registers

11.6.2.1 List of Registers

The FOUT registers are listed in the following table.

Table 11.71 CSCXFOUT Clock Divider Registers

Register Name	Symbol	Address
Clock division ratio register	FOUTDIV	FFF8 2800_H
Clock divider status register	FOUTSTAT	FFF8 2804_H

11.6.2.2 Clock Supply

The clock supply to the CSCXFOUT is shown in the following table.

Table 11.72 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name	Description
CSCXFOUT	PCLK	CPUCLK2	Bus clock (Register access)
	CKSCLK_AFOUT	CKSCLK_AFOUT	Clock source of FOUT clock divider

11.6.2.3 FOUTDIV — Clock Division Ratio Register

This register defines the clock division ratio.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 2800$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	FOUTDIV[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.73 FOUTDIV Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	FOUTDIV[5:0]	Clock Division Ratio N 00_H: Clock output is stopped 01_H: N = 1 02_H: N = 2 ... $3E_H$: N = 62 $3F_H$: N = 63

11.6.2.4 FOUTSTAT — Clock Divider Status Register

This register indicates the clock output status.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 2804$_H$

Value after reset: 0000 0001$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUTCLKACT	FOUTSYNC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.74 FOUTSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	FOUTCLKACT	Clock Divider Output Active 0: Frequency output is stopped. 1: Frequency output is active.
0	FOUTSYNC	Clock Divider Synchronized 0: The clock divider is in the process of synchronization. 1: The clock divider is stable (or stopped).

11.7 Clock Monitor A (CLMA)

The following sections describe clock monitor A (CLMA).

The first section describes the attributes specific to the RH850/F1M microcontrollers, including the number of channels, register base addresses, and input/output signal names. The ensuing sections describe the functions relevant to all operations.

11.7.1 Features of RH850/F1M CLMA

11.7.1.1 Number of Channels

The RH850/F1M microcontrollers incorporate CLMA with the following number of channels.

Table 11.75 Number of Channels

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of channels	3		
Name	CLMA0, CLMA1, CLMA2		

11.7.1.2 Register Base Addresses

The CLMA base addresses are listed in the following table.

The CLMA register addresses are expressed as offsets from the base addresses.

Table 11.76 Register Base Addresses

Base Address Name	Base Address
<CLMA0_base>	FFF8 $C000_H$
<CLMA1_base>	FFF8 $D000_H$
<CLMA2_base>	FFF8 $E000_H$

11.7.1.3 Clock Supply

The clocks monitored by CLMA and the CLMA sampling clocks are indicated below.

Table 11.77 Clock Supply

Channel Name	Unit Clock Name	Clock Name
CLMA0	CLMATMON (monitored clock)	HS IntOSC
	CLMATSMP (sampling clock)	LS IntOSC
	Register access clock	CPUCLK2
CLMA1	CLMATMON (monitored clock)	MainOSC
	CLMATSMP (sampling clock)	LS IntOSC
	Register access clock	CPUCLK2
CLMA2	CLMATMON (monitored clock)	CPLLCLK
	CLMATSMP (sampling clock)	HS IntOSC
	Register access clock	CPUCLK2

11.7.1.4 Reset Sources

The reset sources of the CLMA are listed in the following table. The CLMA are initialized by these reset sources.

Table 11.78 Reset Sources

Channel Name	Reset Source
CLMA0	Reset sources other than transition to the DeepSTOP mode (AWORES)
CLMA1	Reset sources other than transition to the DeepSTOP mode (AWORES)
CLMA2	All reset source (ISORES)
Common Registers (CLMATEST, CLMATESTS)	Reset sources other than transition to the DeepSTOP mode (AWORES)

11.7.1.5 Internal Input/Output Signals

The internal input/output signals of CLMA are listed in the following table.

Table 11.79 Internal Input/Output Signals

Unit Signal Name	Description	Connection
$\overline{\text{CLMATRES}}$	CLMA0 error reset output	Reset controller ($\overline{\text{CLMA0RES}}$)
$\overline{\text{CLMATRES}}$	CLMA1 error reset output	Reset controller ($\overline{\text{CLMA1RES}}$)
$\overline{\text{CLMATRES}}$	CLMA2 error reset output	Reset controller ($\overline{\text{CLMA2RES}}$)

11.7.2 Overview

11.7.2.1 Functional Overview

Clock monitor CLMA detects frequency abnormalities in the monitored clock. It uses sampling clock CLMATSMP to monitor whether the frequency of input clock CLMATMON is within a specific range. Upon detection of an abnormal clock, it outputs a reset request signal.

The main components of the clock monitor are shown in **Figure 11.11**.

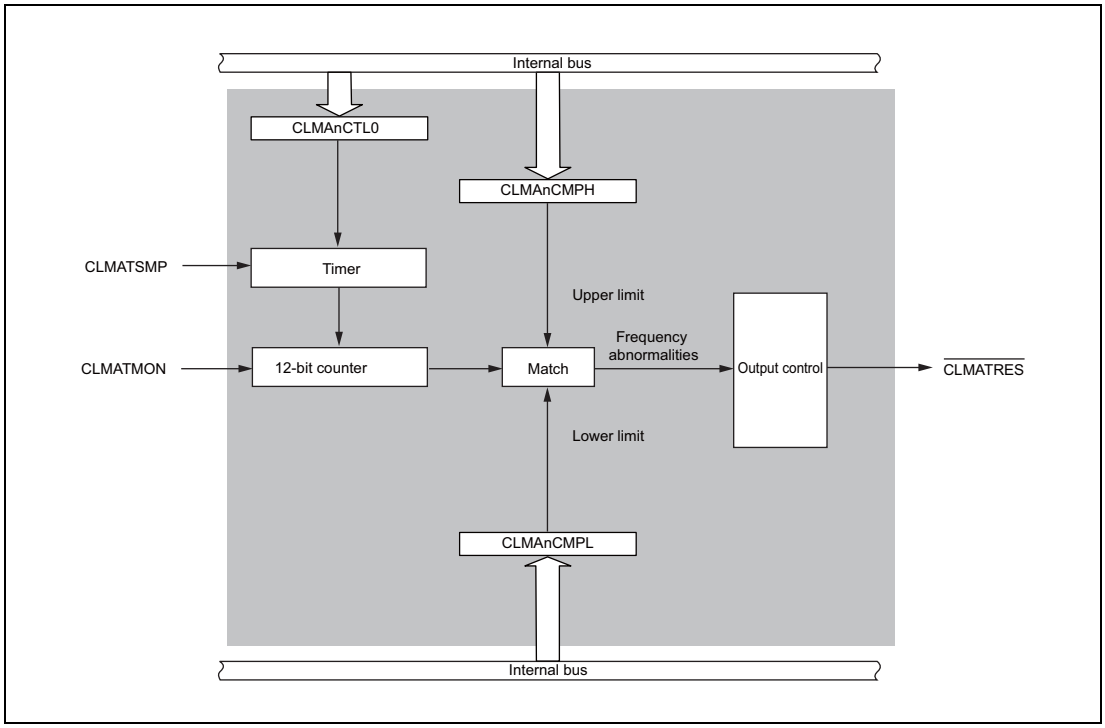


Figure 11.11 Block Diagram of the Clock Monitor A

NOTE

Once enabled, only a reset can disable CLMA.

11.7.3 Enabling CLMA

Clock monitoring is started by the clock monitor when CLMAnCTL0.CLMAnCLME = 1.

When the monitored clock is stopped by register operation or stand-by mode, the corresponding clock monitor is automatically disabled. After the monitored clock starts oscillation again and becomes stable, the clock monitor also starts operation.

Since CLMA2 is initialized on return from DeepSTOP mode, the CLMA2 register must be set again before further operation is started.

11.7.4 Functions

11.7.4.1 Detection of Abnormal Clock Frequencies

Detection Method

1. CLMAn counts the rising edges of the monitored clock CLMATMON within 16 cycles of the sampling clock CLMATSMP and then compares the counter value with the specified thresholds:
   - CLMAnCMPL.CLMAnCMPL[11:0] defines the lower threshold.
   - CLMAnCMPH.CLMAnCMPH[11:0] defines the upper threshold.
2. When CLMATMON frequency*1 is lower than the limit, the counter falls below CLMAnCMPL.CLMAnCMPL[11:0].
3. When the frequency of CLMATMON is higher than the limit, the counter exceeds CLMAnCMPH.CLMAnCMPH[11:0].

Note 1. There is a case that the abnormal state is not detected when the monitor clock completely stops.

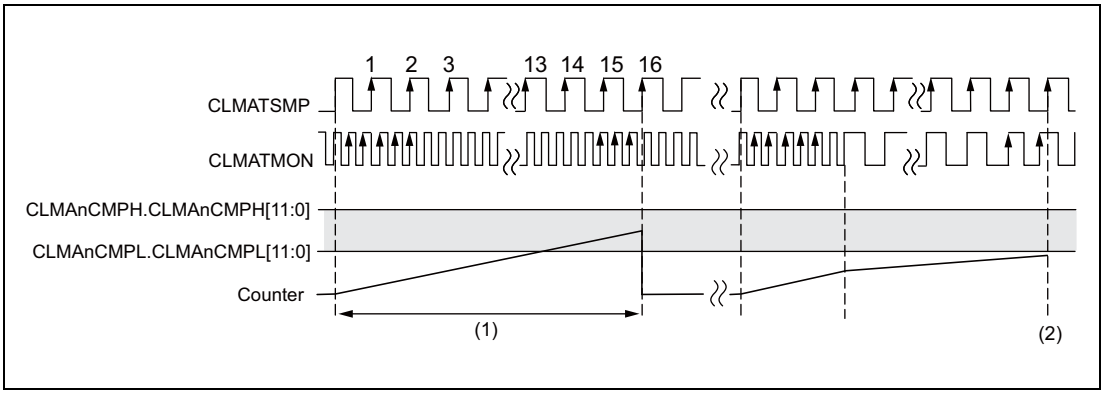


Figure 11.12 Example: $f_{CLMATMON}$ is Lower than the Specified Limit

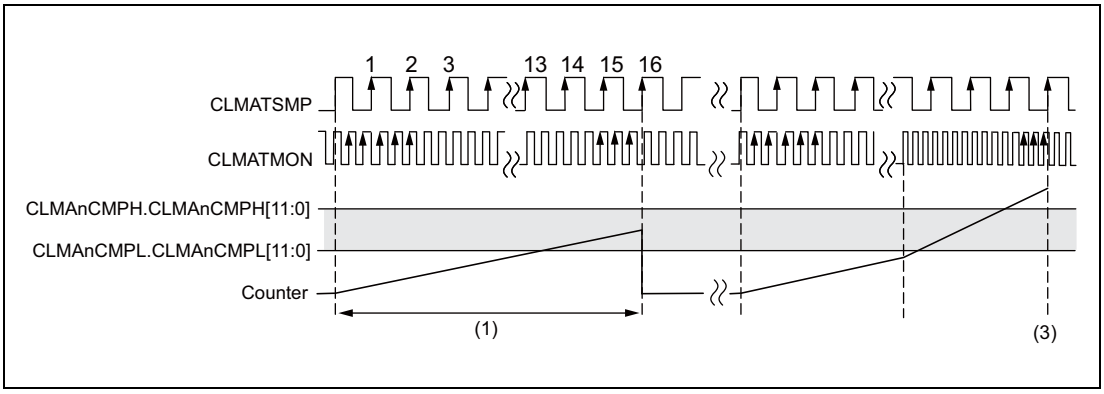


Figure 11.13 Example: $f_{CLMATMON}$ is Higher than the Specified Limit

NOTE

Even if $f_{CLMATMON}$ exceeds or falls below the specified limits during a sampling interval, the counter might be within the valid range.

Abnormal $f_{CLMATMON}$ is detected after one sampling interval.

(1) Calculation method of the thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]

The compare registers CLMAnCMPL and CLMAnCMPH are configured with the minimum and maximum number of clock cycles of CLMATMON that are assumed to be valid within 16 cycles of the sampling clock CLMATSMP.

The expected number of clock cycles is denoted by N.

$$\frac{16}{f_{CLMATSMP}} = \frac{N}{f_{CLMATMON}}$$

$$N = \frac{f_{CLMATMON}}{f_{CLMATSMP}} \times 16$$

Considering the allowed frequency deviations of CLMATMON and CLMATSMP, the threshold values can be calculated by the following formulas:

$$\text{Lower threshold} = N_{min} = \frac{f_{CLMATMON(dmin)}}{f_{CLMATSMP(max)}} \times 16 - 1$$

$$\text{Upper threshold} = N_{max} = \frac{f_{CLMATMON(max)}}{f_{CLMATSMP(min)}} \times 16 + 1$$

NOTE

The jitter of the PLL is covered by “+1” and “–1” in the formulas.

Example:

When $f_{CLMATSMP}$ = 240 kHz (±8%) and $f_{CLMATMON}$ = 16 MHz (±5%), the recommended threshold values are as follows:

N_{min} = 15,200 / 259.2 × 16 - 1
= 937.27
CLMAnCMPL = 937 = $03A9_H$

N_{max} = 16,800 / 220.8 × 16 + 1
= 1218.39
CLMAnCMPH = 1219 = $04C3_H$

Minimum thresholds

The following restrictions must be taken into account:

- CLMAnCMPL ≥ 0001_H
- CLMAnCMPH ≥ CLMAnCMPL + 0003_H

(2) Definition of the initial value input to the threshold registers

The values after reset of the threshold registers are set so that the maximum frequency deviation of the monitored clock is allowed:

- CLMAnCMPL[11:0] = 001_H
- CLMAnCMPH[11:0] = $3FF_H$

11.7.4.2 Notification of Abnormal Clock Frequency

If $f_{CLMATMON}$ exceeds the upper threshold or falls below the lower threshold, this is indicated as follows:

1. The reset request signal $\overline{\text{CLMATRES}}$ is set to low level.
2. The system reset (AWORES and ISORES) is generated and CLMAn is reset.

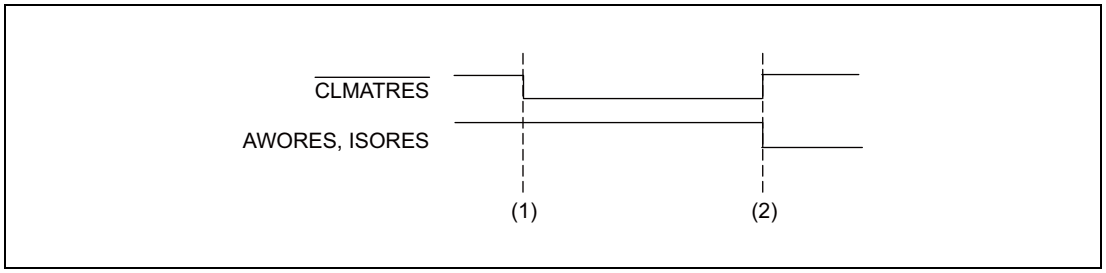


Figure 11.14 Error Request Signal Output if $f_{CLMATMON}$ Exceeds Upper Threshold

CAUTION

For usage notes for CLMAn abnormality detection, see Section 11.7.6, Usage Notes for CLMAn.

11.7.5 Registers

11.7.5.1 List of Registers

The following table lists the CLMA registers.

<CLMAn_base> is defined in **Section 11.7.1.2, Register Base Addresses**.

Table 11.80 List of Clock Monitor Registers

Module Name	Register Name	Symbol	Address
CLMAn	CLMAn control register 0	CLMAnCTL0	<CLMAn_base> + 00_H
CLMAn	CLMAn compare register L	CLMAnCMPL	<CLMAn_base> + 08_H
CLMAn	CLMAn compare register H	CLMAnCMPH	<CLMAn_base> + $0C_H$
CLMAn	CLMA test register	CLMATEST	FFF8 $C100_H$
CLMAn	CLMA test status register	CLMATESTS	FFF8 $C104_H$
CLMAn	CLMAn emulation register 0	CLMAnEMU0	<CLMAn_base> + 18_H

11.7.5.2 CLMAnCTL0 — CLMAn Control Register 0

This register enables the clock monitor CLMAn.

The correct write sequence using the CLMAnPCMD register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 8-bit units.

Address: <CLMAn_base> + 00_H

Value after reset: 00_H

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

Table 11.81 CLMAnCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CLMAnCLME	Enables or disables the clock monitor. 0: CLMAn is disabled. 1: CLMAn is enabled.

CAUTION

After the CLMAnCTL0.CLMAnCLME bit is set to 1, writing 0 is ignored. The only condition for clearing the bit is a reset (AWORES, ISORES).
In addition, it is also cleared when CLMATEST.RESCLM bit is set to 1 during self-test of CLMAn.

11.7.5.3 CLMAnCMPH — CLMAn Compare Register H

This register specifies the upper limit of frequency.

It can only be written when CLMAn is disabled (CLMAnCTL0.CLMAnCLME = 0).

For details, see **Section 11.7.4.1 (1) Calculation method of the thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]**.

Access: This register can be read or written in 16-bit units.

Address: <CLMAn_base> + $0C_H$

Value after reset: $03FF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CLMAnCMPH[11:0]											
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.82 CLMAnCMPH Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CLMAnCMPH [11:0]	Specifies the upper threshold. • The recommended value is $f_{CLMATMON\ (max)}$ / $f_{CLMATSMP\ (min)}$ × 16 +1. • The minimum value is CLMAnCMPL + 0003_H.

11.7.5.4 CLMAnCMPL — CLMAn Compare Register L

This register specifies the lower limit of frequency.
It can only be written when CLMAn is disabled (CLMAnCTL0.CLMAnCLME=0).

For details, see **Section 11.7.4.1 (1) Calculation method of the thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]**.

Access: This register can be read or written in 16-bit units.

Address: <CLMAn_base> + 08_H

Value after reset: 0001_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CLMAnCMPL[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.83 CLMAnCMPL Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CLMAnCMPL [11:0]	Specifies the lower threshold. • The recommended value is $f_{CLMATMON\ (min)}$ / $f_{CLMATSMP\ (max)}$ × 16 – 1 • The minimum value is 0001_H.

11.7.5.5 CLMATEST — CLMA Test Register

This register is used to control self-test of CLMA0, CLMA1, and CLMA2.

The correct write sequence using the PROTCMDCLMA register is required in order to update this register.

For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFF8 $C100_H$

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CLMA2 TESEN	CLMA1 TESEN	CLMA0 TESEN	ERR MSK	MONCL KMSK	RES CLM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.84 CLMATEST Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CLMA2TESEN	CLMA2 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
4	CLMA1TESEN	CLMA1 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
3	CLMA0TESEN	CLMA0 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
2	ERRMSK*1	CLMA Test Reset Signal Mask Setting 0: Signal generation enabled 1: Signal generation disabled (masked)
1	MONCLKMSK*1	Monitor Clock Mask Setting 0: Monitor clock enabled 1: Monitor clock disabled (masked)
0	RESCLM*1	CLMAn Test Reset Signal Control 0: Reset released 1: Reset executed

Note 1. These bits are effective when the CLMAnTESEN bit is 1.

11.7.5.6 CLMATESTS — CLMA Test Status Register

This register is used to confirm the self-test result of CLMA0, CLMA1, and CLMA2.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFF8 C104$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CLMA2 ERRS	CLMA1 ERRS	CLMA0 ERRS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.85 CLMATESTS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	CLMA2ERRS	CLMA2 Error Status 0: Errors are not detected 1: Errors are detected
1	CLMA1ERRS	CLMA1 Error Status 0: Errors are not detected 1: Errors are detected
0	CLMA0ERRS	CLMA0 Error Status 0: Errors are not detected 1: Errors are detected

11.7.5.7 CLMAnEMU0 — CLMAn Emulation Register 0

This register is used to set pseudo flags for emulation.

This register can be accessed only during a break by the debugger and is reset by a break release.

Access: This register can be read or written in 8-bit units.

Address: <CLMAn_base> + 18_H

Value after reset: 00_H

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

Table 11.86 CLMAnEMU0 Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CLMAnSLFST	Specifies whether $f_{CLMATMON}$ is assumed to be high. 0: CLMATMON is assumed to be within the normal frequency range. 1: CLMATMON is assumed to exceed the upper threshold.
0	CLMAnSLSLW	Specifies whether $f_{CLMATMON}$ is assumed to be low. 0: CLMATMON is assumed to be within the normal frequency range 1: CLMATMON is assumed to fall below the lower threshold.

CAUTION

It is prohibited to emulate a low and high CLMATMON at the same time. Thus CLMAnEMU0 must not be set to 03_H.

11.7.6 Usage Notes for CLMAn

Do not use a clock for which CLMAn detected an abnormality. The behavior and performance are not guaranteed if such a clock is used. When the CLMA0 detects clock abnormality, modifying any of the clock domain settings is prohibited.

Table 11.87 provides usage notes for each CLMAn.

Table 11.87 Usage Notes for CLMAn

Monitor Clock	CPU Clock after CLMA Reset Release	Note
HS IntOSC (CLMA0)	EMCLK*1	Set ROSCE.ROSCDISTRG to 1.*1 Do not set the control registers of the MainOSC, the PLL0, and the PLL1. After $\overline{\text{CLMA0RES}}$ occurs, modifying any of the clock domain settings is prohibited.
MainOSC (CLMA1)	EMCLK (= HS IntOSC)	Do not set the control registers of the MainOSC, the PLL0 and the PLL1. After occurrence of the $\overline{\text{CLMA1RES}}$, do not select a clock whose source clock is the MainOSC, the PLL0 or the PLL1.
PLL (CLMA2)	EMCLK (= HS IntOSC)	Do not set the control registers of the PLL0. After occurrence of the $\overline{\text{CLMA2RES}}$, do not select a clock whose source clock is the PLL0.

Note 1. The state of the EMCLK after reset by the $\overline{\text{CLMA0RES}}$ depends on the state of HS IntOSC oscillation. In the case the HS IntOSC is completely stopped, the LS IntOSC is supplied as the EMCLK. In the case the HS IntOSC continues oscillation but the frequency is outside of the lower threshold and the upper threshold of CLMA0, the HS IntOSC is still supplied as EMCLK. After setting the ROSCE.ROSCDISTRG bit to 1, the EMCLK is switched from the HS IntOSC to the LS IntOSC.

Section 12 Stand-By Controller (STBC)

This section describes the functions of the stand-by controller (STBC), the registers, and various stand-by modes.

12.1 Functions

12.1.1 Types of Stand-By Mode

The RH850/F1M supports HALT mode and the following four stand-by modes:

- HALT mode
  HALT mode can be entered from normal RUN mode by performing the CPU1 instruction "HALT".
  This stops the CPU1 operation, while all clocks other than the CPU clock continue to operate and power continues to be supplied.
  For details, see the *RH850G3M User's Manual: Software*.
- STOP mode
  In STOP mode, the clock supply to a certain clock domain can be stopped.
  STOP mode is entered when the STBC0STPT.STBC0STPTRG bit is set to 1.
  The clock supply to clock domains can continue even in STOP mode by setting CKSC_xxx_STPM.xxxSTPMSK = 1. For details on the CKSC_xxx_STPM register, see **Section 11, Clock Controller**.
- DeepSTOP mode
  In order to further reduce power consumption, the power supply of the Isolated area can be switched off.
  DeepSTOP mode is entered when the STBC0PSC.STBC0DISTRG is set to 1.
- Cyclic RUN mode
  In this mode, only the CPU1, peripherals on Always-ON area, and RLIN3 can operate.
  The CPU1 executes the instructions in the retention RAM.
  In DeepSTOP mode, the mode transitions to Cyclic RUN mode when wake-up factor 2 is generated, and in Cyclic STOP mode, the mode transitions to Cyclic RUN mode when either wake-up factor 1 or wake-up factor 2 is generated.
- Cyclic STOP mode
  This mode stops the CPU1 in addition to the functions stopped in Cyclic RUN mode.
  This mode is entered by setting the STBC0STPT.STBC0STPTRG bit to 1 in Cyclic RUN mode.

NOTE

The mode in which CPU1 is running is called RUN mode.

12.1.2 Wake-Up Control

12.1.2.1 Wake-Up Factors for Stand-By Modes

The stand-by controller can initiate return from stand-by mode by the following wake-up factors.

Table 12.1 Wake-Up Factor 1

Wake-Up Factor	Unit	STOP →RUN	DeepSTOP →RUN	Cyclic RUN →RUN*1	Cyclic STOP →RUN*1
TNMI	Port	√	√	√	√
WDTA0NMI	WDTA0	√	√	√	√
INTLVIL*3	LVI	√	√	√	√
INTP0	Port	√	√	√	√
INTP1	Port	√	√	√	√
INTP2	Port	√	√	√	√
INTWDTA0	WDTA0	√	√	√	√
INTP3	Port	√	√	√	√
INTP4	Port	√	√	√	√
INTP5	Port	√	√	√	√
INTP10	Port	√	√	√	√
INTP11	Port	√	√	√	√
WUTRG1	LPS	√	√	√	√
INTTAUJ0I0	TAUJ0	√	√	√	√
INTTAUJ0I1	TAUJ0	√	√	√	√
INTTAUJ0I2	TAUJ0	√	√	√	√
INTTAUJ0I3	TAUJ0	√	√	√	√
WUTRG0	LPS	√	√	√	√
INTP6	Port	√	√	√	√
INTP7	Port	√	√	√	√
INTP8	Port	√	√	√	√
INTP12	Port	√	√	√	√
INTP9	Port	√	√	√	√
INTP13	Port	√	√	√	√
INTP14	Port	√	√	√	√
INTP15	Port	√	√	√	√
INTRTCA01S	RTCA0	√	√	√	√
INTRTCA0AL	RTCA0	√	√	√	√
INTRTCA0R	RTCA0	√	√	√	√
INTDCUTDI	JTAG	√	√	√	√
INTKR0	KR0	√	—	—	—
INTRCANGRECC0*2	RSCAN0	√	—	—	—
INTRCAN0REC*2	RSCAN0	√	—	—	—
INTRCAN1REC*2	RSCAN0	√	—	—	—
INTRCAN2REC*2	RSCAN0	√	—	—	—
INTRCAN3REC*2	RSCAN0	√	—	—	—
INTRCAN4REC*2	RSCAN0	√	—	—	—
INTRCAN5REC*2	RSCAN0	√	—	—	—

Note 1. The mode returns to RUN mode via DeepSTOP mode. When the transition from Cyclic STOP mode to Cyclic RUN mode is made by Wake-Up Factor 1, if the transition to DeepSTOP mode by STBC0PSC.STBC0DISTRG is made without clearing Wake-Up Factor 1, the transition to RUN is made.

Note 2. By using the INTP external interrupt assigned to the alternate-function pin shared with the CAN reception pin, wake-up from stand-by modes such as DeepSTOP mode is possible. As the trigger for waking up from DeepSTOP mode, use a pin of port P0 or P1, which is assigned to the AWO area.

Note 3. Cannot be cleared while REGVCC is below the reference voltage (REGVCC (min.) < VLVI).
To clear the wake-up factor flag (WUF0[2]), WUFC0[2] must be set while WUFMSK0[2] = 1 and REGVCC is above the reference voltage (REGVCC (min.) > VLVI). The INTLVIH interrupt can be used to check that REGVCC is above the reference voltage.

Table 12.2 Wake-Up Factor 2

Wake-Up Factor	Unit	DeepSTOP → Cyclic RUN	Cyclic STOP → Cyclic RUN
INTADCA0I0	ADCA0	—	√*1
INTADCA0I1	ADCA0	—	√*1
INTADCA0I2	ADCA0	—	√*1
INTRLIN30	RLIN30	—	√
INTTAUJ0I0	TAUJ0	√	√
INTTAUJ0I1	TAUJ0	√	√
INTTAUJ0I2	TAUJ0	√	√
INTTAUJ0I3	TAUJ0	√	√
INTRLIN31	RLIN31	—	√
INTRLIN32	RLIN32	—	√
INTRTCA01S	RTCA0	√	√
INTRTCA0AL	RTCA0	√	√
INTRTCA0R	RTCA0	√	√
INTRLIN33	RLIN33	—	√
INTRLIN34	RLIN34	—	√
INTRLIN35	RLIN35	—	√

Note 1. These wake-up factors are only available in LPS analog input mode.

CAUTION

For the pins of the function used for the wake-up factors from DeepSTOP mode, use the multiplexed functions of the ports assigned to the AWO area.

12.1.2.2 Setting of Wake-Up Factors

Wake-up factors for returning from stand-by modes are controlled by the following stand-by controller registers:

- Wake-up factor registers: WUF0, WUF20, WUF_ISO0

  Upon occurrence of an effective wake-up factor, the associated wake-up factor flag is set to 1. By checking these registers and their flags, it is possible to identify the wake-up factor.

- Wake-up factor mask registers: WUFMSK0, WUFMSK20, WUFMSK_ISO0

  Each bit of these registers is assigned to a certain wake-up factor. Wake-up by this factor is enabled if its mask bit is set to 0. Wake-up factors assigned to both wake-up factor 1 and 2 should not be enabled at the same time.

- Wake-up factor clear registers: WUFC0, WUFC20, WUFC_ISO0

  By setting the applicable bits in these registers to 1, the Wake-up factor bit (WUFy) in the Wake-up factor registers (WUF0, WUF20, WUF_ISO0) can be cleared.

NOTE

The Wake-up factor flags in the Wake-up factor registers (WUF0, WUF20, and WUF_ISO0) only indicate the occurrence of wake-up factor. These flags do not indicate a transition from stand-by mode to normal mode.

The assignment of the wake-up factors to the control register bits and status register bits are shown in the following table.

For details about the wake-up control and status registers, see **Section 12.2.2.3, WUF0/WUF20/WUF_ISO0 — Wake-Up Factor Registers**, **Section 12.2.2.4, WUFMSK0/WUFMSK20/WUFMSK_ISO0 — Wake-Up Factor Mask Registers**, and **Section 12.2.2.5, WUFC0/WUFC20/WUFC_ISO0 — Wake-Up Factor Clear Registers**.

Table 12.3 Wake-Up Factor 1 Register Assignment (1/2)

Wake-Up Factor	Register Bit assignment			Unit	144 pins	176 pins	233 pins
TNMI	WUF0[0]	WUFMSK0[0]	WUFC0[0]	Port	√	√	√
WDTA0NMI	WUF0[1]	WUFMSK0[1]	WUFC0[1]	WDTA0	√	√	√
INTLVIL	WUF0[2]	WUFMSK0[2]	WUFC0[2]	LVI	√	√	√
INTP0	WUF0[5]	WUFMSK0[5]	WUFC0[5]	Port	√	√	√
INTP1	WUF0[6]	WUFMSK0[6]	WUFC0[6]	Port	√	√	√
INTP2	WUF0[7]	WUFMSK0[7]	WUFC0[7]	Port	√	√	√
INTWDTA0	WUF0[8]	WUFMSK0[8]	WUFC0[8]	WDTA0	√	√	√
INTP3	WUF0[9]	WUFMSK0[9]	WUFC0[9]	Port	√	√	√
INTP4	WUF0[10]	WUFMSK0[10]	WUFC0[10]	Port	√	√	√
INTP5	WUF0[11]	WUFMSK0[11]	WUFC0[11]	Port	√	√	√
INTP10	WUF0[12]	WUFMSK0[12]	WUFC0[12]	Port	√	√	√
INTP11	WUF0[13]	WUFMSK0[13]	WUFC0[13]	Port	√	√	√
WUTRG1	WUF0[14]	WUFMSK0[14]	WUFC0[14]	LPS	√	√	√
INTTAUJ0I0	WUF0[15]	WUFMSK0[15]	WUFC0[15]	TAUJ0	√	√	√
INTTAUJ0I1	WUF0[16]	WUFMSK0[16]	WUFC0[16]	TAUJ0	√	√	√
INTTAUJ0I2	WUF0[17]	WUFMSK0[17]	WUFC0[17]	TAUJ0	√	√	√

Table 12.3 Wake-Up Factor 1 Register Assignment (2/2)

Wake-Up Factor	Register Bit assignment			Unit	144 pins	176 pins	233 pins
INTTAUJ0I3	WUF0[18]	WUFMSK0[18]	WUFC0[18]	TAUJ0	√	√	√
WUTRG0	WUF0[19]	WUFMSK0[19]	WUFC0[19]	LPS	√	√	√
INTP6	WUF0[20]	WUFMSK0[20]	WUFC0[20]	Port	√	√	√
INTP7	WUF0[21]	WUFMSK0[21]	WUFC0[21]	Port	√	√	√
INTP8	WUF0[22]	WUFMSK0[22]	WUFC0[22]	Port	√	√	√
INTP12	WUF0[23]	WUFMSK0[23]	WUFC0[23]	Port	√	√	√
INTP9	WUF0[24]	WUFMSK0[24]	WUFC0[24]	Port	√	√	√
INTP13	WUF0[25]	WUFMSK0[25]	WUFC0[25]	Port	√	√	√
INTP14	WUF0[26]	WUFMSK0[26]	WUFC0[26]	Port	√	√	√
INTP15	WUF0[27]	WUFMSK0[27]	WUFC0[27]	Port	√	√	√
INTRTCA01S	WUF0[28]	WUFMSK0[28]	WUFC0[28]	RTCA0	√	√	√
INTRTCA0AL	WUF0[29]	WUFMSK0[29]	WUFC0[29]	RTCA0	√	√	√
INTRTCA0R	WUF0[30]	WUFMSK0[30]	WUFC0[30]	RTCA0	√	√	√
INTDCUTDI	WUF0[31]	WUFMSK0[31]	WUFC0[31]	JTAG	√	√	√
INTKR0	WUF_ISO0[1]	WUFMSK_ISO0[1]	WUFC_ISO0[1]	KR0	√	√	√
INTRCANGRECC0	WUF_ISO0[2]	WUFMSK_ISO0[2]	WUFC_ISO0[2]	RSCAN0	√	√	√
INTRCAN0REC	WUF_ISO0[3]	WUFMSK_ISO0[3]	WUFC_ISO0[3]	RSCAN0	√	√	√
INTRCAN1REC	WUF_ISO0[4]	WUFMSK_ISO0[4]	WUFC_ISO0[4]	RSCAN0	√	√	√
INTRCAN2REC	WUF_ISO0[5]	WUFMSK_ISO0[5]	WUFC_ISO0[5]	RSCAN0	√	√	√
INTRCAN3REC	WUF_ISO0[6]	WUFMSK_ISO0[6]	WUFC_ISO0[6]	RSCAN0	√	√	√
INTRCAN4REC	WUF_ISO0[7]	WUFMSK_ISO0[7]	WUFC_ISO0[7]	RSCAN0	√	√	√
INTRCAN5REC	WUF_ISO0[8]	WUFMSK_ISO0[8]	WUFC_ISO0[8]	RSCAN0	√	√	√

Table 12.4 Wake-Up Factor 2 Register Assignment

Wake-Up Factor	Register Bit assignment			Unit	144 pins	176 pins	233 pins
INTADCA0I0	WUF20[0]	WUFMSK20[0]	WUFC20[0]	ADCA0	√	√	√
INTADCA0I1	WUF20[1]	WUFMSK20[1]	WUFC20[1]	ADCA0	√	√	√
INTADCA0I2	WUF20[2]	WUFMSK20[2]	WUFC20[2]	ADCA0	√	√	√
INTRLIN30	WUF20[3]	WUFMSK20[3]	WUFC20[3]	RLIN30	√	√	√
INTTAUJ0I0	WUF20[4]	WUFMSK20[4]	WUFC20[4]	TAUJ0	√	√	√
INTTAUJ0I1	WUF20[5]	WUFMSK20[5]	WUFC20[5]	TAUJ0	√	√	√
INTTAUJ0I2	WUF20[6]	WUFMSK20[6]	WUFC20[6]	TAUJ0	√	√	√
INTTAUJ0I3	WUF20[7]	WUFMSK20[7]	WUFC20[7]	TAUJ0	√	√	√
INTRLIN31	WUF20[8]	WUFMSK20[8]	WUFC20[8]	RLIN31	√	√	√
INTRLIN32	WUF20[9]	WUFMSK20[9]	WUFC20[9]	RLIN32	√	√	√
INTRTCA01S	WUF20[10]	WUFMSK20[10]	WUFC20[10]	RTCA0	√	√	√
INTRTCA0AL	WUF20[11]	WUFMSK20[11]	WUFC20[11]	RTCA0	√	√	√
INTRTCA0R	WUF20[12]	WUFMSK20[12]	WUFC20[12]	RTCA0	√	√	√
INTRLIN33	WUF20[13]	WUFMSK20[13]	WUFC20[13]	RLIN33	√	√	√
INTRLIN34	WUF20[14]	WUFMSK20[14]	WUFC20[14]	RLIN34	√	√	√
INTRLIN35	WUF20[15]	WUFMSK20[15]	WUFC20[15]	RLIN35	√	√	√

12.1.3 On-Chip Debug Wake-Up

The On-Chip Debug unit (OCD) generates a wake-up event while the microcontroller runs the application program in the following cases:

- The debugger issues a stop request
- A breakpoint is hit

In either case all stand-by modes are terminated, provided the OCD debug event is enabled as a wake-up factor via the WUFMSK0 register.

CAUTION

If the OCD wake-up event is disabled, it is not possible to wake up from stand-by modes via an On-chip debugger request.

The OCD wake-up event can be enabled as a wake-up factor for all stand-by modes by setting WUFMSK0[31] = 0.

When the hot plug-in function is used, make sure to enable the OCD wake-up event and return from stand-by mode by INTDCUTDI interrupt.

12.1.4 I/O Buffer Control

This section describes the behavior of the I/O buffers during various stand-by modes.

The port groups in the Isolated area support the I/O buffer hold state.
The port groups in the Always-ON area remain their state before entering stand-by mode.
For details on the port group assignment to the Isolated area and to the Always-ON area, see **Section 39, Power Supply and Power Domains**.

12.1.4.1 I/O Buffer Hold State

During the I/O buffer hold state, the I/O buffers maintains the state it was in before entering this state. Therefore, no external or internal signal can change the state of the I/O buffer until the I/O buffer hold state is terminated.

12.1.4.2 I/O Buffers during STOP Mode

In STOP mode, the I/O buffers remain their state before entering STOP mode (I/O buffer hold state is not entered).

12.1.4.3 I/O Buffers during DeepSTOP Mode

In DeepSTOP mode, the I/O buffers of port groups in the Isolated area transition to I/O buffer hold state.

After wake-up from DeepSTOP mode, the I/O buffers remain in I/O buffer hold state until cancel the state by software. To cancel I/O buffer hold state, follow the steps shown below.

1. Re-configure the peripheral or port function.
2. Set IOHOLD.IOHOLD = 0.

The following table is a summary of the I/O buffer in Isolated area during standby mode and after wakeup.

Table 12.5 Buffer Operation during Standby Mode and after WakeUp (I/O Buffers in the Isolated Area)

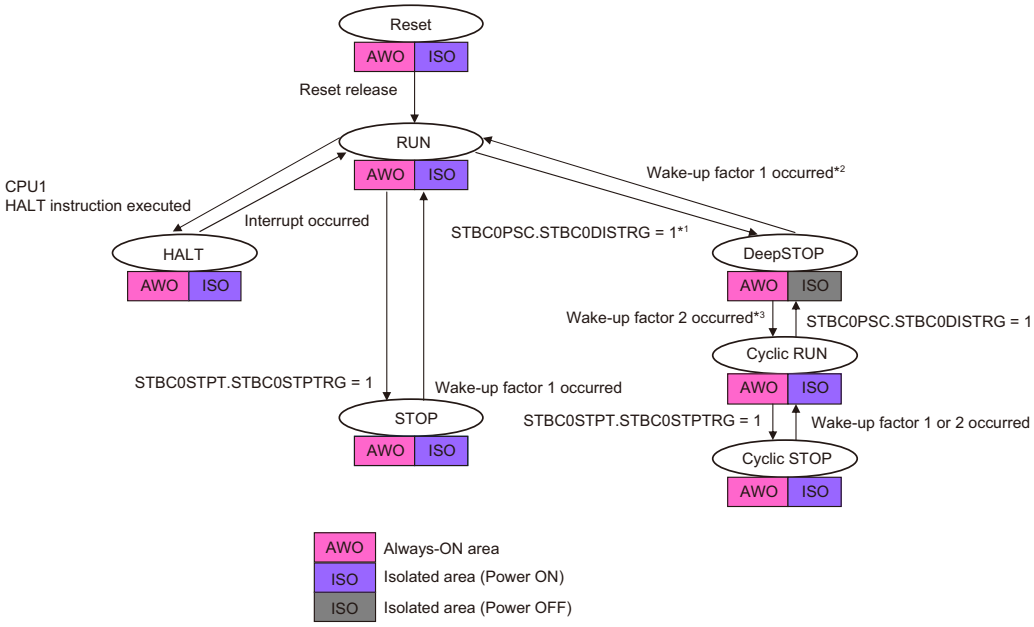
	Before Standby	During Standby	After Wakeup
STOP mode	Normal operation		
DeepSTOP mode	Normal operation	I/O buffer hold state	I/O buffer hold state *1

Note 1. Set the IOHOLD.IOHOLD bit to “0” to release the I/O buffer hold state.

The port groups in the Always-ON area don't support I/O buffer hold state. They continue operation and remain its state before entering DeepSTOP mode. In the case an alternative function of modules in Isolated area is assigned to the pin in the Always-ON area, the state of the I/O buffer may change in the transition to the DeepSTOP mode due to initialization of the modules in the Isolated area by ISORES. To avoid this behavior, it is recommended to change to function of modules in Always-ON area (e. g. Port mode) before entering DeepSTOP mode.

12.1.5 Transition to Stand-By Mode

The figure below shows transition between RUN mode and stand-by mode.



Note 1. Set a clock other than the PLL clock as the CPU clock source.
Note 2. To return from DeepSTOP mode to RUN mode, ISO peripheral function registers must be set again. The CPU1 starts the operation from the reset vector address on the code flash.
Note 3. The CPU1 starts the operation from the reset vector address on the retention RAM.

Figure 12.1 Transition to Stand-By Mode

12.1.6 Clock Supply

The clock supply to the standby controller is shown in the following table.

Table 12.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
Standby controller	Register access	EMCLK

12.2 Registers

12.2.1 List of Registers

The following table lists the stand-by controller registers.

Table 12.7 List of Registers

Module Name	Register Name	Symbol	Address
STBC	Power save control register	STBC0PSC	FFF8 0100_H
	Power stop trigger register	STBC0STPT	FFF8 0110_H
	Wake-up factor registers	WUF0	FFF8 0400_H
		WUF20	FFF8 0520_H
		WUF_ISO0	FFF8 8110_H
	Wake-up factor mask registers	WUFMSK0	FFF8 0404_H
		WUFMSK20	FFF8 0524_H
		WUFMSK_ISO0	FFF8 8114_H
	Wake-up factor clear registers	WUFC0	FFF8 0408_H
		WUFC20	FFF8 0528_H
		WUFC_ISO0	FFF8 8118_H
	I/O buffer hold control register	IOHOLD	FFF8 $0B00_H$

12.2.2 Details of Stand-By Controller Control Registers

12.2.2.1 STBC0PSC — Power Save Control Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 0100_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STBC0DISTRG	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 12.8 STBC0PSC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	STBC0DISTRG	0: No effect 1: Transition to DeepSTOP mode
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

12.2.2.2 STBC0STPT — Power Stop Trigger Register

The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 0110_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STBC0STPTRG
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.9 STBC0STPT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	STBC0STPTRG	0: No effect 1: Transition to STOP mode – In RUN mode: Transition to STOP mode – In Cyclic RUN mode: Transition to Cyclic STOP mode

12.2.2.3 WUF0/WUF20/WUF_ISO0 — Wake-Up Factor Registers

These registers indicate the generation of wake-up factors.

WUF0 and WUF20 are initialized by all reset sources except the transition to DeepSTOP mode (AWORES). WUF_ISO0 is initialized by all reset sources (ISORES).

Access: These registers are read-only registers that can be read in 32-bit units.

Address: WUF0: FFF8 0400$_H$
WUF20: FFF8 0520$_H$
WUF_ISO0: FFF8 8110$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WUF31	WUF30	WUF29	WUF28	WUF27	WUF26	WUF25	WUF24	WUF23	WUF22	WUF21	WUF20	WUF19	WUF18	WUF17	WUF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUF15	WUF14	WUF13	WUF12	WUF11	WUF10	WUF09	WUF08	WUF07	WUF06	WUF05	WUF04	WUF03	WUF02	WUF01	WUF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.10 WUF0/WUF20/WUF_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFy	Indicates the generation of a wake-up factor. 0: Wake-up factor is not generated 1: Wake-up factor is generated

NOTE

While the WUFMSKy bit in the wake-up factor mask register is 1, WUFy is not set to 1 at the generation of a wake-up factor.

Wake-Up Factors

As for the assignment of wake-up factors to the wake-up factor register bits, see **Table 12.3, Wake-Up Factor 1 Register Assignment** and **Table 12.4, Wake-Up Factor 2 Register Assignment**.

The bit to which a wake-up factor is not assigned is read as the value “0”.

12.2.2.4 WUFMSK0/WUFMSK20/WUFMSK_ISO0 — Wake-Up Factor Mask Registers

These registers enable wake-up factors.

WUFMSK0 and WUFMSK20 are initialized by all reset sources except the transition to DeepSTOP mode (AWORES). WUFMSK_ISO0 is initialized by all reset sources (ISORES).

Access: These registers can be read or written in 32-bit units.

Address: WUFMSK0: FFF8 0404_H
WUFMSK20: FFF8 0524_H
WUFMSK_ISO0: FFF8 8114_H

Value after reset: WUFMSK0: FFFF $FFFF_H$
WUFMSK20: FFFF $FFFF_H$
WUFMSK_ISO0: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WUFMSK31	WUFMSK30	WUFMSK29	WUFMSK28	WUFMSK27	WUFMSK26	WUFMSK25	WUFMSK24	WUFMSK23	WUFMSK22	WUFMSK21	WUFMSK20	WUFMSK19	WUFMSK18	WUFMSK17	WUFMSK16
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUFMSK15	WUFMSK14	WUFMSK13	WUFMSK12	WUFMSK11	WUFMSK10	WUFMSK09	WUFMSK08	WUFMSK07	WUFMSK06	WUFMSK05	WUFMSK04	WUFMSK03	WUFMSK02	WUFMSK01	WUFMSK00
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.11 WUFMSK0/WUFMSK20/WUFMSK_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFMSKy	Enables/disables a wake-up factor. 0: Wake-up factor is enabled 1: Wake-up factor is disabled

NOTE

While the WUFMSKy bit is 1, WUFy of the wake-up factor register is not set to 1 at the generation of a wake-up factor.

Wake-Up factors

As for the assignment of wake-up factors to the wake-up factor register bits, see **Table 12.3, Wake-Up Factor 1 Register Assignment** and **Table 12.4, Wake-Up Factor 2 Register Assignment**.

When writing to these registers, write the value “1” to the bits to which wake-up factors are not assigned.

12.2.2.5 WUFC0/WUFC20/WUFC_ISO0 — Wake-Up Factor Clear Registers

These registers clear the WUFy bits in the wake-up factor registers.

Access: These registers are write-only registers that can be written in 32-bit units.

Address: WUFC0: FFF8 0408$_H$
WUFC20: FFF8 0528$_H$
WUFC_ISO0: FFF8 8118$_H$

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WUFC 31	WUFC 30	WUFC 29	WUFC 28	WUFC 27	WUFC 26	WUFC 25	WUFC 24	WUFC 23	WUFC 22	WUFC 21	WUFC 20	WUFC 19	WUFC 18	WUFC 17	WUFC 16
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUFC 15	WUFC 14	WUFC 13	WUFC 12	WUFC 11	WUFC 10	WUFC 09	WUFC 08	WUFC 07	WUFC 06	WUFC 05	WUFC 04	WUFC 03	WUFC 02	WUFC 01	WUFC 00
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 12.12 WUFC0/WUFC20/WUFC_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFCy	Clears the wake-up factor bit WUFy in the wake-up factor registers. 0: WUFy is not modified 1: WUFy is cleared

Wake-Up Factors

As for the assignment of wake-up factors to the wake-up factor register bits, see **Table 12.3, Wake-Up Factor 1 Register Assignment** and **Table 12.4, Wake-Up Factor 2 Register Assignment**.

When writing to these registers, write the value “0” to the bits to which wake-up factors are not assigned.

12.2.2.6 IOHOLD — I/O Buffer Hold Control Register

This register specifies the hold state of the I/O buffer in DeepSTOP mode. The correct write sequence using the PROTCMD0 register is required in order to update this register. For details, see **Section 5, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (AWORES).

Access: This register can be read or written in 32-bit units.

Address: FFF8 0B00$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IOHOLD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.13 IOHOLD Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	IOHOLD	0: I/O hold state is released 1: I/O hold state This bit is automatically set to 1 at the transition to DeepSTOP mode. Setting this bit to 1 by software is prohibited. To release the I/O hold state after the wake-up, set this bit to 0 by software.

12.3 Mode Transition

This section explains the mode transition procedures.

12.3.1 STOP Mode

In STOP mode, most of the clock supplies to the Always-ON area and the Isolated area are stopped, The clock stop mask registers control clock supply to related clock domains in stand-by mode. Stop all of the peripheral functions before transition to STOP mode if the clock supply to the function will be stopped in STOP mode.

The transition procedure (example) to STOP mode is shown below.

Preparation for STOP mode

- Stop all of the peripheral functions to which the clock supply is to be stopped.
- Disable the interrupt handling by issuing the CPU instruction “DI”.
- Set the interrupt control registers.
  - Clear the interrupt flag (ICxxx.RFxxx = 0).
  - Mask the interrupts for non-wake-up factors (ICxxx.MKxxx = 1).
  - Release the masks of the interrupts for wake-up factors (ICxxx.MKxxx = 0).
- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUF0/WUFC_ISO0 registers).
  - Mask the non-wake-up factor (the WUFMSK0/WUFMSK_ISO0 registers).
  - Release the masks of the wake-up factors (the WUFMSK0/WUFMSK_ISO0 registers).
- Set the clock stop mask register to select the clock domains to be stopped and the ones to continue operating (using the CKSC_xxx_STPM.xxxxSTPMSK bit).
- Specify whether to oscillate or stop each clock source. In addition, set the clock stop mask register to select the clock sources to be stopped and the ones to continue operating (using the MOSCSTPMSK bit in the MOSCSTPM register and the ROSCSTPMSK bit in the ROSCSTPM register).

Start of STOP mode

Set the STBC0STPTRG bit in the STBC0STPT register to 1 to transition to STOP mode.

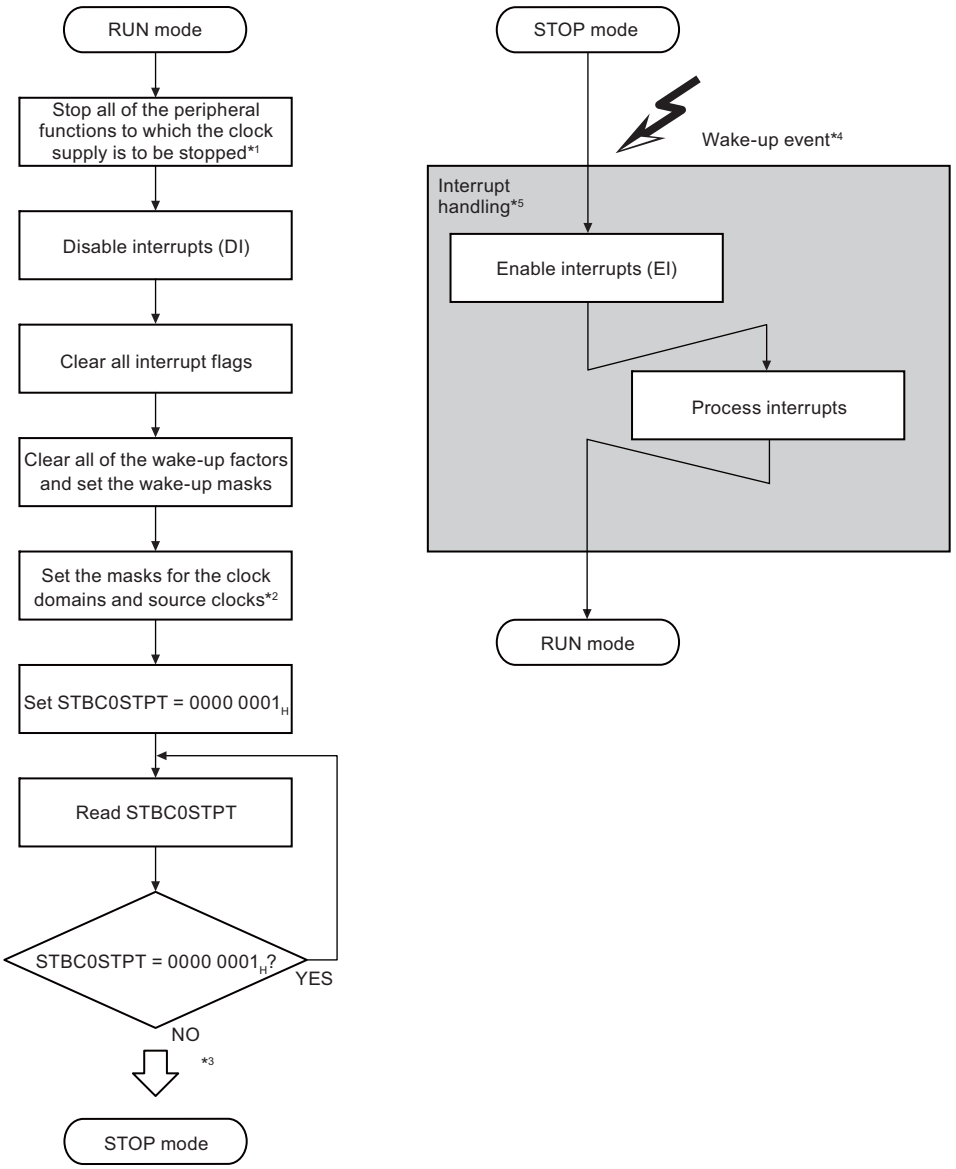
End of STOP mode

When a wake-up factor is generated, the microcontroller returns from STOP mode.

Wake-up handling

The generation of the wake-up factors can be determined by the wake-up factor flags (WUF0, WUF_ISO0).

When an interrupt is enabled by the CPU instruction “EI”, the generated wake-up interrupt will be executed.



Note 1. When the operation of the peripheral function is stopped during operating due to the transition to the STOP mode, the operation of the peripheral function may be incorrect. Before the transition to the STOP mode, stop all of the peripheral functions to which the clock supply is to be stopped.

Note 2. Though the clock mask can be set before this flow is started, it must be set before $0000\ 0001_H$ is written to STBC0STPT.

Note 3. The clock supply to the CPU is stopped and the operation transitions to the STOP mode while checking that STBC0STPT = $0000\ 0001_H$.

Note 4. STBC0STPT is set to $0000\ 0000_H$ at the generation of a wake-up factor. The generated wake-up factor can be checked by the WUF0 and WUF_ISO0 registers.

Note 5. This processing is optional. It is required when executing the interrupt handling after the wake-up.

Figure 12.2 Example of STOP Mode Transition

12.3.2 DeepSTOP Mode

In DeepSTOP mode, the clock supply to all areas and the power supply to the Isolated area are stopped.

However, clock supply to the peripheral functions in the AWO (Always-ON) area can be continued by setting the clock stop mask register.

Select the clock other than the PLL as the CPU operating clock, before the transition to DeepSTOP mode.

The transition procedure (example) to DeepSTOP mode is shown below.

Preparation for DeepSTOP mode

- Stop all of the peripheral functions to which the clock supply is to be stopped.
- Disable the interrupt handling by issuing the CPU instruction “DI”.
- Set the interrupt control registers.
  - Clear the interrupt flag (ICxxx.RFxxx = 0).
  - Mask the interrupts for non-wake-up factors (ICxxx.MKxxx = 1).
  - Release the masks of the interrupts for wake-up factors (ICxxx.MKxxx = 0).
- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUF0 and WUFC20 registers).
  - Mask the non-wake-up factor (the WUFMSK0 and WUFMSK20 registers).
  - Release the masks of the wake-up factors (the WUFMSK0 and WUFMSK20 registers).

CAUTION

When a wake-up factor is assigned to both wake-up factor 1 registers and wake-up factor 2 registers, it can be used only in one of them.

- Set the clock stop mask register to select the clock domains to be stopped and the ones to continue operating (using the CKSC_xxx_STPM.xxxxSTPMSK bit).
- Specify whether to oscillate or stop each clock source. In addition, set the clock stop mask register to select the clock sources to be stopped and the ones to continue operating (using the MOSCSTPMSK bit in the MOSCSTPM register and the ROSCSTPMSK bit in the ROSCSTPM register).

Start of DeepSTOP mode

Set the STBC0DISTRG bit in the STBC0PSC register to 1 to transition to DeepSTOP mode.

End of DeepSTOP mode

When a wake-up factor is generated, the microcontroller returns from DeepSTOP mode.

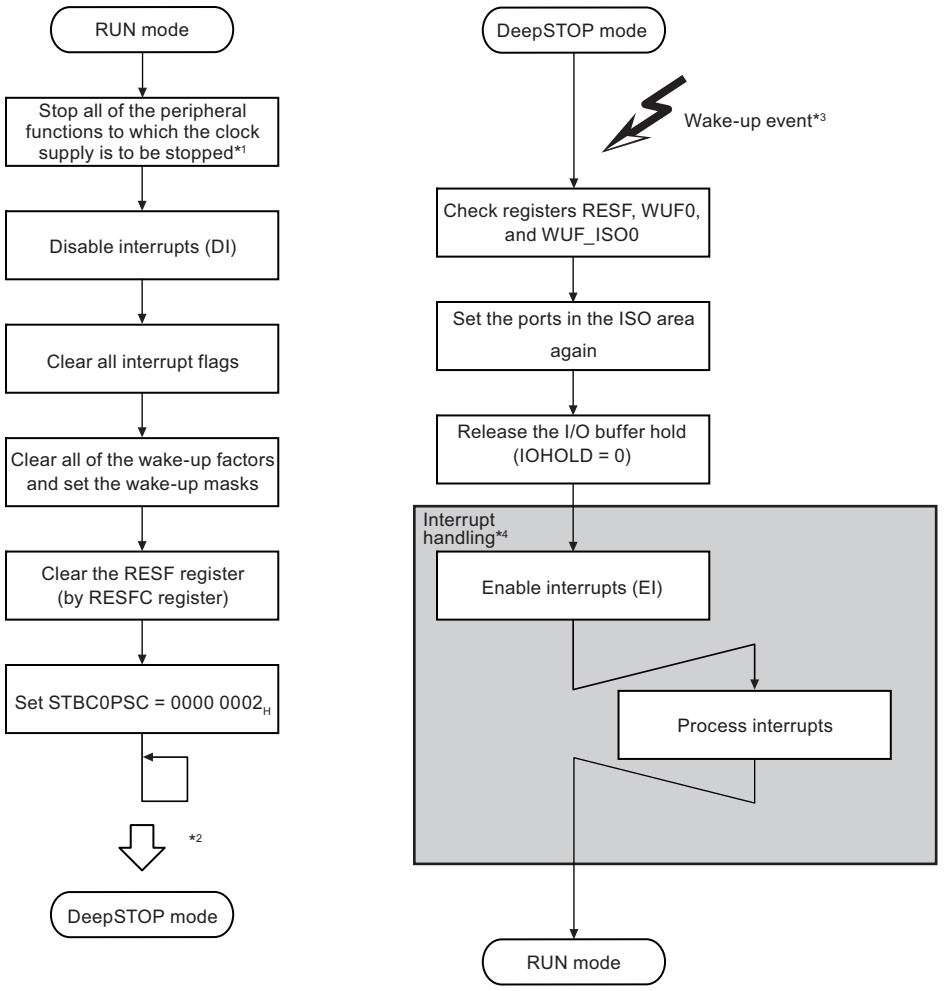
Wake-up handling

- When returned from DeepSTOP mode due to wake-up factor 1, the microcontroller starts the operation from the reset vector address.
  If one of the following interrupts has been generated before recovery from DeepSTOP mode to RUN mode, the microcontroller restart operation from the exception handler address:
  - FENMI: FENMI handler address ($E0_H$)
  - FEINT: FEINT handler address ($F0_H$)

  Note that the General-purpose registers and Local RAM are undefined value after return from DeepSTOP mode.
- The generation of the wake-up factors can be determined by the wake-up factor flags (WUF0).
- The ports in the Isolated area maintain the I/O buffer hold state.
  Release the I/O buffer hold state by executing the following steps:

  1. Re-configure the peripheral functions and port functions.

  2. Set IOHOLD.IOHOLD = 0.
- To execute an interrupt of the wake-up factor after the wake-up, evaluate the information of wake-up factor flag by software and set the interrupt request flag in the interrupt control register. Then, when an interrupt is enabled by the CPU instruction “EI”, the generated wake-up interrupt will be executed.



Note 1. When the operation of the peripheral function is stopped during operating due to the transition to the DeepSTOP mode, the operation of the peripheral function may be incorrect. Before the transition to the DeepSTOP mode, stop all of the peripheral functions to which the clock supply is to be stopped.

Note 2. After setting STBC0PSC = 0000 0002_H, wait for the transition to the DeepSTOP mode by the unconditional loop.

Note 3. The CPU1 starts the program from the reset vector after the generation of a wake-up factor. The return from the DeepSTOP mode by a reset can be checked by the RESF register. In addition, the generated wake-up factor can be checked by the WUF0 and WUF_ISO0 registers.

Note 4. This processing is optional. It is required when executing the interrupt handling after the wake-up.

Figure 12.3 Example of DeepSTOP Mode Transition

12.3.3 Cyclic RUN Mode

In Cyclic RUN mode, the functions except the CPU1, AWO area peripheral functions, and RLIN3 are stopped. In this mode, PLL and Flash Memory are not available. The transition procedure (example) to Cyclic RUN mode is shown below.

Preparation of Cyclic RUN mode

Allocate the program for Cyclic RUN mode to the Retention RAM. The reset vector base address in Cyclic RUN mode is set to the first address of the Retention RAM (FEF0 0000_H).

The instruction to transition to DeepSTOP mode should be arranged in the interrupt exception handler or a polling routine of interrupt request which is used as the source of returning to the RUN mode.

For details on the exception vector, see the *RH850G3M User's Manual: Software*.

CAUTION

Do not change the PSW.EBV bit from its value after reset in Cyclic RUN mode (Do not set the PSW.EBV bit to 1 in Cyclic RUN mode).

- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUFC20 register).
  - Mask the non-wake-up factor (the WUFMSK20 register).
  - Release the masks of the wake-up factors (the WUFMSK20 register).
- Transition to DeepSTOP mode. For details on the transition to DeepSTOP mode, see **Section 12.3.2, DeepSTOP Mode**.

Start of Cyclic RUN mode

The operation transitions to Cyclic RUN mode from DeepSTOP mode at the generation of wake-up factor 2. The microcontroller starts operation from the reset vector address of Cyclic RUN mode (the first address of the Retention RAM (FEF0 0000_H)).

If one of the following interrupts has been generated during recovery from DeepSTOP mode to Cyclic RUN mode, the microcontroller restart operation from the exception handler address:

- FENMI: FENMI handler address in Cyclic RUN mode (FEF0 0000_H + $E0_H$)
- FEINT: FEINT handler address in Cyclic RUN mode (FEF0 0000_H + $F0_H$)

Note that the General-purpose registers and Local RAM are undefined value after the transition to Cyclic RUN mode from DeepSTOP mode.

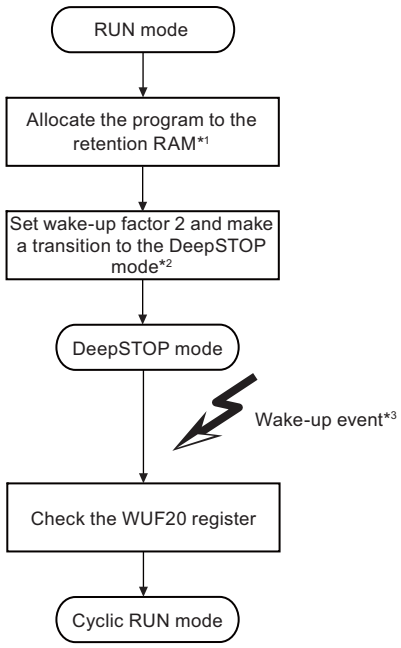
The operation transitions to Cyclic RUN mode from Cyclic STOP mode at the generation of wake-up factors 1 and 2.

End of Cyclic RUN mode

The Cyclic RUN mode ends at the transition to the Cyclic STOP mode by setting the STBC0STPT.STBC0STPTRG bit to 1, or at the transition to the DeepSTOP mode by setting the STBC0PSC.STBC0DISTRG bit to 1.

Wake-up handling

The generation of the wake-up factors can be determined by the wake-up factor flags (WUF20).



Note 1. When the mode transitions from the Cyclic RUN mode to the RUN mode by a wake-up factor, the transition to the DeepSTOP mode should be made in the processing of the interrupt vector for the wake-up factor. In that case, allocate the interrupt processing program to the retention RAM as well.

Note 2. Before the transition to the DeepSTOP mode, clear the flag for wake-up factor 2 in the WUFC20 register and set wake-up factor 2 that is to be used in the WUFMSK20 register. Other parts of the procedure for transition to DeepSTOP mode are the same as the normal procedure.

Note 3. The CPU1 starts the program from the top address (FEF0 0000_H) of the retention RAM after the generation of a wake-up factor. The generated wake-up factor can be checked by the WUF20 register.

Figure 12.4 Example of Cyclic RUN Mode Transition

12.3.4 Cyclic STOP Mode

In Cyclic STOP mode, the functions except the AWO area peripheral function and RLIN3 are stopped.

The transition procedure (example) to Cyclic STOP mode is shown below.

Preparation for Cyclic STOP mode

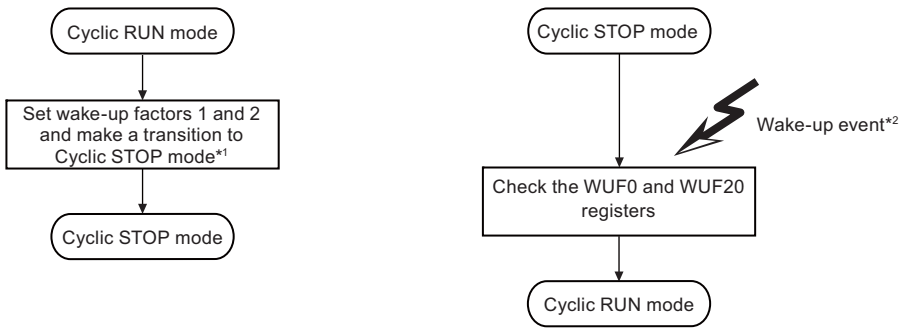
- Transition to Cyclic RUN mode.
- Set the wake-up related registers.
  - Clear the wake-up factor flags (the WUF0/WUFC20 register).
  - Mask the non-wake-up factor (the WUFMSK0/WUFMSK20 register).
  - Release the masks of the wake-up factors (the WUFMSK0/WUFMSK20 register).

Start of Cyclic STOP mode

Set the STBC0STPT.STBC0STPTR bit to 1 to transition to Cyclic STOP mode.

End of Cyclic STOP mode

The operation transitions to Cyclic RUN mode at the generation of wake-up factor 1 or 2.



Note 1. Set wake-up factors 1 and 2 to be used for transitioning to RUN mode and Cyclic RUN mode, respectively. When the mode transitions to RUN mode by wake-up factor 1, the transition processing to DeepSTOP mode should be added in Cyclic RUN mode.

Note 2. When a wake-up factor is generated in Cyclic STOP mode, the mode transitions to Cyclic RUN mode and the operation starts immediately after the processing shifted to Cyclic STOP mode. The generated wake-up factors can be checked by the WUF0 and WUF20 registers.

Figure 12.5 Example of Cyclic STOP Mode Transition

12.4 Writing to the Stand-By Controller Related Registers

The following stand-by controller registers are write-protected registers.

- STBC0PSC register
- STBC0STPT register
- IOHOLD register

The write-protected registers are protected against the illegal writing due to an incorrect program operation.

For details on the write-protected sequence, see **Section 5, Write-Protected Registers**.

12.5 Cautions when Using Stand-By Modes

12.5.1 Cautions Concerning Transitioning to DeepSTOP Mode When Using a Debugger

When using a debugger, executing a program that causes the mode to transition to DeepSTOP mode immediately after the program is started may cause improper communication between the OCD emulator and microcontroller because the microcontroller will enter DeepSTOP mode before the preparations for communication between the OCD emulator and microcontroller are completed.

The communication preparation period depends on the OCD emulator's host PC environment and the operating frequency of the microcontroller, so when performing debugging that causes the program to enter DeepSTOP mode immediately after the program starts, insert a wait between reset release and the DeepSTOP execution instruction so that the debugger starts normally.

In DeepSTOP mode, the debugging controller stops. For return from DeepSTOP mode by the debugger, see **Section 12.1.3, On-Chip Debug Wake-Up**.

Section 13 Low-Power Sampler (LPS)

This section contains a generic description of the low-power sampler (LPS).

The first part in this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the LPS.

13.1 Features of RH850/F1M LPS

13.1.1 Number of Units

This microcontroller has the following number of LPS units.

Table 13.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	1		
Name	LPSn (n = 0)		

Table 13.2 Unit Configurations and Channels

Unit Name LPSn	Channels per Unit	Function	Channel Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
LPS0	1	Digital port input m for port polling	DPINm	24 ch		
		Analog input m for A/D converter	ADCA0Im	16 ch		

Table 13.3 Indices

Index	Description
n	Throughout this section, the individual LPS units are identified by the index “n” (n = 0).
m	Throughout this section, the number of digital port input channels for LPS port polling is indicated by the index “m” (m = 0 to 23) and the number of analog input channels for A/D converter is indicated by the index “m” (m = 0 to 15)
k	The external multiplexer select output signal for digital port is indicated by the index “k”.
x	LPS sequence start trigger input signal is indicated by the index “x”.

NOTE

Descriptions of functions and registers in this section are based on the maximum configurations. Adjust the indices in the text to the proper value for each product. When writing a value to a register that will result in writing to bits outside the range of the index for the product you are using, write the value after reset to these bits.

The following table shows the values indicated by the indices of each product.

Table 13.4 Indices of Products

Indices of Each Product		
144 pins	176 pins	233 pins
k = 0 to 2		
x = 0 to 3		

13.1.2 Register Base Address

The LPS base address is shown in the following table.

LPS register addresses are given as offsets from the base address.

Table 13.5 Register Base Address

Base Address Name	Base Address
<LPS0_base>	FFF8 3000_H

13.1.3 Clock Supply

The LPS clock supply is shown in the following table.

If the operation request signal for the low-power sampler (LPS) is at the active level, the clock for clock domains for which the HS IntOSC is selected also operates.

To stop the function of a clock domain, set the target clock domain to “disabled” before making a transition to standby mode.

Table 13.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
LPSn	Register access	EMCLK

13.1.4 Interrupt Request

The LPS interrupt requests are listed in the following table.

Table 13.7 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
LPS0			
INTCWEND	Port polling end interrupt (LPS)	112	—
INTADCA0I0*1	ADCA0 SG1 end interrupt	18	4
INTADCA0I1*1	ADCA0 SG2 end interrupt	19	5
INTADCA0I2*1	ADCA0 SG3 end interrupt	20, 32	6

Note 1. These signals are output from ADCA0.

13.1.5 Reset Sources

The LPS reset sources are shown in the following table. LPS is initialized by the reset source.

Table 13.8 Reset Sources

Unit Name	Reset Source
LPS0	All reset sources except transition to DeepSTOP mode (AWORES)

13.1.6 External Input/Output Signals

External input/output signals of LPS are listed below.

Table 13.9 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
LPS0		
DPO	Port output signal for digital input	DPO
DPSELk	External multiplexer select output signal for digital port	SELDPk
DPINm	Digital port input signal	DPINm
APO	Port output signal for analog input	APO
ADCA0Im	ADCA input channel signal	ADCA0Im

CAUTION

When the P0_0 pin is used as DPO, note that the port P0_0 outputs low-level as $\overline{\text{RESETOUT}}$ function during reset and after release from the reset.

For details, see **Section 2.11.1.1, P0_0: $\overline{\text{RESETOUT}}$**.

13.1.7 Internal Input/Output Signals

Internal input/output signals for connecting the LPS and STBC or the LPS and TAUJ are listed below.

Table 13.10 Internal Input/Output Signals

Unit Signal Name	Description	Connected to
WUTRG0	LPS wake-up source trigger 0 output signal	STBC
WUTRG1	LPS wake-up source trigger 1 output signal	STBC
INTTAUJ0Ix	LPS sequence start trigger x input signal	TAUJ0

13.2 Overview

13.2.1 Functional Overview

To monitor the external input without consuming CPU resources, the low-power sampler (LPS) can check the digital input ports and analog input ports without using the CPU. The figure below shows a connection example between the main components of the LPS and the external circuit.

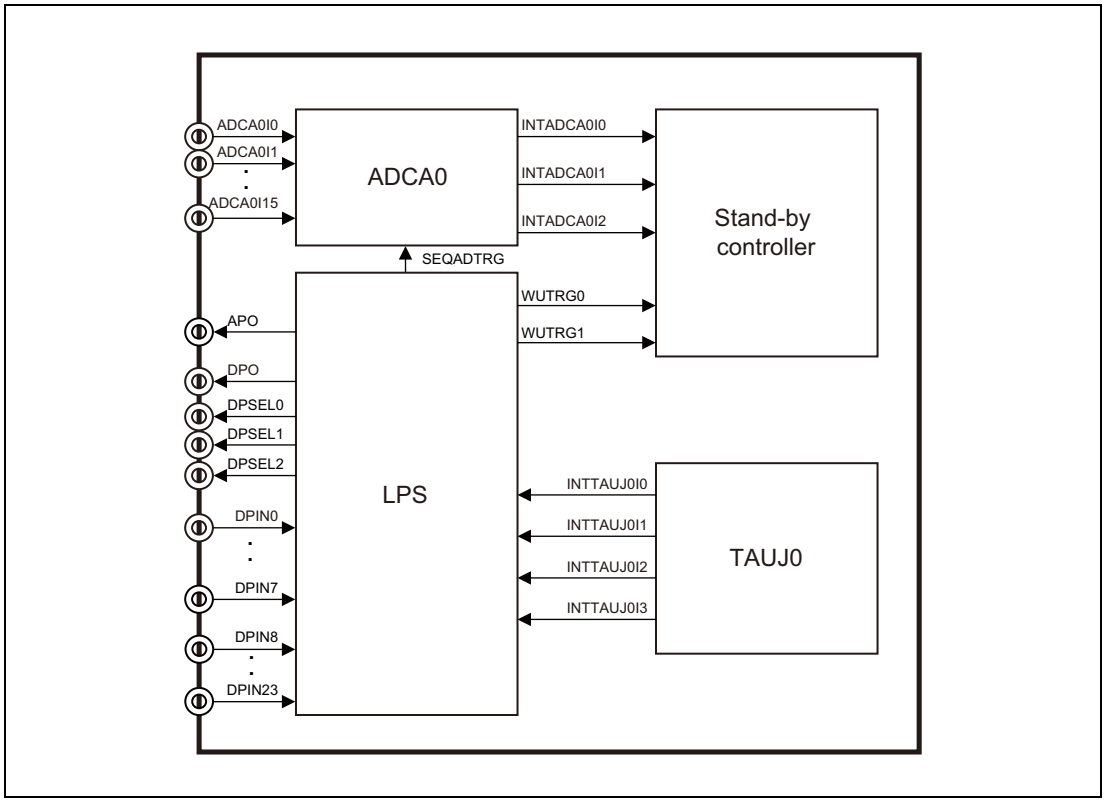


Figure 13.1 Block Diagram of the LPS

CAUTION

DPSEL2 to DPSEL0 are assigned to the same pins as DPIN10 to DPIN8 as alternative-function. They cannot be used simultaneously.

13.3 Registers

13.3.1 List of Registers

LPS registers are listed in the following table.

For details about <LPS0_base>, see **Section 13.1.2, Register Base Address**.

Table 13.11 List of Registers

Module Name	Register Name	Symbol	Address
—	LPS control register	SCTLR	<LPS0_base > + 00_H
—	Event flag register	EVFR	<LPS0_base > + 04_H
—	DPIN select register 0	DPSELR0	<LPS0_base > + 08_H
—	DPIN select register M	DPSELRM	<LPS0_base > + $0C_H$
—	DPIN select register H	DPSELRH	<LPS0_base > + 10_H
—	DPIN data set register 0	DPDSR0	<LPS0_base > + 14_H
—	DPIN data set register M	DPDSRM	<LPS0_base > + 18_H
—	DPIN data set register H	DPDSRH	<LPS0_base > + $1C_H$
—	DPIN data input monitor register 0	DPDIMR0	<LPS0_base > + 20_H
—	DPIN data input monitor register 1	DPDIMR1	<LPS0_base > + 24_H
—	DPIN data input monitor register 2	DPDIMR2	<LPS0_base > + 28_H
—	DPIN data input monitor register 3	DPDIMR3	<LPS0_base > + $2C_H$
—	DPIN data input monitor register 4	DPDIMR4	<LPS0_base > + 30_H
—	DPIN data input monitor register 5	DPDIMR5	<LPS0_base > + 34_H
—	DPIN data input monitor register 6	DPDIMR6	<LPS0_base > + 38_H
—	DPIN data input monitor register 7	DPDIMR7	<LPS0_base > + $3C_H$
—	Count value register	CNTVAL	<LPS0_base > + 40_H
—	LPS operation status register	SOSTR	<LPS0_base > + 44_H

13.3.2 SCTLR — LPS Control Register

This register is used to configure the LPS.

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	NUMDP2	NUMDP1	NUMDP0	TJIS1	TJIS0	ADEN	DPEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.12 SCTLR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	NUMDP[2:0]	These bits specify the number of times the port is read in digital input mode. If two or more times are specified, the external multiplexer is controlled by the DPSEL[2:0] pins. The bits for which comparison is enabled in the DPSELR0, DPSELRM, and DPSELRH registers are compared regardless of the repeat number setting, and WUTRG will be generated according to the results.
3, 2	TJIS[1:0]	Sequence Start Trigger Select 00: INTTAUJ0I0 01: INTTAUJ0I1 10: INTTAUJ0I2 11: INTTAUJ0I3 These bits should be set before the sequence operation is started (when the SCTLR.DPEN bit = 0, the SCTLR.ADEN bit = 0, and the SOSTR.SOF bit = 0). (When changing the SCTLR.DPEN bit and the SCTLR.ADEN bit, write the same value to these bits.)
1	ADEN	0: Analog input mode is disabled 1: Analog input mode is enabled

NUMDP[2:0]	Number of Times the Port Is Read
000_B	One time
001_B	Two times
010_B	Three times
011_B	Four times
100_B	Five times
101_B	Six times
110_B	Seven times
111_B	Eight times

These bits should be set before the TAUJ0 and sequence operations are started (when the SCTLR.DPEN bit = 0, the SCTLR.ADEN bit = 0, and the SOSTR.SOF bit = 0).
(When changing the SCTLR.DPEN bit and the SCTLR.ADEN bit, write the same value to these bits.)

Table 13.12 SCTLR Register Contents (2/2)

Bit Position	Bit Name	Function
0	DPEN	0: Digital input mode is disabled 1: Digital input mode is enabled

13.3.3 EVFR — Event Flag Register

This register indicates the result of comparing the data sequentially captured at the digital input pins and stored in the DPDIMR7 to DPDIMR0 registers with the comparison target data in the DPDSRH/DPDSRM/DPDSR0 registers.

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 04_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DINEVF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 13.13 EVFR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	DINEVF	This bit indicates the result of comparing the data captured at the digital input pins and stored in the DPDIMR7 to DPDIMR0 registers with the comparison target data in the DPDSRH/DPDSRM/DPDSR0 registers. Read: 0: The result of comparison is a match. 1: The result of comparison is a mismatch. Write: 0: Clear the bit. 1: Prohibited. This bit is set to 1 when a mismatch is detected even in one bit. Only 0 can be written to clear this bit.

13.3.4 DPSELR0 — DPIN Select Register 0

This register specifies the compare target bits in the DPDSR0 and DPDIMR0 registers.

Write to the DPSELR0 register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 08_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D0EN_23	D0EN_22	D0EN_21	D0EN_20	D0EN_19	D0EN_18	D0EN_17	D0EN_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0EN_15	D0EN_14	D0EN_13	D0EN_12	D0EN_11	D0EN_10	D0EN_9	D0EN_8	D0EN_7	D0EN_6	D0EN_5	D0EN_4	D0EN_3	D0EN_2	D0EN_1	D0EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.14 DPSELR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	D0EN_n (n = 23 to 0)	These bits enable or disable comparing each bit of the first data captured at the digital input pins and stored in the DPDIMR0 register with the comparison target data in the DPDSR0 register. 0: Disables comparison. 1: Enables comparison.

13.3.5 DPSELRM — DPIN Select Register M

This register specifies the compare target bits in the DPDSRM and DPDIMRm (m = 4 to 1) registers.

Write to the DPSELRM register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: DPSELRM can be read or written in 32-bit units.
DPSELRML and DPSELRMH can be read or written in 16-bit units.
DPSELR1, DPSELR2, DPSELR3, and DPSELR4 can be read or written in 8-bit units.

Address: DPSELRM: <LPS0_base> + $0C_H$
DPSELRML: <LPS0_base> + $0C_H$, DPSELRMH: <LPS0_base> + $0E_H$
DPSELR1: <LPS0_base> + $0C_H$, DPSELR2: <LPS0_base> + $0D_H$, DPSELR3: <LPS0_base> + $0E_H$,
DPSELR4: <LPS0_base> + $0F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D4EN_7	D4EN_6	D4EN_5	D4EN_4	D4EN_3	D4EN_2	D4EN_1	D4EN_0	D3EN_7	D3EN_6	D3EN_5	D3EN_4	D3EN_3	D3EN_2	D3EN_1	D3EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D2EN_7	D2EN_6	D2EN_5	D2EN_4	D2EN_3	D2EN_2	D2EN_1	D2EN_0	D1EN_7	D1EN_6	D1EN_5	D1EN_4	D1EN_3	D1EN_2	D1EN_1	D1EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.15 DPSELRM Register Contents

Bit Position	Bit Name	Function
31 to 24	D4EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the fifth data captured at the digital input pins and stored in the DPDIMR4 register with the comparison target data in the DPDSR4 register. 0: Disables comparison. 1: Enables comparison.
23 to 16	D3EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the fourth data captured at the digital input pins and stored in the DPDIMR3 register with the comparison target data in the DPDSR3 register. 0: Disables comparison. 1: Enables comparison.
15 to 8	D2EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the third data captured at the digital input pins and stored in the DPDIMR2 register with the comparison target data in the DPDSR2 register. 0: Disables comparison. 1: Enables comparison.
7 to 0	D1EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the second data captured at the digital input pins and stored in the DPDIMR1 register with the comparison target data in the DPDSR1 register. 0: Disables comparison. 1: Enables comparison.

13.3.6 DPSELRH — DPIN Select Register H

This register specifies the compare target bits in the DPDSRH and DPDIMRm (m = 7 to 5) registers.

Write to the DPSELRH register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: DPSELRH can be read or written in 32-bit units.
DPSELRHL and DPSELRHH can be read or written in 16-bit units.
DPSELR5, DPSELR6, and DPSELR7 can be read or written in 8-bit units.

Address: DPSELRH: <LPS0_base> + 10_H
DPSELRHL: <LPS0_base> + 10_H, DPSELRHH: <LPS0_base> + 12_H
DPSELR5: <LPS0_base> + 10_H, DPSELR6: <LPS0_base> + 11_H, DPSELR7: <LPS0_base> + 12_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D7EN_7	D7EN_6	D7EN_5	D7EN_4	D7EN_3	D7EN_2	D7EN_1	D7EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D6EN_7	D6EN_6	D6EN_5	D6EN_4	D6EN_3	D6EN_2	D6EN_1	D6EN_0	D5EN_7	D5EN_6	D5EN_5	D5EN_4	D5EN_3	D5EN_2	D5EN_1	D5EN_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.16 DPSELRH Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	D7EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the eighth data captured at the digital input pins and stored in the DPDIMR7 register with the compare target data in the DPDSR7 register. 0: Disables comparison. 1: Enables comparison.
15 to 8	D6EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the seventh data captured at the digital input pins and stored in the DPDIMR6 register with the compare target data in the DPDSR6 register. 0: Disables comparison. 1: Enables comparison.
7 to 0	D5EN_n (n = 7 to 0)	These bits enable or disable comparing each bit of the sixth data captured at the digital input pins and stored in the DPDIMR5 register with the compare target data in the DPDSR5 register. 0: Disables comparison. 1: Enables comparison.

13.3.7 DPDSR0 — DPIN Data Set Register 0

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR0 register.

Write to the DPDSR0 register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: This register can be read or written in 32-bit units.

Address: <LPS0_base> + 14_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D0_23	D0_22	D0_21	D0_20	D0_19	D0_18	D0_17	D0_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0_15	D0_14	D0_13	D0_12	D0_11	D0_10	D0_9	D0_8	D0_7	D0_6	D0_5	D0_4	D0_3	D0_2	D0_1	D0_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.17 DPDSR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 0	D0_n (n = 23 to 0)	Data to be compared with the first digital port input (DPINm)

13.3.8 DPDSRM — DPIN Data Set Register M

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR4 to DPDIMR1 registers.

Write to the DPDSRM register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: DPDSRM can be read or written in 32-bit units.
DPDSRML and DPDSRMH can be read or written in 16-bit units.
DPDSR1, DPDSR2, DPDSR3, and DPDSR4 can be read or written in 8-bit units.

Address: DPDSRM: <LPS0_base> + 18_H
DPDSRML: <LPS0_base> + 18_H, DPDSRMH: <LPS0_base> + $1A_H$
DPDSR1: <LPS0_base> + 18_H, DPDSR2: <LPS0_base> + 19_H, DPDSR3: <LPS0_base> + $1A_H$,
DPDSR4: <LPS0_base> + $1B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D4_7	D4_6	D4_5	D4_4	D4_3	D4_2	D4_1	D4_0	D3_7	D3_6	D3_5	D3_4	D3_3	D3_2	D3_1	D3_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D2_7	D2_6	D2_5	D2_4	D2_3	D2_2	D2_1	D2_0	D1_7	D1_6	D1_5	D1_4	D1_3	D1_2	D1_1	D1_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.18 DPDSRM Register Contents

Bit Position	Bit Name	Function
31 to 24	D4_n (n = 7 to 0)	Data to be compared with the fifth digital port input (DPINm)
23 to 16	D3_n (n = 7 to 0)	Data to be compared with the fourth digital port input (DPINm)
15 to 8	D2_n (n = 7 to 0)	Data to be compared with the third digital port input (DPINm)
7 to 0	D1_n (n = 7 to 0)	Data to be compared with the second digital port input (DPINm)

13.3.9 DPDSRH — DPIN Data Set Register H

This register specifies the data to be compared with the data captured at a digital input pin and stored in the DPDIMR7 to DPDIMR5 registers.

Write to the DPDSRH register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: DPDSRH can be read or written in 32-bit units.
DPDSRHL and DPDSRHH can be read or written in 16-bit units.
DPDSR5, DPDSR6, and DPDSR7 can be read or written in 8-bit units.

Address: DPDSRH: <LPS0_base> + $1C_H$
DPDSRHL: <LPS0_base> + $1C_H$, DPDSRHH: <LPS0_base> + $1E_H$
DPDSR5: <LPS0_base> + $1C_H$, DPDSR6: <LPS0_base> + $1D_H$, DPDSR7: <LPS0_base> + $1E_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D7_7	D7_6	D7_5	D7_4	D7_3	D7_2	D7_1	D7_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D6_7	D6_6	D6_5	D6_4	D6_3	D6_2	D6_1	D6_0	D5_7	D5_6	D5_5	D5_4	D5_3	D5_2	D5_1	D5_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.19 DPDSRH Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	D7_n (n = 7 to 0)	Data to be compared with the eighth digital port input (DPINm)
15 to 8	D6_n (n = 7 to 0)	Data to be compared with the seventh digital port input (DPINm)
7 to 0	D5_n (n = 7 to 0)	Data to be compared with the sixth digital port input (DPINm)

13.3.10 DPDIMR0 — DPIN Data Input Monitor Register 0

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 23)) in digital input mode. DPDIMR0 stores the data acquired for the first time.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <LPS0_base> + 20_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	D0M_23	D0M_22	D0M_21	D0M_20	D0M_19	D0M_18	D0M_17	D0M_16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0M_15	D0M_14	D0M_13	D0M_12	D0M_11	D0M_10	D0M_9	D0M_8	D0M_7	D0M_6	D0M_5	D0M_4	D0M_3	D0M_2	D0M_1	D0M_0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.20 DPDIMR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned.
23 to 0	D0M_n (n = 23 to 0)	The first digital port input (DPINm) data

13.3.11 DPDIMR1 — DPIN Data Input Monitor Register 1

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR1 stores the data acquired for the second time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 24_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D1M_7	D1M_6	D1M_5	D1M_4	D1M_3	D1M_2	D1M_1	D1M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 13.21 DPDIMR1 Register Contents

Bit Position	Bit Name	Function
7 to 0	D1M_n (n = 7 to 0)	The second digital port input (DPINm) data

13.3.12 DPDIMR2 — DPIN Data Input Monitor Register 2

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR2 stores the data acquired for the third time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 28_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D2M_7	D2M_6	D2M_5	D2M_4	D2M_3	D2M_2	D2M_1	D2M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 13.22 DPDIMR2 Register Contents

Bit Position	Bit Name	Function
7 to 0	D2M_n (n = 7 to 0)	The third digital port input (DPINm) data

13.3.13 DPDIMR3 — DPIN Data Input Monitor Register 3

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR3 stores the data acquired for the fourth time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + $2C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D3M_7	D3M_6	D3M_5	D3M_4	D3M_3	D3M_2	D3M_1	D3M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 13.23 DPDIMR3 Register Contents

Bit Position	Bit Name	Function
7 to 0	D3M_n (n = 7 to 0)	The fourth digital port input (DPINm) data

13.3.14 DPDIMR4 — DPIN Data Input Monitor Register 4

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR4 stores the data acquired for the fifth time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 30_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D4M_7	D4M_6	D4M_5	D4M_4	D4M_3	D4M_2	D4M_1	D4M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 13.24 DPDIMR4 Register Contents

Bit Position	Bit Name	Function
7 to 0	D4M_n (n = 7 to 0)	The fifth digital port input (DPINm) data

13.3.15 DPDIMR5 — DPIN Data Input Monitor Register 5

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR5 stores the data acquired for the sixth time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 34_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D5M_7	D5M_6	D5M_5	D5M_4	D5M_3	D5M_2	D5M_1	D5M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 13.25 DPDIMR5 Register Contents

Bit Position	Bit Name	Function
7 to 0	D5M_n (n = 7 to 0)	The sixth digital port input (DPINm) data

13.3.16 DPDIMR6 — DPIN Data Input Monitor Register 6

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode or MIX mode. DPDIMR6 stores the data acquired for the seventh time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 38_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D6M_7	D6M_6	D6M_5	D6M_4	D6M_3	D6M_2	D6M_1	D6M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 13.26 DPDIMR6 Register Contents

Bit Position	Bit Name	Function
7 to 0	D6M_n (n = 7 to 0)	The seventh digital port input (DPINm) data

13.3.17 DPDIMR7 — DPIN Data Input Monitor Register 7

This register stores the data which the LPS acquired from the digital port input (DPINm (m = 0 to 7)) in multiplexer mode. DPDIMR7 stores the data acquired for the eighth time.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + $3C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	D7M_7	D7M_6	D7M_5	D7M_4	D7M_3	D7M_2	D7M_1	D7M_0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 13.27 DPDIMR7 Register Contents

Bit Position	Bit Name	Function
7 to 0	D7M_n (n = 7 to 0)	The eighth digital port input (DPINm) data

13.3.18 CNTVAL — Count Value Register

This register specifies the stabilization time of the external circuits (digital signal source and analog signal source).

- In digital mode
  The time from when the DPO output is set to 1 to the time when the port input is acquired for the first time
- In analog mode
  The time from when the APO output is set to 1 to the time when the LPS outputs the A/D conversion trigger to the ADCA0

Write to the CNTVAL register before the sequence operation is started (when the SOSTR.SOF bit = 0).

CAUTION

In analog mode, make sure to secure the stabilization time longer than 1 μs for the stabilization of the A/D converter.

Access: This register can be read or written in 16-bit units.

Address: <LPS0_base> + 40_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT17	CNT16	CNT15	CNT14	CNT13	CNT12	CNT11	CNT10	CNT07	CNT06	CNT05	CNT04	CNT03	CNT02	CNT01	CNT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.28 CNTVAL Register Contents

Bit Position	Bit Name	Function
15 to 8	CNT1n (n = 7 to 0)	These bits set the stabilization time of the external circuit (analog signal source). Stabilization time = $(1/f_{RH})$ × 16 × CNT1n (set value)
7 to 0	CNT0n (n = 7 to 0)	These bits set the stabilization time of the external circuit (digital signal source). Stabilization time = $(1/f_{RH})$ × 16 × CNT0n (set value)

13.3.19 SOSTR — LPS Operation Status Register

This register indicates the operating state of the LPS.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <LPS0_base> + 44_H

Value after reset: 00_H

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

Table 13.29 SOSTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	SOF	LPS Operation Status Flag 0: Initial state before the occurrence of the start trigger 1: LPS operation is in progress (after the start trigger occurs) If the start trigger occurs while the SOF bit is set to 1 (during the LPS operation), the start trigger is canceled.

13.4 Digital Input Mode

With the digital input port DPINm and the externally connected multiplexer, up to 64 input ports can be monitored as shown in **Table 13.30, Combination of Monitored Ports**.

Port DPSELk is used to switch the external multiplexer. The DPSELk output is switched for the number of times specified in the SCTLR register.

TAUJ0 is used to set the timing to check the value input to the port.

Table 13.30 Combination of Monitored Ports

Combination (Number of Ports x Number of Checks)	Ports Used	Total Number
Direct mode When input ports are checked simultaneously without using the external multiplexer Up to 24 ports × 1	DPIN23 to DPIN0	Up to 24
Multiplexer mode When input ports are checked by using a small number of pins and the external multiplexer Up to 8 ports × 8	DPIN7 to DPIN0 DPSEL2 to DPSEL0	Up to 64
MIX mode When input ports are checked using a combination of the above two modes Up to 14 ports x 1 + Up to 7 ports x 7	DPIN7 to DPIN0 DPIN16 to DPIN11 DPSEL2 to DPSEL0*1	Up to 63

Note 1. DPIN16 to DPIN11 and DPIN7 are checked only for the first time. DPIN10 to DPIN8 cannot be used because they are shared with DPSEL2 to DPSEL0.

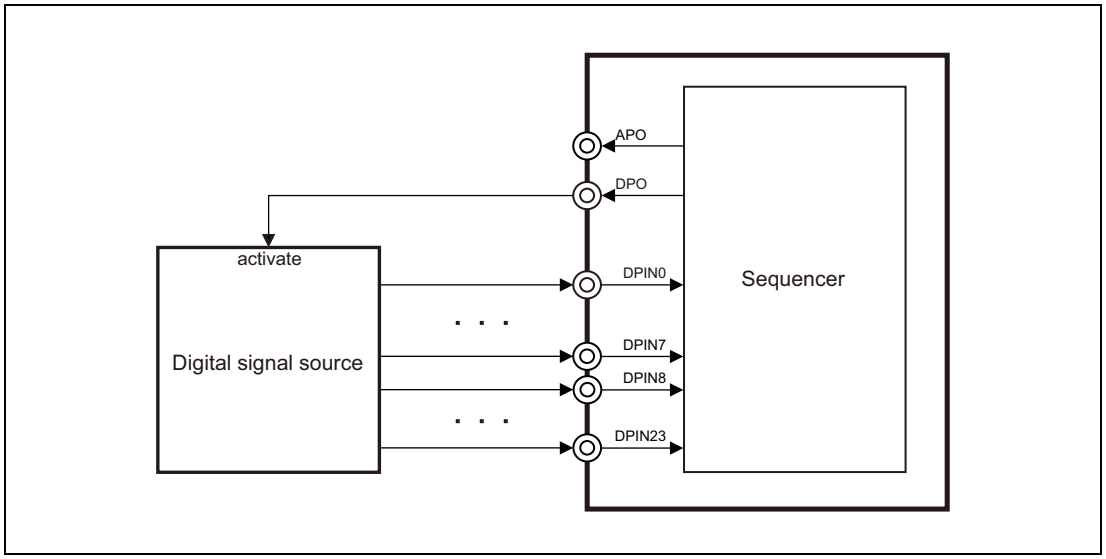


Figure 13.2 Direct Mode Connection Example

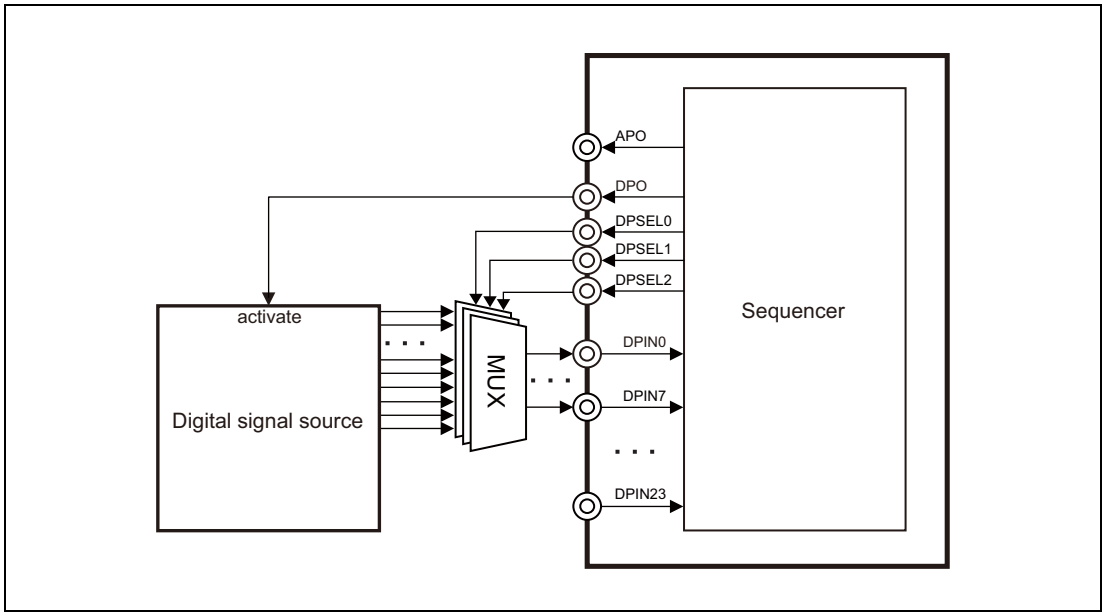


Figure 13.3 Multiplexer Mode Connection Example

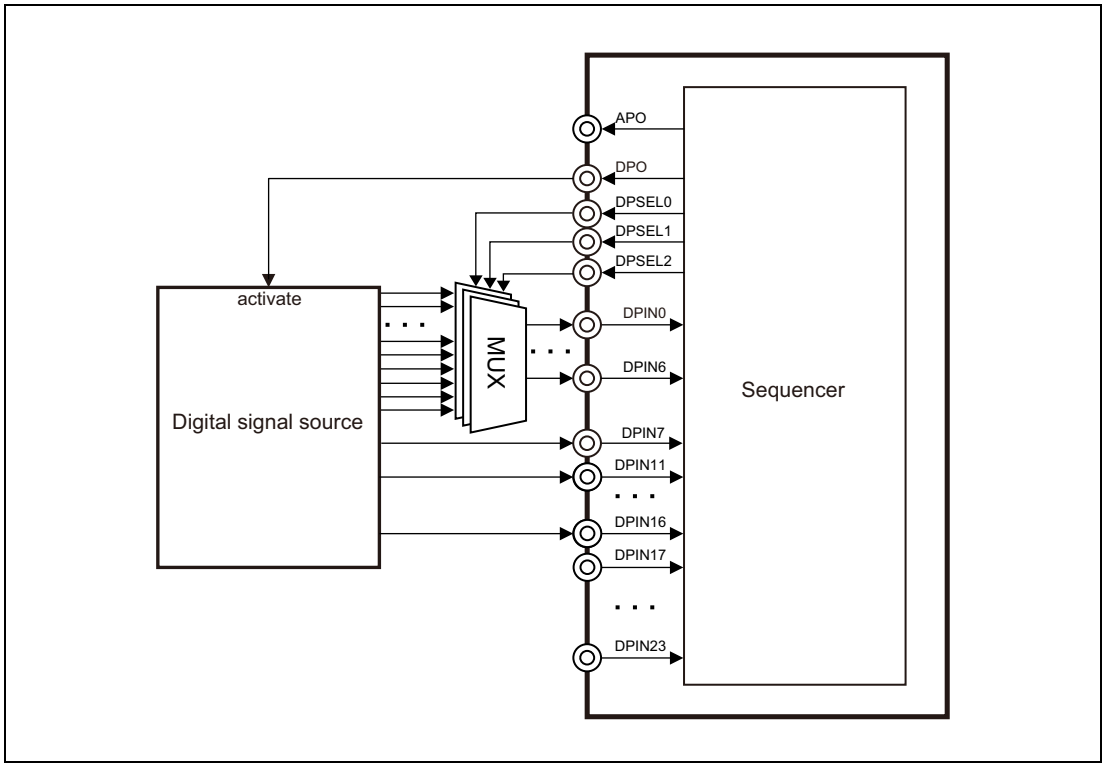


Figure 13.4 MIX Mode Connection Example

CAUTION

DPSEL2 to DPSEL0 are assigned to the same pins as DPIN10 to DPIN8 as alternative-function. They cannot be used simultaneously.

Preparation

- Set NUMDP[2:0] and TJIS[1:0] bits in the SCTLR register to specify the number of times the port is to be read, and the TAUJ0 interrupt to be used as sequence start trigger.
- Set TAUJ0 to interval timer mode.
- Set the wait time of the digital signal source by using the lower 8 bits in the CNTVAL register.
- Set expected values in the DPDSR0, DPDSRM and DPDSRH registers.
- Set the ports to be checked in the DPSELR0, DPSELRM, and DPSELRH registers.

Start

- Start the TAUJ0.
- Set the SCTLR.DPEN bit to 1.

After the operation starts, ports are checked at the interval set in TAUJ0. The operation continues regardless of whether the mode is RUN mode or power save mode. If the HS IntOSC is stopped in stand-by mode, it can only resume operation while the sequencer is running.

Upon completion of checking all ports that have been set, the INTCWEND interrupt occurs. In addition, if the input value of the port is different from the expected value set by the DPDSR0, DPDSRM, or DPDSRH register, the wake-up factor WUTRG0 occurs. The following figures show an example of the operation in digital input mode.

Stop

To stop the LPS operation in Digital Input Mode (by changing the SCTLR.DPEN bit setting from 1 to 0), follow the procedure shown below. In this example, the P0_0 pin is used as DPO.

1. Set the port register to specify low level output on the pin (P0.P0_0 = 0).
2. Change the setting for the P0_0 pin from the alternative port mode to the port mode (PMC0.PMC0_0 = 0).
3. Set SCTLR.DPEN = 0.

Note 1. The above procedure applies when the P0_0 pin is used as DPO. If the P0_2 pin is used as DPO, specify the P0_2 pin settings in the same way.

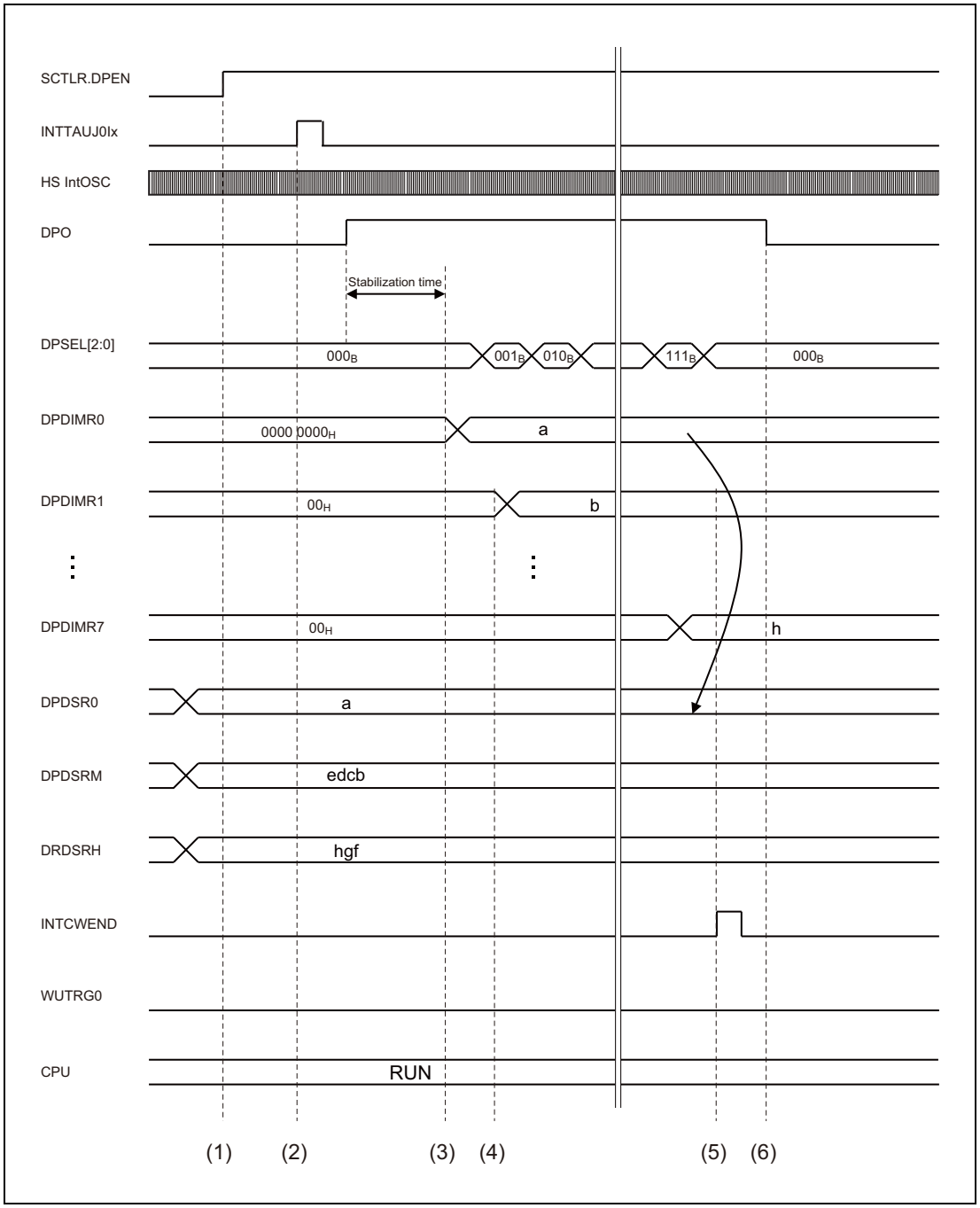


Figure 13.5 Operation of Digital Input Mode (8 Ports × 8) when the Input Value is not Changed (RUN Mode)

(1) Set the SCTLR.DPEN bit to 1 by software to enable the digital input mode of the LPS.

(2) When the INTTAUJ0Ix interrupt specified by the SCTLR.TJS bit is generated, the sequencer outputs the high level from the DPO pin and waits for the time specified by CNTVAL.CNT0n to secure the stabilization of the external digital signal source.

(3) After the completion of the signal source stabilization, the LPS stores the DPIN[7:0] input value to the DPDIMR0 register and increments the DPSEL[2:0] pins to switch the external multiplexer.

(4) After the switching of the DPSEL[2:0] pins, the LPS stores the values in the DPDIMRn registers in order from DPDIMR1 and continues to increment the DPSEL[2:0] pins.

(5) After the value is stored up to the DPDIMR7 register, the INTCWEND interrupt is generated and the value is compared with the expected value set in the DPDSR0, DPDSRM, and DPDSRH registers.

(6) When the value is not different from the expected value, the wake-up factor WUTRG0 is not generated. The LPS stops the DPO output and returns to the waiting state for the trigger.

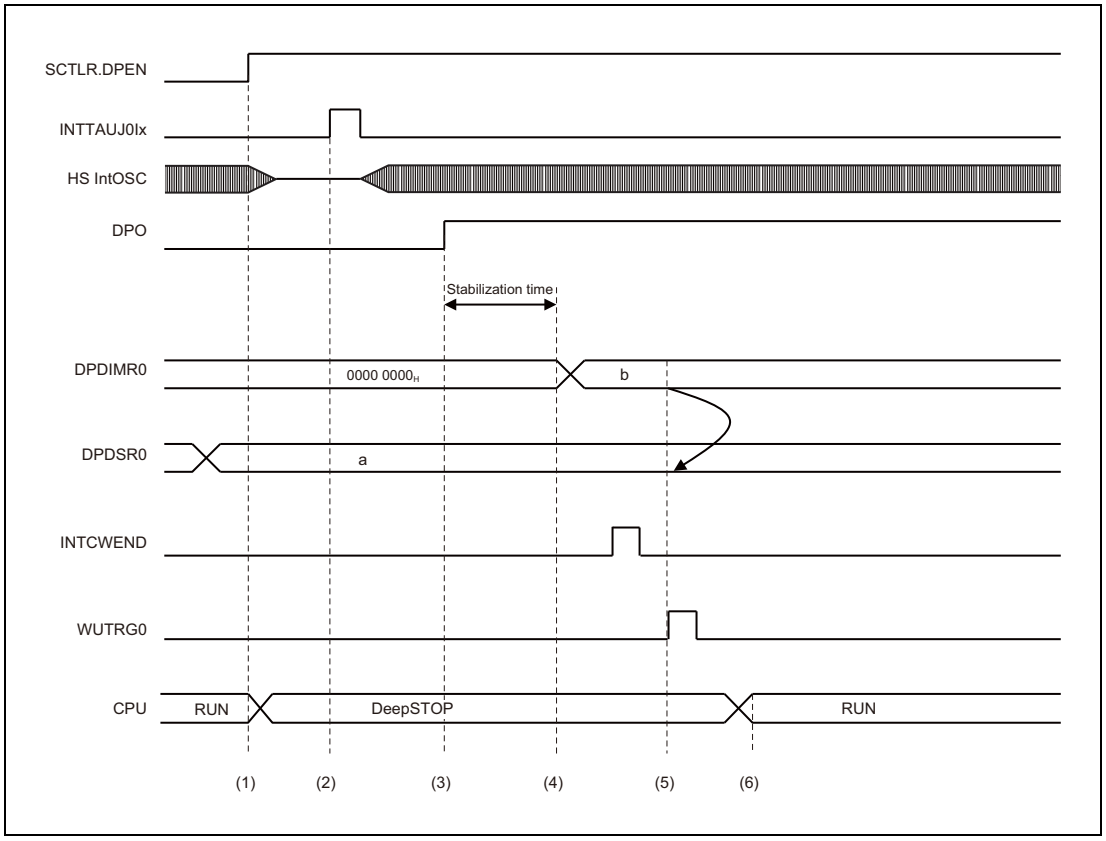


Figure 13.6 Operation of Digital Input Mode (24 Ports × 1) when the Input Value is Changed (DeepSTOP Mode)

(1) Set the STBC0PSC.STBC0DISTRG bit to 1 to transition to the DeepSTOP mode, while the SCTLR.DPEN bit is set to 1 by software to enable the digital input mode of the LPS.

(2) When the INTTAUJ0Ix interrupt specified by the SCTLR.TJS bit is generated, the LPS enables the HS IntOSC to start the oscillation.

(3) After the completion of the HS IntOSC stabilization time, the LPS outputs the high level from the DPO pin and waits for the time specified by CNTVAL.CNT0n to secure the stabilization of the external digital signal source.

(4) After the completion of the signal source stabilization, the LPS stores the DPIN[23:0] input value to the DPDIMR0 register and the INTCWEND interrupt is generated.

(5) The value stored in the DPDIMR0 register is compared with the expected value set in the DPDSR0 register. When the value is different from the expected value, the wake-up factor WUTRG0 is generated.

(6) The CPU returns to RUN mode at the generation of WUTRG0. The DPO pin is driven high until the EVFR.DINEVF bit is cleared to 0 by software.

13.5 Analog Input Mode

The analog input port ADCA0Im (m = 0 to 15) can be monitored.

TAUJ0 is used to set the timing to check the value input to the port.

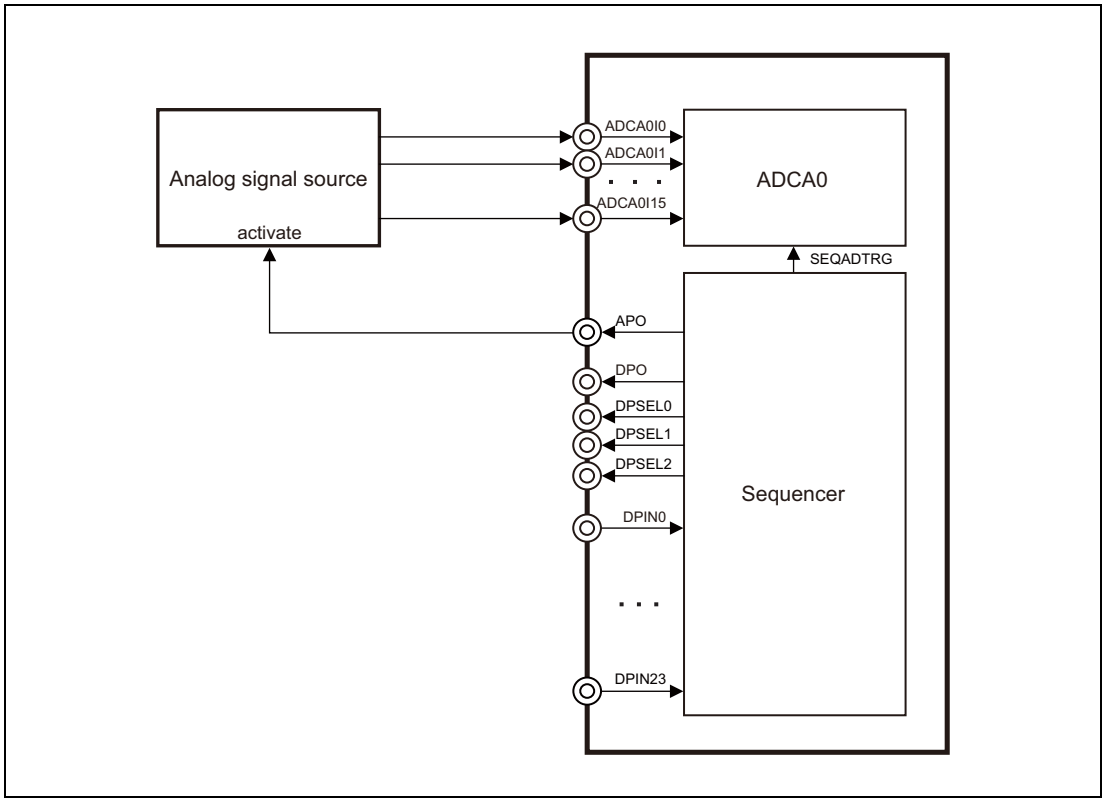


Figure 13.7 Analog Input Mode Connection Example

Preparation

- Set TJIS[1:0] bits in the SCTLR register to specify the TAUJ0 interrupt to be used as sequence start trigger.
- Set TAUJ0 to interval timer mode.
- Set the wait time of the analog signal source by using the upper 8 bits in the CNTVAL register.
- Set the ADCA0.

CAUTIONS

1. **When the LPS is in use, the A/D conversion completion interrupt (INT_SGx) should be output after the conversion of all channels of the LPS has been completed. The setting is as follows.**
   - **Set the ADIE bit in virtual channel register j (ADCA0VCRj) to 0 (a scan group x end interrupt (INT_SGx) is not generated when A/D conversion for virtual channel j ends in SGx.).**
   - **Set the ADIE bit in the scan group x control register (ADCA0SGCRx) to 1 (INT_SGx is output when the scan for SGx ends).**
2. **Over the period from the generation of the LPS sequence start trigger set by the SCLTR.TJIS[1:0] bits to the completion of A/D conversion for all channels of the LPS, only proceed with A/D conversion for PWM-Diag.**

3. **Over the period from the generation of the LPS sequence start trigger set by the SCLTR.TJIS[1:0] bits to the completion of A/D conversion for all channels of LPS, do not forcibly end A/D conversion by using the ADCA0ADHALTR.HALT bit.**
4. **When the LPS is in use, do not use the following modes.**
   - **Continuous scan mode (the setting ADCA0SGCRx.SCANMD = 1 is prohibited)**
   - **Multicycle scan mode with 2 or more cycles (the settings ADCA0SGMCYCRx.MCYC = 01_B and 11_B are prohibited)**
   - **Channel repeat mode with 2 or more cycles (the settings ADCA0SGCRx.SCT = 01_B and 10_B are prohibited)**

Start

- Start the TAUJ0.
- Set the SCTLR.ADEN bit to 1.

After the operation starts, ports are checked at the interval set in TAUJ0. The operation continues regardless of whether the mode is RUN mode or power save mode. If the HS IntOSC is stopped in stand-by mode, it can only resume operation while the sequencer is running.

To detect whether the analog input value differs from the expected value, use the A/D error interrupt request (INTADCA0ERR) of the A/D converter.

In addition, if the analog input value is different from the expected value, the wake-up factor WUTRG1 occurs.

For details on the A/D error interrupt request (INTADCA0ERR), see **Section 31.4.13, A/D Error Interrupt Request**[*1]. The following figures show an example of the operation in analog input mode.

Note 1. In **Section 31 A/D Converter (ADCA)**, the name of the A/D error interrupt request is described as “INT_ADE”.

Stop

To stop the LPS operation in Analog Input Mode (by changing the SCTLR.ADEN bit setting from 1 to 0), follow the procedure shown below. Note that the P0_1 pin is used as APO.

1. Set the port register to specify low level output on the pin (P0.P0_1 = 0).
2. Change the setting for the P0_1 pin from the alternative port mode to the port mode (PMC0.PMC0_1 = 0).
3. Set SCTLR.ADEN = 0.

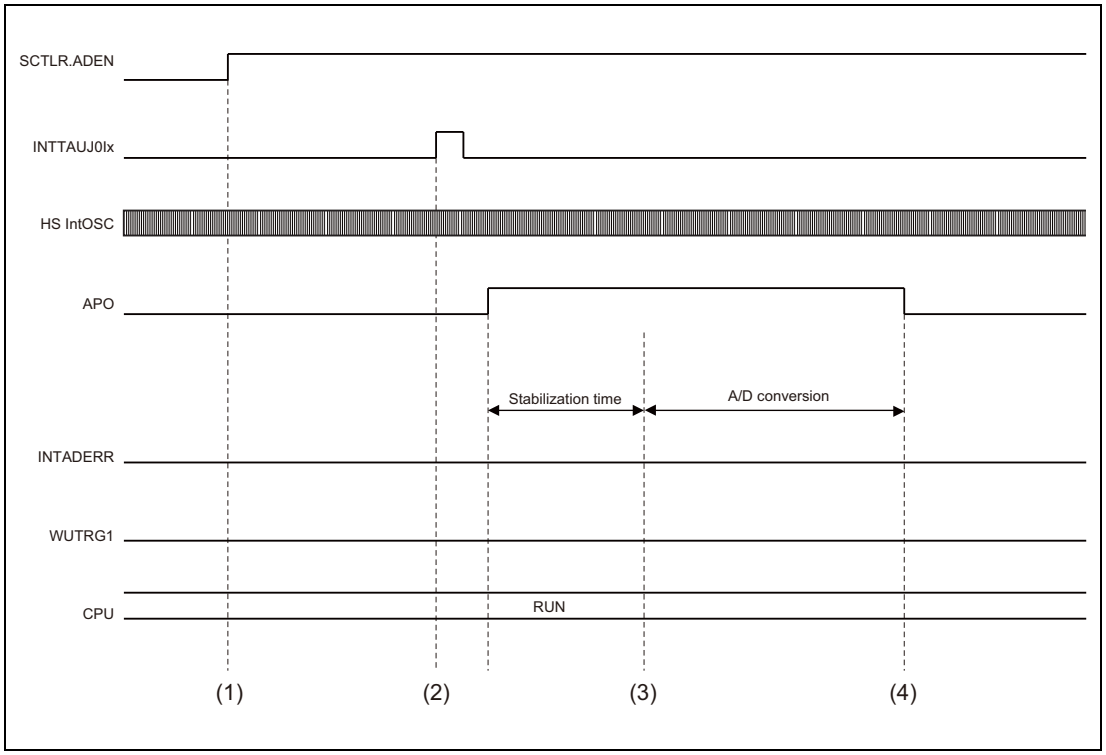


Figure 13.8 Operation of Analog Input Mode when the Conversion Result is within the Expected Range (RUN Mode)

(1) Set the conversion trigger, scan group, and expected range of the A/D converter by software. Then, set the SCTLR.ADEN bit to 1 to enable the analog input mode of the LPS.

(2) When the INTTAUJ0Ix interrupt specified by the SCTLR.TJS bit is generated, the LPS outputs the high level from the APO pin at the same time it enables the A/D converter, and waits for the time specified by CNTVAL.CNT1n to secure the stabilization of the external analog signal source. Set the stabilization time not less than 1 μs.

(3) After the completion of the signal source stabilization, the LPS triggers the start of conversion to the A/D converter and then the A/D conversion of ADCA0Im (m = 0 to 15), set in the A/D converter scan group, is started.

(4) When the INTADCA0ERR interrupt is not generated as a result of A/D conversion, the LPS halts the A/D converter and resets the APO pin.

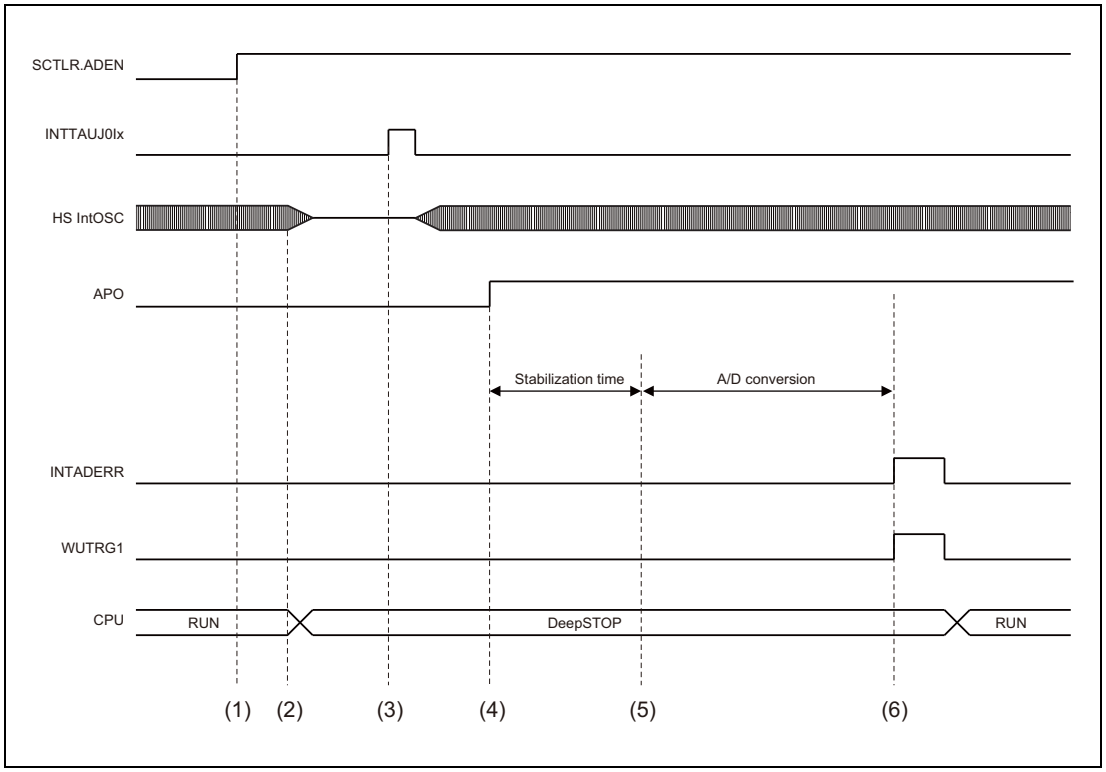


Figure 13.9 Operation of Analog Input Mode when the Conversion Result is not within the Expected Range (DeepSTOP Mode)

(1) Set the conversion trigger, scan group, and expected range of the A/D converter by software. Then, set the SCTLR.ADEN bit to 1 to enable the analog input mode of the LPS.

(2) Set the STBC0PSC.STBC0DISTRG bit to 1 by software to transition to the DeepSTOP mode.

(3) When the INTTAUJ0Ix interrupt specified by the SCTLR.TJS bit is generated, the LPS enables the HS IntOSC to start the oscillation.

(4) After the completion of the HS IntOSC stabilization, the LPS outputs the high level from the APO pin at the same time it enables the A/D converter, and waits for the time specified by CNTVAL.CNT1n to secure the stabilization of the external analog signal source.

(5) After the completion of the signal source stabilization, the LPS triggers the start of conversion to the A/D converter and then the A/D conversion of ADCA0Im (m = 0 to 15), set in the A/D converter scan group, is started.

(6) When the INTADCA0ERR interrupt is generated as a result of A/D conversion, the wake-up factor WUTRG1 is generated and the CPU returns to RUN mode. The APO pin is driven high until the upper limit/lower limit error flag of the A/D converter is cleared to 0 by software.

Section 14 External Memory Access Controller (MEMC)

This section contains a generic description of the external memory access controller (MEMC).

The first part in this section describes the features specific to RH850/F1M microcontrollers, including register base addresses and input/output signals. The ensuing sections describe the functions and registers of MEMC.

14.1 Features of MEMC

14.1.1 Products that Incorporate MEMC

MEMC is incorporated in the following products.

Table 14.1 Products that Incorporate MEMC

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	—	1	
Unit Name	—	MEMCn (n = 0)	

Table 14.2 Indices

Index	Description
n	Throughout this section, the individual MEMC units are identified by the index “n” (n = 0).
x	Throughout this section, chip select areas are identified by “x” (x = 0 to 3). For example, the external memory area of CSx is described as the CSx area.

Table 14.3 External Memory Access Functions

Control Signal Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Address output	—	20 bits	20 bits
Chip select output	—	4	4
Data bus	—	16 bits	16 bits

Note 1. The lower 16 bits of the address output are multiplexed with the data bus.

14.1.2 Register Base Address

The MEMC base address is shown in the following table.

The MEMC register addresses are expressed as offsets from the base address.

Table 14.4 Register Base Address

Base Address Name	Base Address
<MEMC0_base>	$1003\ 0000_H$

14.1.3 Clock Supply

The MEMC clock supply is shown in the following table.

Table 14.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
MEMC	MEMC0CLK	CPUCLK2*1

Note 1. This supply clock runs at half the frequency of the CPUCLK2 signal.

14.1.4 Reset Sources

The MEMC reset sources are shown below. MEMC is initialized by the following reset sources.

Table 14.6 Reset Sources

Unit Name	Reset Source
MEMC	All reset sources (ISORES)

14.1.5 External Input/Output Signals

External input/output signals of MEMC are listed below.

Table 14.7 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal Name
MEMC		
MEMC0A[19:16]	Address bus output signal	MEMC0A[19:16]
MEMC0AD[15:0]	Address/data bus I/O signal	MEMC0AD[15:0]
MEMC0ASTB	Address strobe output signal	$\overline{\text{MEMC0ASTB}}$
MEMC0BEN[1:0]	Byte enable output signal	$\overline{\text{MEMC0BEN[1:0]}}$
MEMC0CLK	Bus clock output signal	MEMC0CLK
MEMC0CS[3:0]	Chip select output signal	$\overline{\text{MEMC0CS[3:0]}}$
MEMC0RD	Read strobe output signal	$\overline{\text{MEMC0RD}}$
MEMC0WAIT	External wait request input signal	$\overline{\text{MEMC0WAIT}}$
MEMC0WR	Write strobe output signal	$\overline{\text{MEMC0WR}}$

Note 1. For the port to be used as MEMC, set the output driver strength to high driver strength (PDSCn_m=1).

14.2 Overview

The external memory access controller provides four chip select areas, and wait time is selectable in each chip select area.

The bus clock runs at half the frequency of the MEMC supply clock.

14.2.1 Functional Overview

The main features of the external memory access controller:

- Multiplexed bus mode
- Four chip select areas and 16-bit bus width
- The data endian format can be selected for each chip select area individually.
- Various wait functions can be set individually for each chip select area.
- External wait on SRAM access cycles
- External wait error detection

14.2.1.1 Multiplexed Bus

This is an operation mode that connects address output and data input/output to external memory using the same signal line, making it possible to reduce the number of pins required for external memory connection.

14.2.1.2 Chip Select Output Function

The external bus area of the memory space is divided into four chip select areas, and a chip select signal can be output for each chip select area. The allocation of these chip select areas is fixed by the system and cannot be changed through programming.

In addition, the bus width is 16-bit fixed.

14.2.1.3 Data Endian Setting Function

The data endian (little endian/big endian) can be specified for each chip select area.

14.2.1.4 Programmable Wait Setting Functions

This microcontroller has the following wait functions, which can be set for each chip select area.

- Programmable data wait
- Data hold wait
- Data setup wait
- Address setup wait
- Address hold wait
- Idle cycle

14.2.1.5 External Wait Function

Waits of any width can be inserted externally from the $\overline{\text{MEMC0WAIT}}$ pin. The $\overline{\text{MEMC0WAIT}}$ pin is sampled just before the data output cycle, and the data latch timing can be delayed by any amount.

14.2.1.6 External Wait Error Detection Function

This microcontroller has a function to forcibly cancel a wait state if an external wait is continuously input for 128 clock cycles, to prevent the system hanging up if an external wait is continuously input due to a defect of the MEMC0WAIT pin.

14.3 Registers

14.3.1 List of Registers

The MEMC registers are listed in the following table.

For details on <MEMCn_base>, see **Section 14.1.2, Register Base Address**.

Table 14.8 List of Registers

Module Name	Register Name	Symbol	Address
MEMC	Data endian configuration register	DEC	<MEMCn_base> + 02_H
	Data wait configuration register	DWC	<MEMCn_base> + 08_H
	Data hold wait configuration register	DHC	<MEMCn_base> + $0C_H$
	Data setup wait configuration register	DSC	<MEMCn_base> + $0E_H$
	Address wait configuration register	AWC	<MEMCn_base> + 10_H
	Idle cycle configuration register	ICC	<MEMCn_base> + 14_H
	External wait error configuration register	EWC	<MEMCn_base> + $1A_H$

14.3.2 DEC — Data Endian Configuration Register

The DEC register is used to set the endianness of the external bus.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + 02_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	DE3	—	DE2	—	DE1	—	DE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W

Table 14.9 DEC Register Contents

Bit Position	Bit Name	Function
15 to 7, 5, 3, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6, 4, 2, 0	DEx	Data Endian Setting These bits set the endian of each chip select area. 0: Little endian 1: Big endian

The relationship between each chip select area and the control bit is shown below.

Table 14.10 Relationship between DEx Bit and Chip Select Area

Chip Select Area	DEx Bit
CS3 area	DE3
CS2 area	DE2
CS1 area	DE1
CS0 area	DE0

14.3.3 DWC — Data Wait Configuration Register

The DWC register is used to set the number of data wait states for the external bus.

Access: These registers can be read or written in 16-bit units.

Address: <MEMCn_base> + 08_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DW33	DW32	DW31	DW30	DW23	DW22	DW21	DW20	DW13	DW12	DW11	DW10	DW03	DW02	DW01	DW00
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.11 DWC Register Contents

Bit Position	Bit Name	Function
15 to 0	DWx3, DWx2, DWx1, DWx0	Data Wait Setting These bits set the number of data wait states for each chip select area.

DWx3	DWx2	DWx1	DWx0	Number of Data Wait States
0	0	0	0	setting prohibited
0	0	0	1	1 clock cycle
0	0	1	0	2 clock cycles
0	0	1	1	3 clock cycles
0	1	0	0	4 clock cycles
0	1	0	1	5 clock cycles
0	1	1	0	6 clock cycles
0	1	1	1	7 clock cycles
1	0	0	0	8 clock cycles
1	0	0	1	9 clock cycles
1	0	1	0	10 clock cycles
1	0	1	1	11 clock cycles
1	1	0	0	12 clock cycles
1	1	0	1	13 clock cycles
1	1	1	0	14 clock cycles
1	1	1	1	15 clock cycles

The relationship between each chip select area and the control bit is shown below.

Table 14.12 Relationship between DWx3 to DWx0 Bits and Chip Select Area

Chip Select Area	DWx3 to DWx0 Bits
CS3 area	DW33, DW32, DW31, DW30
CS2 area	DW23, DW22, DW21, DW20
CS1 area	DW13, DW12, DW11, DW10
CS0 area	DW03, DW02, DW01, DW00

14.3.4 DHC — Data Hold Wait Configuration Register

The DHC register is used to set the number of data hold waits for the external bus.

By setting this register, a data hold wait of "DHC register setting + one cycle" is inserted in a write cycle.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + $0C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DH31	DH30	DH21	DH20	DH11	DH10	DH01	DH00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.13 DHC Register Contents

Bit Position	Bit Name	Function
15 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	DHx1, DHx0	Data Hold Wait Setting These bits set the number of data wait states for each chip select area.

DHx1	DHx0	Number of Data Hold Wait
0	0	No data hold wait
0	1	1 clock cycle
1	0	2 clock cycles
1	1	3 clock cycles

The relationship between each chip select area and the control bit is shown below.

Table 14.14 Relationship between DHx1, DHx0 Bits and Chip Select Area

Chip Select Area	DHx1, DHx0 Bits
CS3 area	DH31, DH30
CS2 area	DH21, DH20
CS1 area	DH11, DH10
CS0 area	DH01, DH00

14.3.5 DSC — Data Setup Wait Configuration Register

The DSC register is used to set the number of data setup wait states for the external bus.

This wait is inserted in a write cycle.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + $0E_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DS31	DS30	DS21	DS20	DS11	DS10	DS01	DS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.15 DSC Register Contents

Bit Position	Bit Name	Function
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	DSx1, DSx0	Data Setup Wait Setting These bits set the number of data setup wait states for each chip select area.

DSx1	DSx0	Number of Data Setup Wait States
0	0	No data setup wait
0	1	1 clock
1	0	2 clocks
1	1	3 clocks

The relationship between each chip select area and the control bit is show below.

Table 14.16 Relationship between DSx1 and DSx0 Bits and Chip Select Area

Chip Select Area	DSx1, DSx0 Bits
CS3 area	DS31, DS30
CS2 area	DS21, DS20
CS1 area	DS11, DS10
CS0 area	DS01, DS00

14.3.6 AWC — Address Wait Configuration Register

The AWC register is used to set the address wait period of the external bus for each chip select area.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + 10_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AHW31	AHW30	ASW31	ASW30	AHW21	AHW20	ASW21	ASW20	AHW11	AHW10	ASW11	ASW10	AHW01	AHW00	ASW01	ASW00
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.17 AWC Register Contents

Bit Position	Bit Name	Function
15, 14, 11, 10, 7, 6, 3, 2	AHWx1, AHWx0	Address Hold Wait Setting These bits set the number of address hold wait states for each chip select area.
13, 12, 9, 8, 5, 4, 1, 0	ASWx1, ASWx0	Address Setup Wait Setting These bits set the number of address setup wait states for each chip select area.

AHWx1	AHWx0	Number of Address Hold Wait
0	0	No address hold wait
0	1	1 clock
1	0	2 clocks
1	1	3 clocks

ASWx1	ASWx0	Number of Address Setup Wait
0	0	No address setup wait
0	1	1 clock
1	0	2 clocks
1	1	3 clocks

The relationship between each chip select area and the control bit is shown below.

Table 14.18 Relationship between ASWx1/ASWx0/AHWx1/AHWx0 Bits, and Chip Select Area

Chip Select Area	ASWx1, ASWx0 and AHWx1, AHWx0 Bits
CS3 area	ASW31, ASW30, AHW31, AHW30
CS2 area	ASW21, ASW20, AHW21, AHW20
CS1 area	ASW11, ASW10, AHW11, AHW10
CS0 area	ASW01, ASW00, AHW01, AHW00

14.3.7 ICC — Idle Cycle Configuration Register

The ICC register is used to set the number of idle cycles of the external bus. The number of idle cycles can be set for each chip select area.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + 14_H

Value after reset: 3333_H

CAUTION

The Idle cycles set by the ICC register is invalid during bus sizing cycle. However, idle cycles are valid only during read cycle. The number of idle cycles to be inserted depends on the ICC register setting. When “no idle cycle” is set, 1 clock cycle is inserted.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WIC31	WIC30	RIC31	RIC30	WIC21	WIC20	RIC21	RIC20	WIC11	WIC10	RIC11	RIC10	WIC01	WIC00	RIC01	RIC00
Value after reset	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.19 ICC Register Contents

Bit Position	Bit Name	Function
15, 14, 11, 10, 7, 6, 3, 2	WICx1, WICx0	Idle Cycle Setting Bits after Write Cycle These bits set the number of idle cycles for each chip select area after a write cycle.
13, 12, 9, 8, 5, 4, 1, 0	RICx1, RICx0	Idle Cycle Setting Bits after Read Cycle These bits set the number of idle cycles for each chip select area after a read cycle.

WICx1	WICx0	Number of Idle Cycles
0	0	No idle cycle
0	1	1 clock
1	0	2 clocks
1	1	3 clocks

RICx1	RICx0	Number of Idle Cycles
0	0	No idle cycle
0	1	1 clock
1	0	2 clocks
1	1	3 clocks

The relationship between each chip select area and control bits is shown below.

Table 14.20 Relationship between RICx1/RICx0/WICx1/WICx0 Bits and Chip Select Area

Chip Select Area	RICx1, RICx0, WICx1, WICx0 Bits
CS3 area	RIC31, RIC30, WIC31, WIC30
CS2 area	RIC21, RIC20, WIC21, WIC20
CS1 area	RIC11, RIC10, WIC11, WIC10
CS0 area	RIC01, RIC00, WIC01, WIC00

14.3.8 EWC — External Wait Error Configuration Register

The EWC register is used to enable and disable the external wait error function.

Access: This register can be read or written in 16-bit units.

Address: <MEMCn_base> + $1A_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	EW3	—	EW2	—	EW1	—	EW0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W

Table 14.21 EWC Register Contents

Bit Position	Bit Name	Function
15 to 7, 5, 3, 1	Reserved	When read, the value after reset is read. When writing, write the value after reset.
6, 4, 2, 0	EWx	External wait error setting These bits are used to enable and disable the external wait error in each chip select area. 0: External wait error disabled 1: External wait error enabled When this function is enabled, a wait state is forcibly released and SYSERR exception is occurred if an external wait request is detected continuously for 128 clock cycles.

The relationship between each chip select area and control bits is shown below.

Table 14.22 Relationship between EWx Bits and Chip Select Area

Chip Select Area	EWx Bits
CS3 area	EW3
CS2 area	EW2
CS1 area	EW1
CS0 area	EW0

14.4 Functions

14.4.1 Bus Control Functions

14.4.1.1 Chip Select Output Function

The connected external memory area is divided into and managed in 4 chip select areas, as shown in **Figure 14.1, External Memory Map**.

When a bus cycle is generated for the external bus, this microcontroller makes the $\overline{\text{MEMC0CS[3:0]}}$ output pin corresponding to the access target address active (low level), along with outputting the access target address from the MEMC0A[19:16] and MEMC0AD[15:0].

The various settings for the external bus, such as the number of wait/idle states, can all be made for each chip select area.

By using these functions, different types of memory can be connected to each chip select area.

The allocation of the chip select areas is fixed by the system and cannot be changed through programming.

The memory map is shown below.

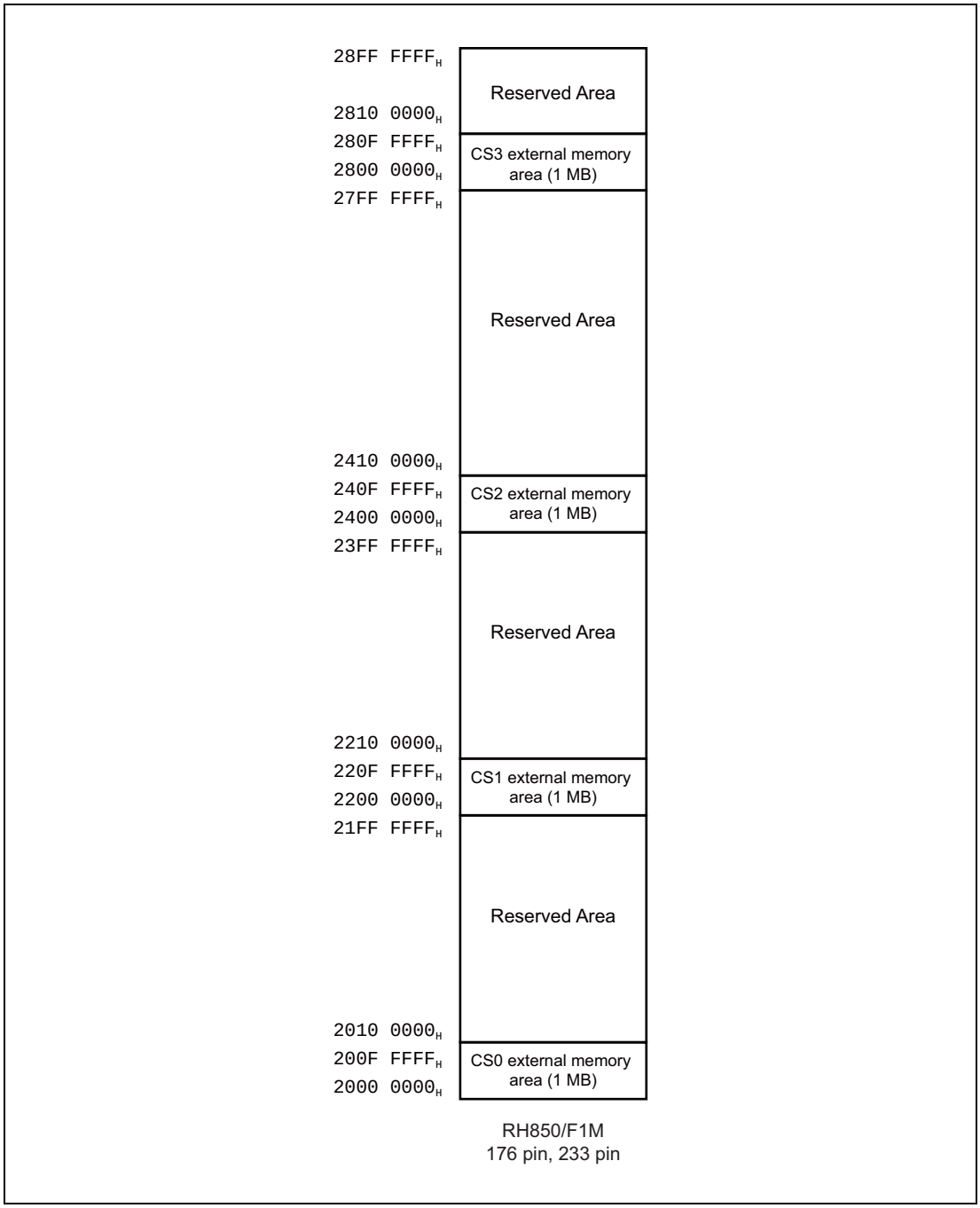
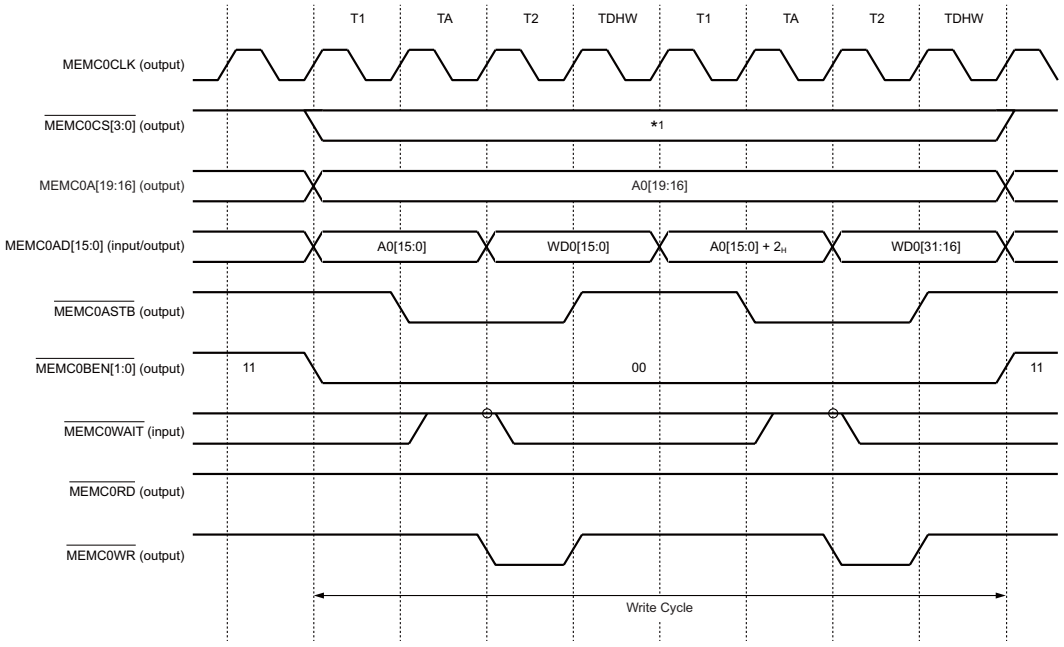


Figure 14.1 External Memory Map

14.4.1.2 Bus Sizing Function

Access requests from the CPU (or DMA) are executed after being divided in accordance with the bit width of the external bus of the access destination.



Note 1. Indicates the chip select area to be accessed.

Note: T_1: Address setup period
T_A: Address strobe period
T_2: Read (write) strobe period
T_{DHW}: Data hold wait period

Figure 14.2 Bus Cycles when Making a 32-Bit Write Access to the External Bus

14.4.1.3 Data Endian Setting Function

Either little endian or big endian can be selected as the data endian of the external bus interface. This setting can be made for each chip select area with the DEC register.

CAUTIONS

1. **In this microcontroller, instruction fetch operation with big endian is not supported. Assembler and debugger were made for little endian.**
2. **Misaligned access in big endian format is not supported.**

14.4.2 Wait Functions

Wait functions are listed below.

Table 14.23 Wait Functions

Wait Function	Data Wait		Data Hold Wait	Data Setup Wait	Address Wait	Idle State
	Programmable	External Pin				
Read access	√	√	—	—	√	√
Write access	√	√	√	√	√	√
Setting register	DWC	—	DHC	DSC	AWC	ICC
Max. number of waits	15	—	3	3	3	3

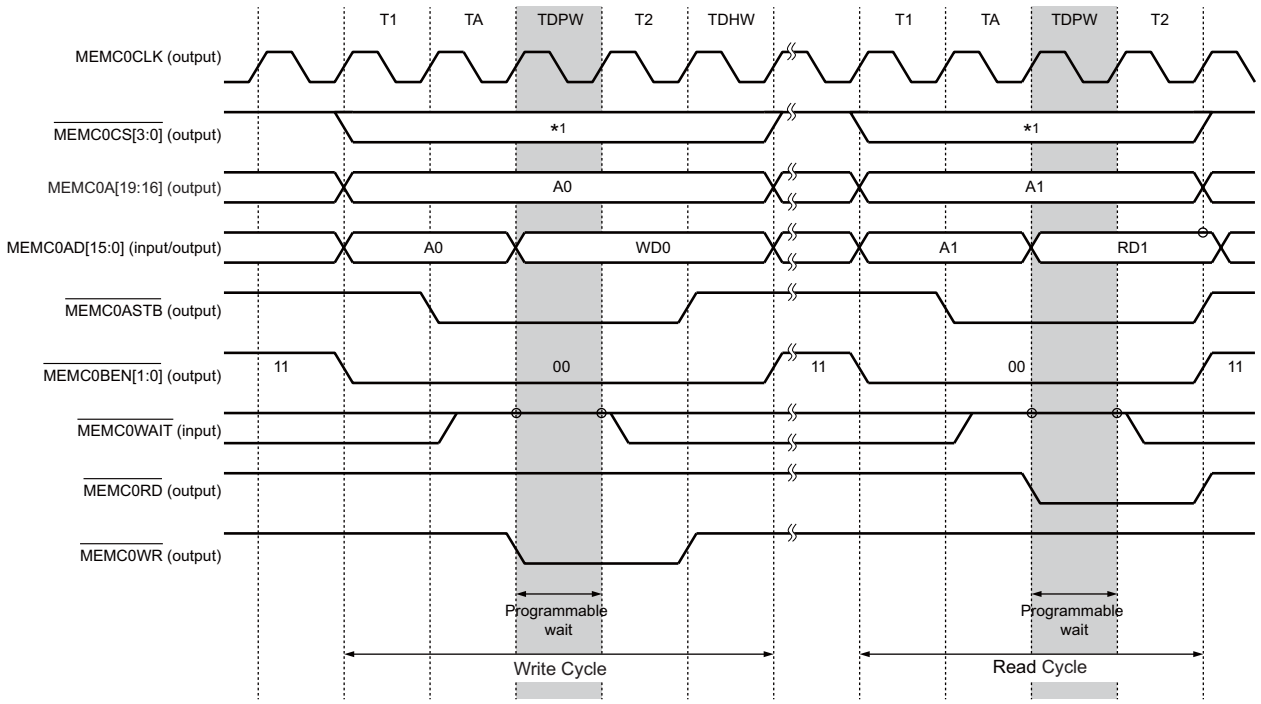
14.4.2.1 Programmable Data Wait Function

This wait function is used to delay the data latch timing by extending the read strobe and write strobe periods.

This function is enabled during data transfer.

Up to 15 cycles can be inserted.

Data waits can be set for each chip select area using the DWC register.



Note 1. Indicates the chip select area to be accessed.

Note: T_1: Address setup period
T_A: Address strobe period
T_2: Read (write) strobe period
T_{DPW}: Programmable data wait period
T_{DHW}: Data hold wait period

Figure 14.3 Bus Cycles when Inserting Programmable Data Waits

14.4.2.2 External Wait Function

Data waits of any length can be inserted from the $\overline{\text{MEMC0WAIT}}$ pin.

The $\overline{\text{MEMC0WAIT}}$ pin input level is sampled immediately after completion of the T_A cycle and the T_{DPW}, T_{DEW} cycles.

Data wait cycles obtained by ORing the programmable data wait set by data wait control register (DWC register) and the external wait set by the $\overline{\text{MEMC0WAIT}}$ pin input, are inserted.

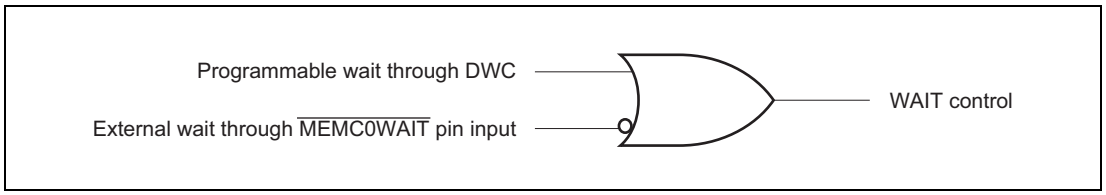
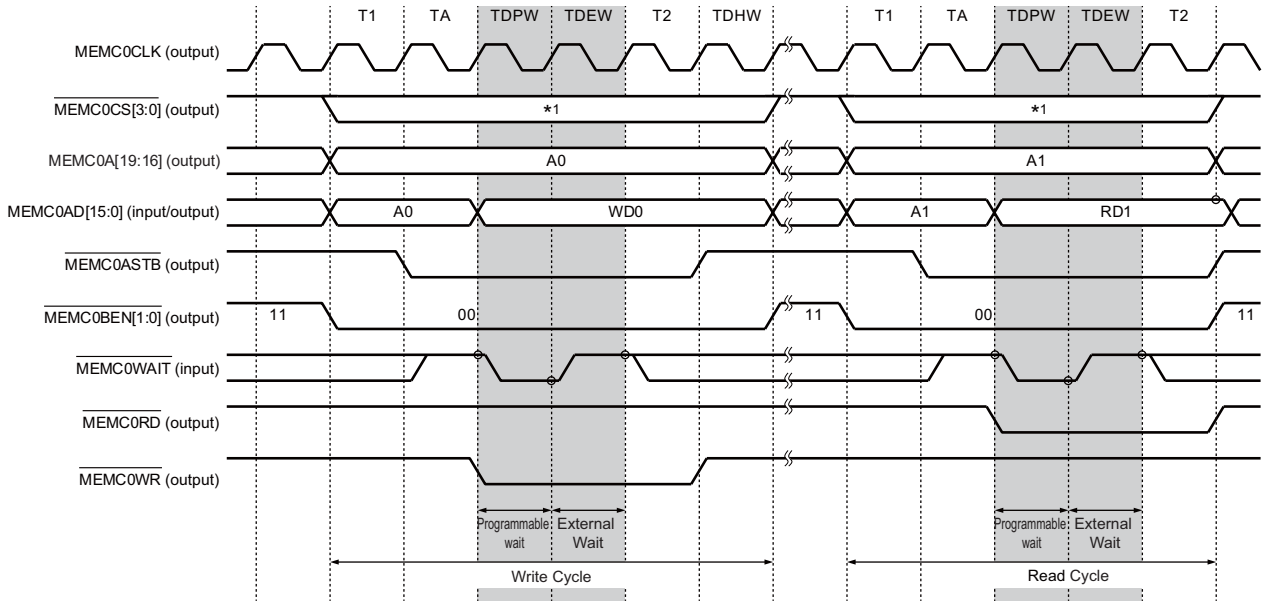


Figure 14.4 Internal Data Wait Generator



Note 1. Indicates the chip select area to be accessed.

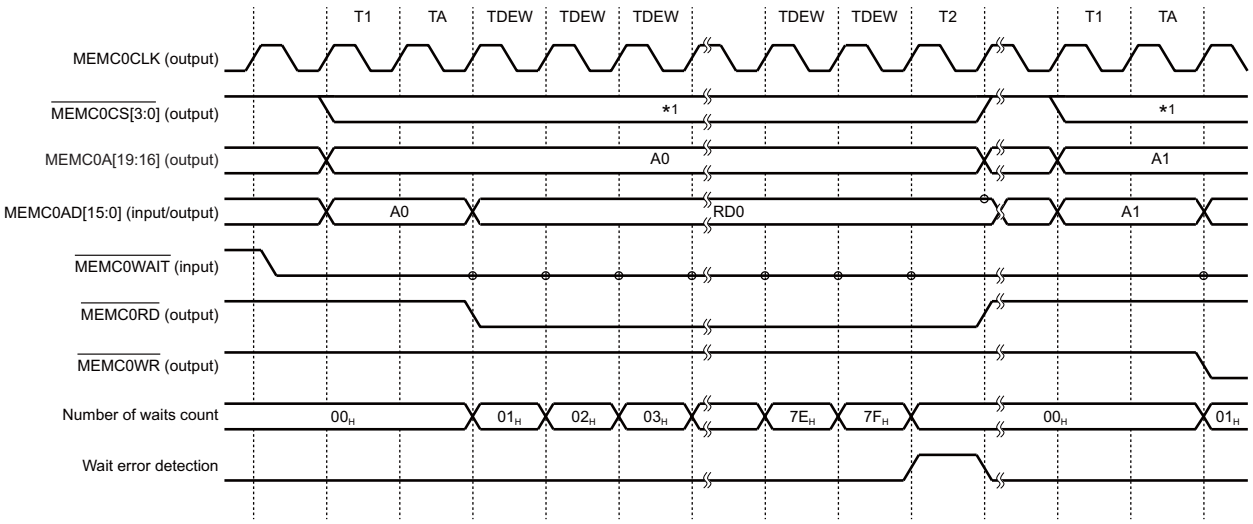
Note: T_1: Address setup period
T_A: Address strobe period
T_2: Read (write) strobe period
T_{DPW}: Programmable data wait period
T_{DEW}: External data wait period
T_{DHW}: Data hold wait period

Figure 14.5 Bus Cycles when Inserting Programmable Data Waits and External Waits (while DWC = 1)

14.4.2.3 External Wait Error Detection Function

By setting the EWx bit in the external wait error configuration register (EWC register) to 1, a wait state is forcibly canceled and SYSERR exception is generated if an external wait request is input continuously for 128 clock cycles.

This function prevents the system hanging up due to an unexpected failure of the $\overline{\text{MEMC0WAIT}}$ input pin.



Note 1. Indicates the chip select area to be accessed.

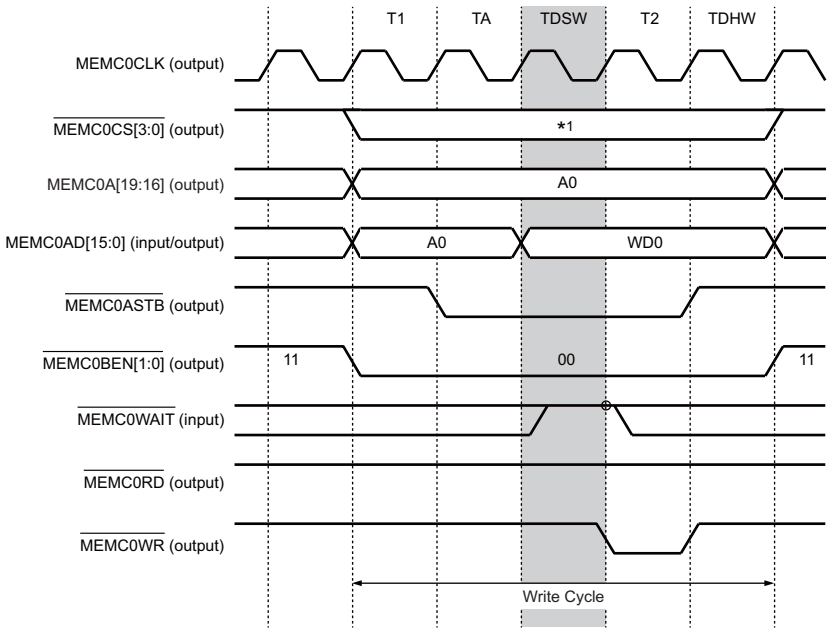
Note: T_1: Address setup period
T_A: Address strobe period
T_{DEW}: External data wait period
T_2: Read (write) strobe period

Figure 14.6 Operation Timing During External Wait Error Detection

14.4.2.4 Data Setup Wait Function

This function inserts waits before a transfer state to secure setup time for data write strobe, and is only valid during a write cycle.

Waits of up to 3 cycles per chip select area can be inserted by setting the DSC register.



Note 1. Indicates the chip select area to be accessed.

Note: T_1: Address setup period
T_A: Address strobe period
T_{DSW}: Data setup wait period
T_2: Read (write) strobe period
T_{DHW}: Data hold wait period

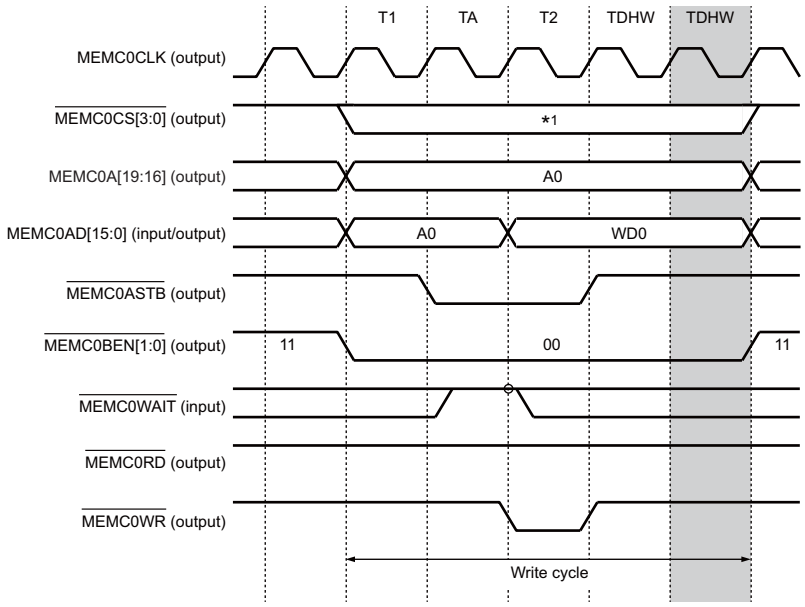
Figure 14.7 Bus Cycles when Inserting Data Setup Waits

14.4.2.5 Data Hold Wait Function

This function inserts a wait in the state following the rising edge of the write strobe signal in order to secure the hold time for the data write strobe.

This microcontroller always inserts 1 data hold wait state upon occurrence of a write cycle. This data hold wait can be extended by up to 3 cycles by setting the DHC register, allowing insertion of 4 cycles.

The number of cycles to extend the data hold wait can be set for each chip select area with the DHC register.



Note 1. Indicates the chip select area to be accessed.

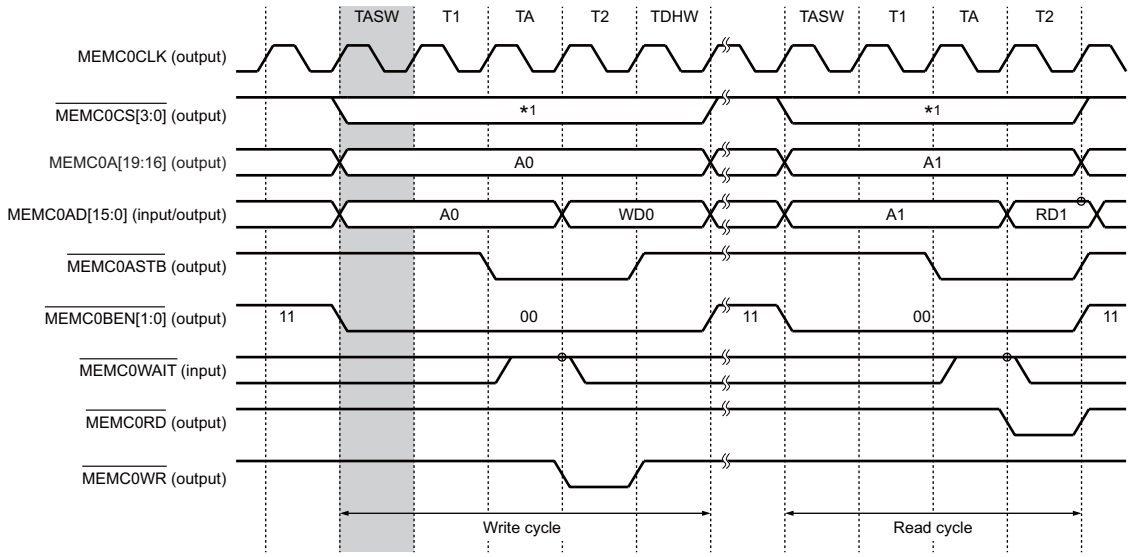
Note: T_1: Address setup period
T_A: Address strobe period
T_2: Read (write) strobe period
T_{DHW}: Data hold wait period

Figure 14.8 Bus Cycles when Inserting Data Hold Waits

14.4.2.6 Address Setup Wait Function

The address setup wait function inserts a wait before the address transfer state in order to secure the setup time for the address strobe.

Up to 3 wait cycles can be inserted for each chip select area by setting the AWC register.



Note 1. Indicates the chip select area to be accessed.

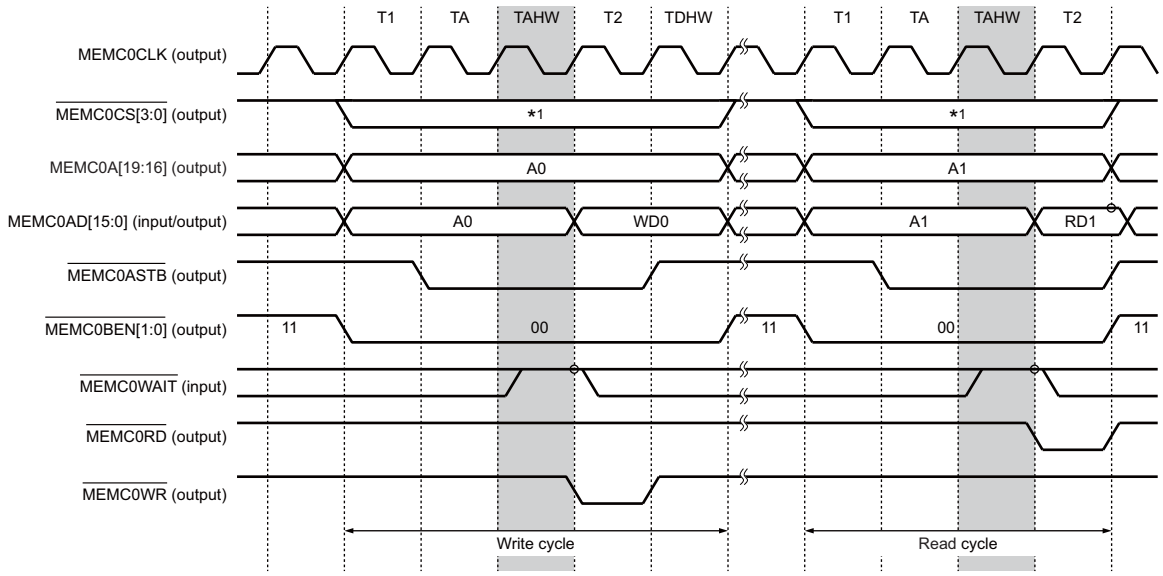
Note: T_1: Address setup period
T_{ASW}: Address setup wait period
T_A: Address strobe period
T_2: Read (write) strobe period
T_{DHW}: Data hold wait period

Figure 14.9 Bus Cycles when Inserting Address Setup Waits

14.4.2.7 Address Hold Wait Function

The address hold wait function inserts waits after the address transfer state to secure hold time for address strobe.

Up to 3 wait cycles can be inserted for each chip select area by setting the AWC register.



Note 1. Indicates the chip select area to be accessed.

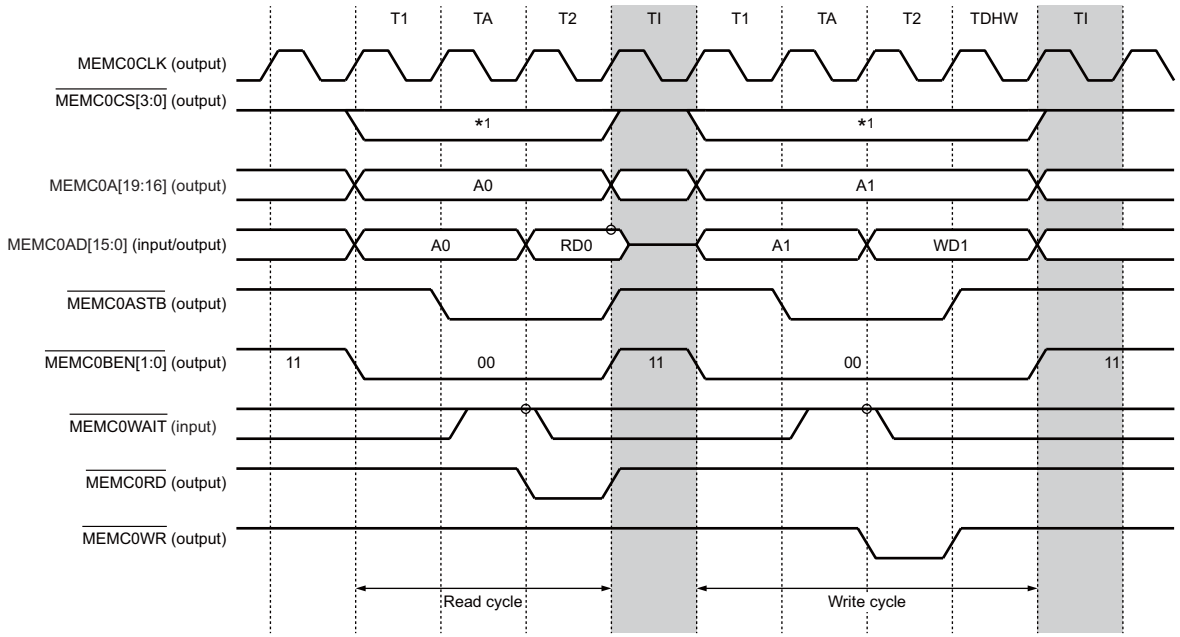
Note: T_1: Address setup period
T_A: Address strobe period
T_{AHW}: Address hold wait period
T_2: Read (write) strobe period
T_{DHW}: Data hold wait period

Figure 14.10 Bus Cycles when Inserting Address Hold Waits

14.4.2.8 Idle Insertion Function

This function inserts an idle state after the last state of each cycle in order to prevent bus conflicts between cycles.

This function can be set independently after a read cycle/write cycle for each chip select area by setting the ICC register.



Note 1. Indicates the chip select area to be accessed.

Note: T_1: Address setup period
T_A: Address strobe period
T_2: Read (write) strobe period
T_I: Idle state period
T_{DHW}: Data hold wait period

Figure 14.11 Bus Cycles when Inserting Idle States

14.4.3 Memory Connection Example

14.4.3.1 SRAM Connection Example

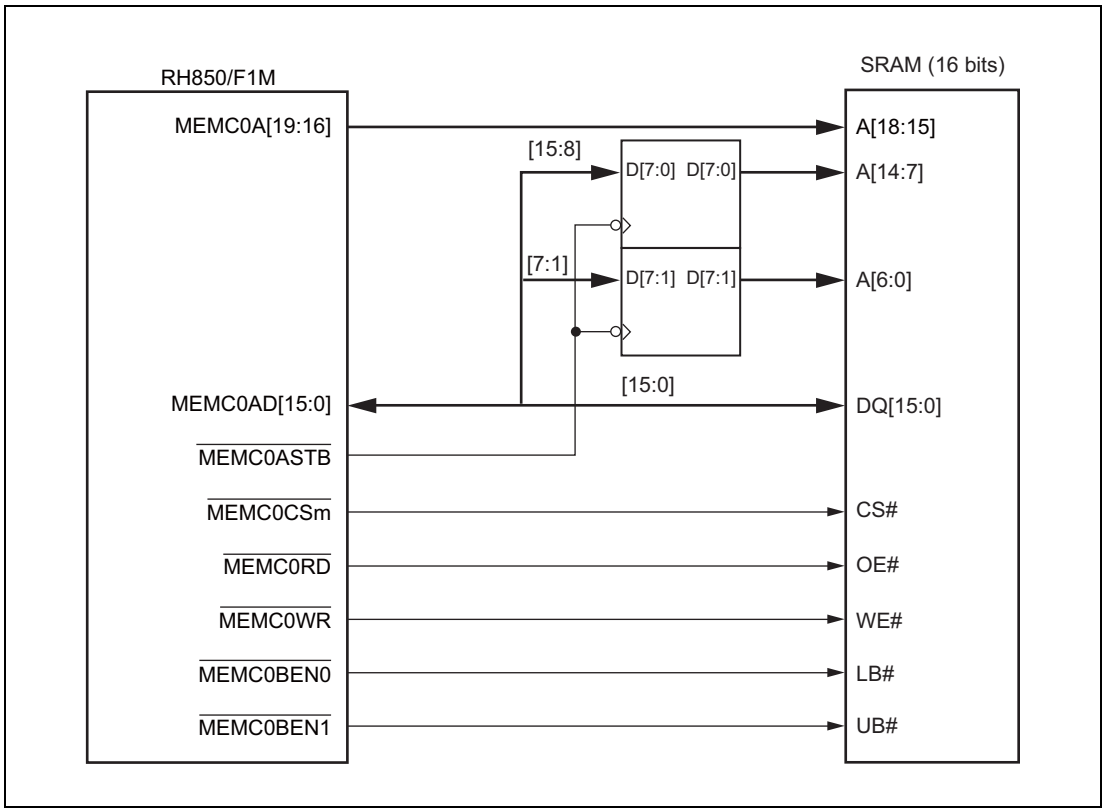


Figure 14.12 SRAM (16 Bits) Connection Example

14.4.4 Data Access Flow

The data transfer flow to the external memory differs according to the data width, specified endian, external bus width, and start address.

For misaligned access, the CPU performs the division and coupling of data. Whether the access is misaligned or not depends on the data width and start address.

Misaligned

As the accesses listed below are misaligned, the CPU divides the cycles.

- When half-word (16-bit) data is read from/written to an odd address
- When word (32-bit) data is read from/written to an address that is not a multiple of 4

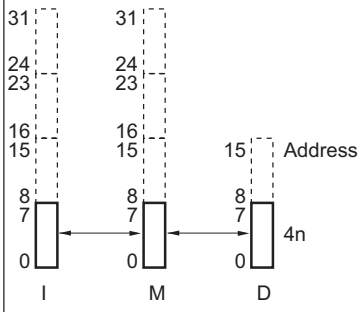
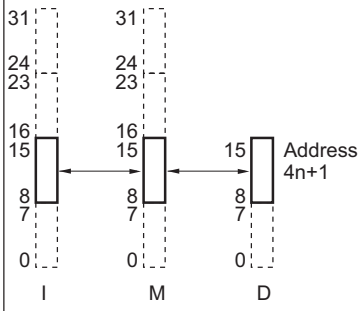
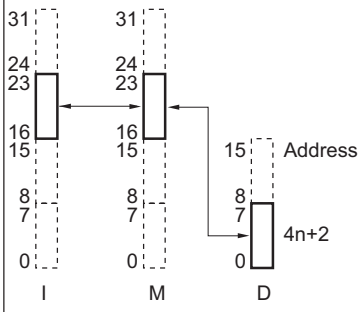
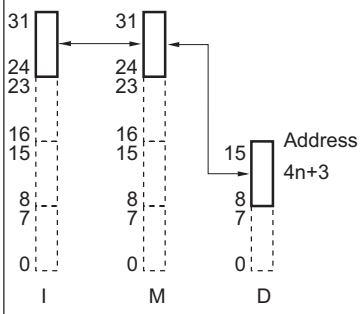
Table 14.24 Misaligned Access Conditions

Access Conditions		Cycles Divided by the CPU		
Data Width	**Address**	**1st**	**2nd**	**3rd**
16 bits	2n + 1	8-bit access to 2n + 1	8-bit access to 2n + 2	—
32 bits	4n + 1	8-bit access to 4n + 1	16-bit access to 4n + 2	8-bit access to 4n + 4
32 bits	4n + 2	16-bit access to 4n + 2	16-bit access to 4n + 4	—
32 bits	4n + 3	8-bit access to 4n + 3	16-bit access to 4n + 4	8-bit access to 4n + 6

The data flows for each condition are shown on the following pages.

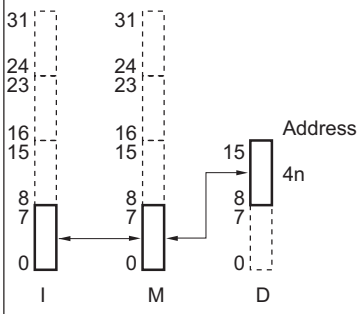
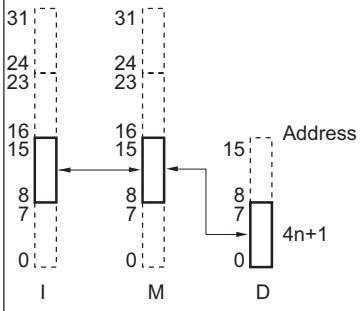
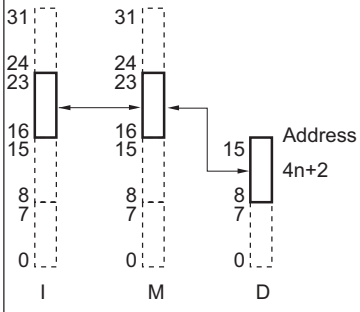
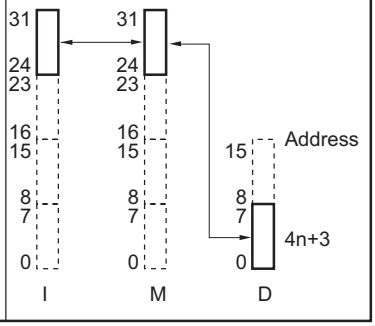
14.4.4.1 Data Flow for Byte Access (for Reading and Writing)

Table 14.25 Data Flow for Byte Access (Little Endian)

Address to be Accessed	Data Transfer Flow
4n	
4n+1	
4n+2	
4n+3	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

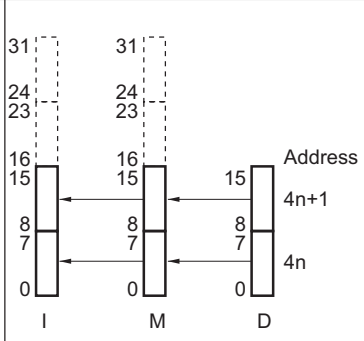
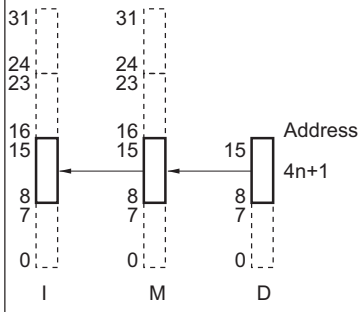
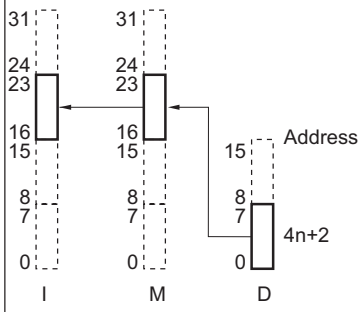
Table 14.26 Data Flow for Byte Access (Big Endian)

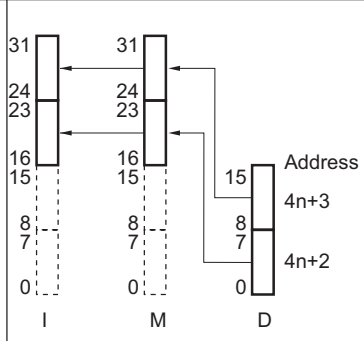
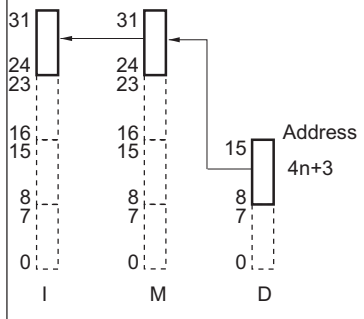
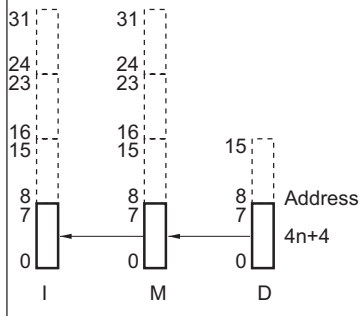
Address to be Accessed	Data Transfer Flow
4n	
4n+1	
4n+2	
4n+3	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

14.4.4.2 Data Flow for Half-Word Read Access

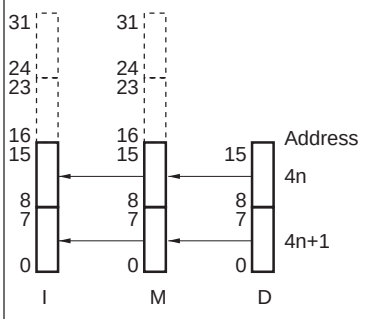
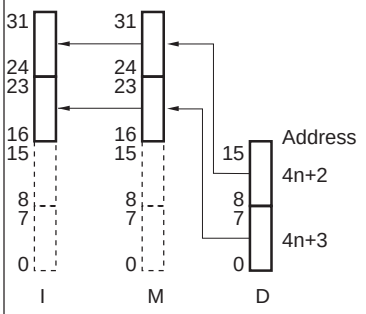
Table 14.27 Data Flow for Half-Word Read Access (Little Endian)

Address to be Accessed		Data Transfer Flow
4n	1st	
4n+1	1st	
	2nd	

Address to be Accessed		Data Transfer Flow
4n+2	1st	
4n+3	1st	
	2nd	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 14.28 Data Flow for Half-Word Read Access (Big Endian)

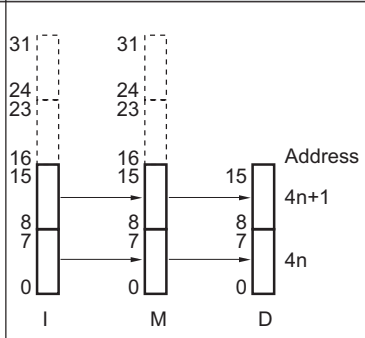
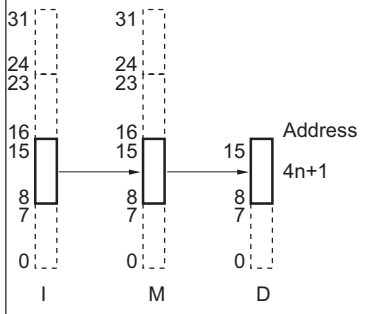
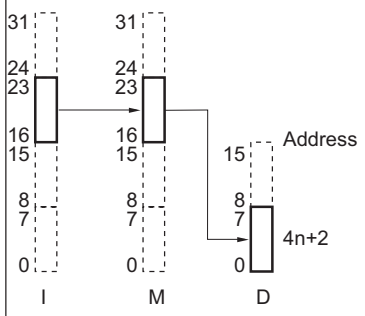
Address to be Accessed	Data Transfer Flow
4n	
4n+2	

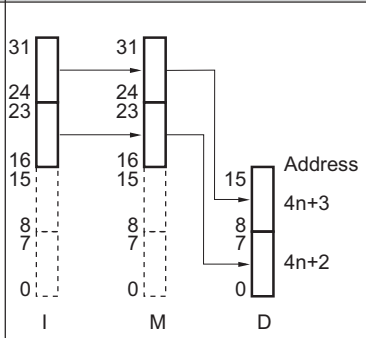
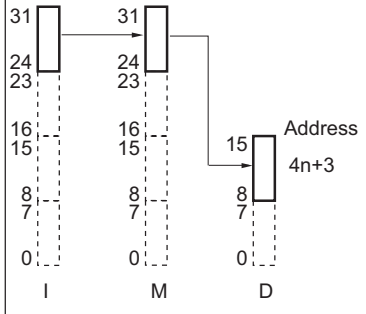
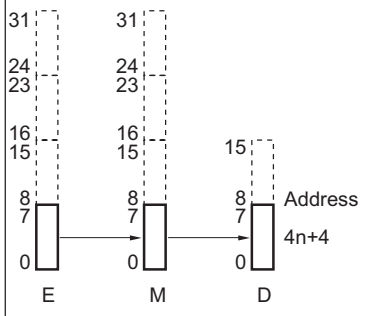
Note 1. I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing an address starting with 4n + 1 or 4n + 3 is prohibited.

14.4.4.3 Data Flow for Half-Word Write Access

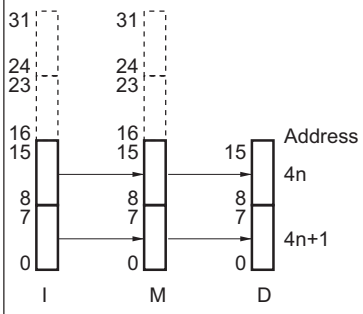
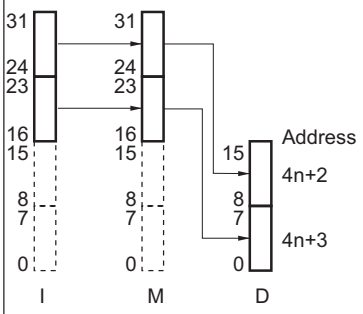
Table 14.29 Data Flow for Half-Word Write Access (Little Endian)

Address to be Accessed		Data Transfer Flow
4n	1st	
4n+1	1st	
	2nd	

Address to be Accessed		Data Transfer Flow
4n+2	1st	
4n+3	1st	
	2nd	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 14.30 Data Flow for Half-Word Write Access (Big Endian)

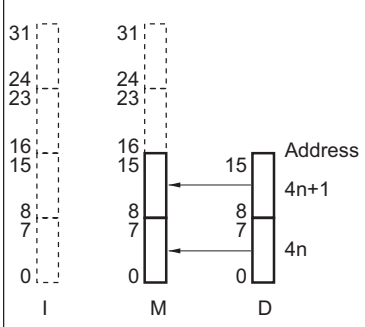
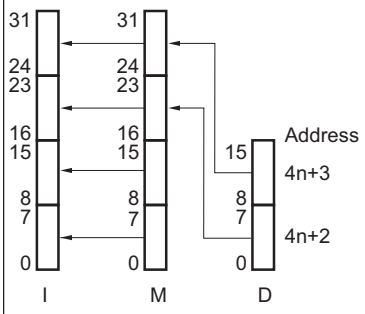
Address to be Accessed	Data Transfer Flow
4n	
4n+2	

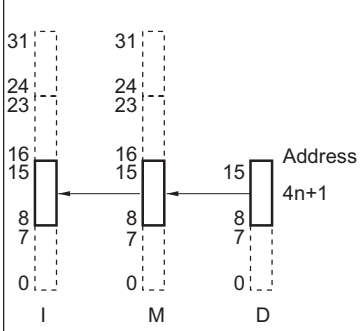
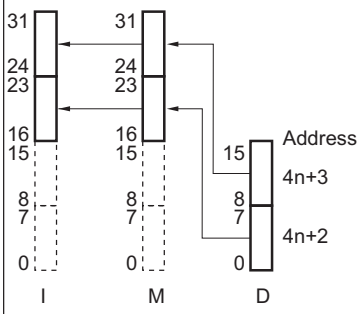
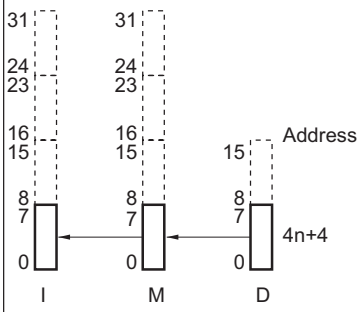
Note 1. I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing an address starting with 4n + 1 or 4n + 3 is prohibited.

14.4.4.4 Data Flow for Word Read Access

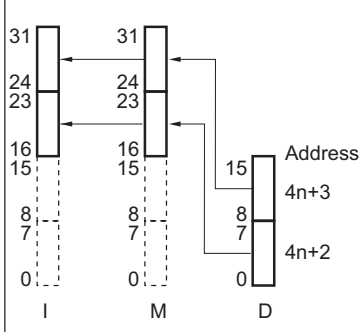
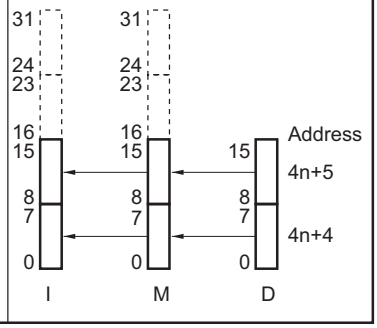
Table 14.31 Data Flow for Word Read Access (Little Endian) (1/2)

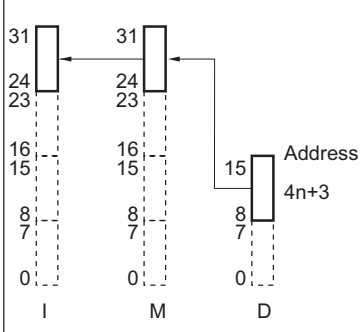
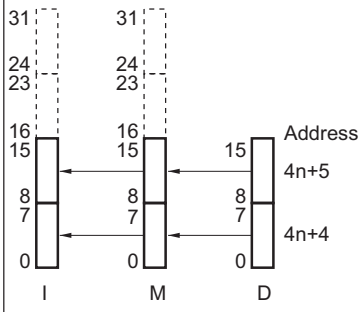
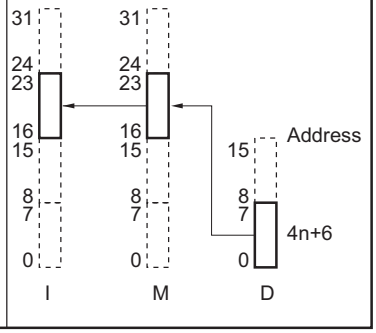
Address to be Accessed		Data Transfer Flow
4n	1st	
	2nd	

Address to be Accessed		Data Transfer Flow
4n+1	1st	
	2nd	
	3rd	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

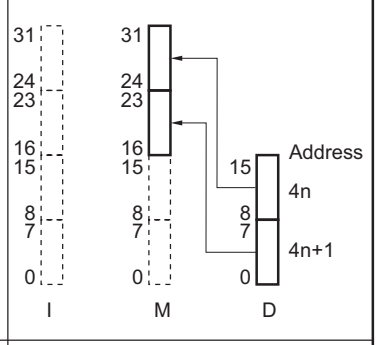
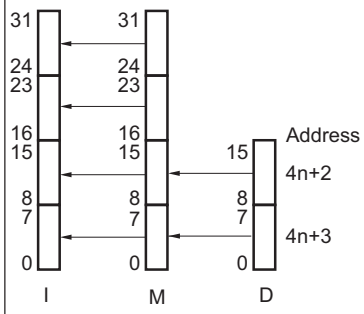
Table 14.31 Data Flow for Word Read Access (Little Endian) (2/2)

Address to be Accessed		Data Transfer Flow
4n+2	1st	
	2nd	

Address to be Accessed		Data Transfer Flow
4n+3	1st	
	2nd	
	3rd	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 14.32 Data Flow for Word Read Access (Big Endian)

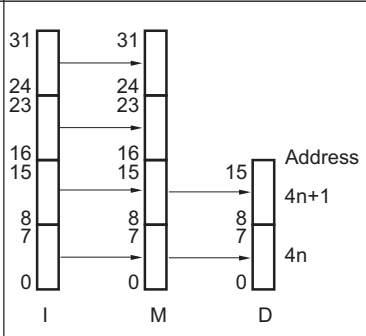
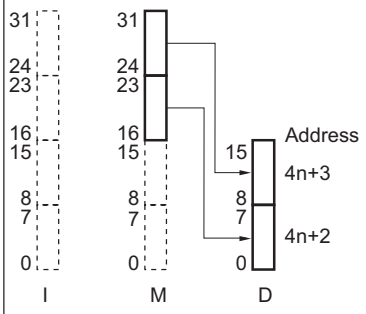
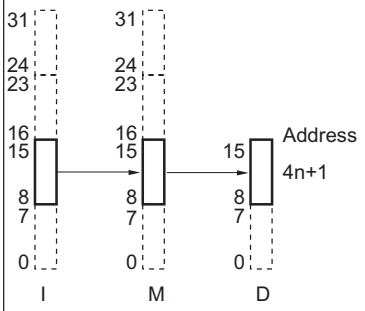
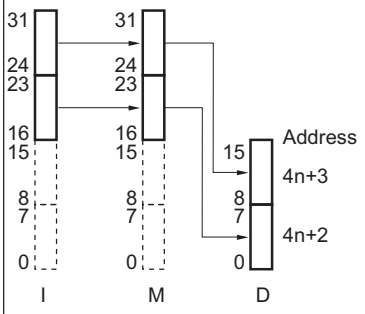
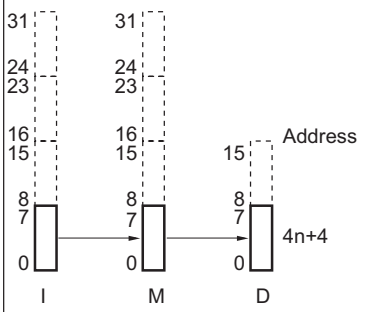
Address to be Accessed		Data Transfer Flow
4n	1st	
	2nd	

Note 1. I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing an address starting with 4n + 1, 4n + 2, or 4n + 3 is prohibited.

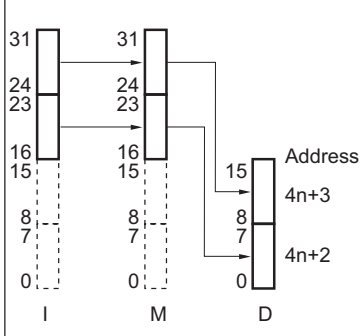
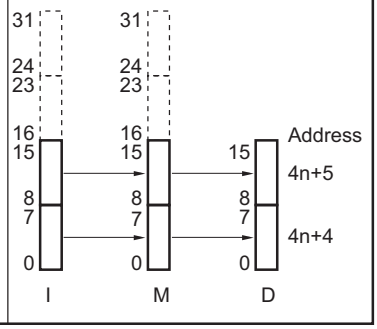
14.4.4.5 Data Flow for Word Write Access

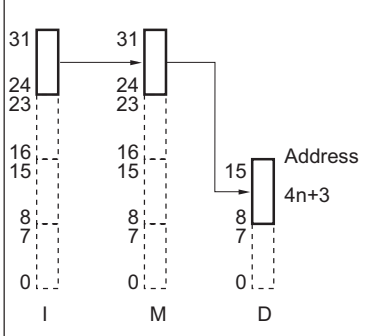
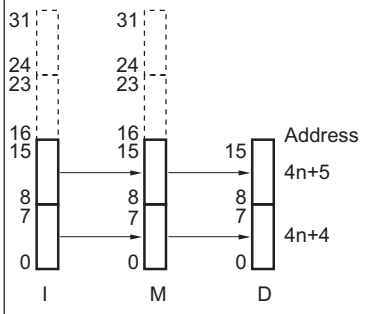
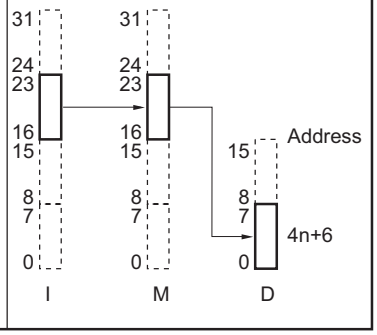
Table 14.33 Data Flow for Word Write Access (Little Endian) (1/2)

Address to be Accessed		Data Transfer Flow
4n	1st	
	2nd	
4n+1	1st	
	2nd	
	3rd	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

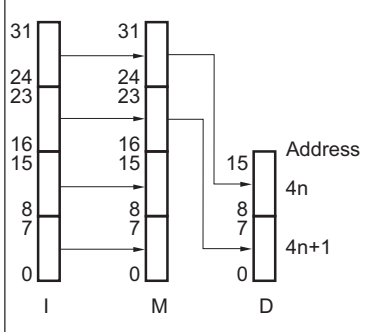
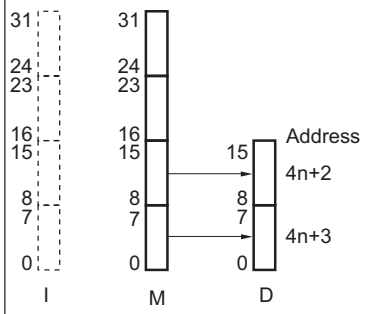
Table 14.33 Data Flow for Word Write Access (Little Endian) (2/2)

Address to be Accessed		Data Transfer Flow
4n+2	1st	
	2nd	

Address to be Accessed		Data Transfer Flow
4n+3	1st	
	2nd	
	3rd	

Note: I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 14.34 Data Flow for Word Write Access (Big Endian)

Address to be Accessed		Data Transfer Flow
4n	1st	
	2nd	

Note 1. I: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing an address starting with 4n + 1, 4n + 2, or 4n + 3 is prohibited.

14.5 Notes on Use of MEMC

Stop MEMC before transitioning to standby mode.

Section 15 Clocked Serial Interface G (CSIG)

This section contains a generic description of the Clocked Serial Interface G (CSIG).

The first part in this section describes the features specific to RH850/F1H, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the CSIG.

15.1 Features of RH850/F1M CSIG

15.1.1 Number of Units

This microcontroller has the following number of CSIG units.

Each CSIG unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 15.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	2		
Name	CSIGn (n = 0, 1)		

Table 15.2 Index

Index	Description
n	Throughout this section, the individual CSIG units are identified by the index “n” (n = 0, 1): for example, CSIGnCTL0 is the CSIGn control register 0.

15.1.2 Register Base Address

CSIG base addresses are listed in the following table.

CSIG register addresses are given as offsets from the base addresses.

Table 15.3 Register Base Addresses

Base Address Name	Base Address
<CSIG0_base>	FFD8 8000_H
<CSIG1_base>	FFD8 $A000_H$

15.1.3 Clock Supply

The CSIG clock supply is shown in the following table.

Table 15.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
CSIGn	PCLK	CKSCLK_ICSI
	Register access	CKSCLK_ICSI

15.1.4 Interrupt Request

CSIG interrupt requests are listed in the following table.

Table 15.5 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
CSIG0			
INTCSIGTIC	Communication status interrupt	27,118	8
INTCSIGTIR	Receive status interrupt	28,119	9
INTCSIGTIRE	Communication error interrupt	57	—
CSIG1			
INTCSIGTIC	Communication status interrupt	223	109
INTCSIGTIR	Receive status interrupt	224	110
INTCSIGTIRE	Communication error interrupt	225	—

15.1.5 Reset Sources

CSIG reset sources are listed in the following table. CSIG is initialized by these reset sources.

Table 15.6 Reset Sources

Unit Name	Reset Source
CSIGn	All reset sources (ISORES)

15.1.6 External Input/Output Signals

External input/output signals of CSIG are listed below.

Table 15.7 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal Name
CSIG0		
CSIGTSCK	Serial clock signal	CSIG0SC*1
CSIGTSI	Serial data input signal	CSIG0SI
CSIGTSO	Serial data output signal	CSIG0SO*1
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG0SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG0RYI
CSIGTRYO	Ready / busy output signal	CSIG0RYO
CSIG1		
CSIGTSCK	Serial clock signal	CSIG1SC*1
CSIGTSI	Serial data input signal	CSIG1SI
CSIGTSO	Serial data output signal	CSIG1SO*1
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG1SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG1RYI
CSIGTRYO	Ready / busy output signal	CSIG1RYO

Note 1. For the port pins that are used as CSIGnSO and CSIGnSC, set the output driving ability to high (PDSCn_m = 1).

15.1.7 Data Consistency Check

The port and the alternative function for data consistency check of CSIGnSO (CSIGTSO) output are shown in the following table. See **Section 15.5.10, Error Detection** for details on data consistency checking.

Table 15.8 Port Pins for Data Consistency Checking

Unit Signal Name	Port Pin Name	Alternative Function
CSIG0		
CSIGTSO	P0_13	ALT_OUT4
	P10_6	ALT_OUT2
CSIG1		
CSIGTSO	P11_9	ALT_OUT1

15.2 Overview

15.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Master mode or slave mode is selectable.
- Slave select input signal ($\overline{\text{CSIGTSSI}}$) is available.
- Built-in baud rate generator
- Transfer clock frequency is adjustable in master mode, whereas it is determined by the input clock in slave mode.
- Maximum transfer clock frequency:
  - In master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)
- Clock phases and data phases are selectable.
- Data transfer with MSB first or LSB first is selectable.
- Transfer data length is selectable from 7 to 16 bits in 1-bit units
- Incorporates an EDL (extended data length) function for transferring more than 16 bits of data
- Three selectable transfer modes:
  - transmit-only mode
  - receive-only mode
  - transmit/receive mode
- Built-in handshake function
- Built-in error detection (data consistency check, parity, overrun)
- Three different interrupt request signals (INTCSIGTIC, INTCSIGTIR, INTCSIGTIRE)
- Built-in LBM (Loop Back Mode) function for self-test

15.2.2 Functional Overview Description

The CSIG uses three signals for communication:

- Transmission clock CSIGTSCK (output in master mode, input in slave mode)
- Serial data output signal CSIGTSO
- Serial data input signal CSIGTSI

The CSIGnCTL2 register is used to select whether the CSIG should be operated in master mode or slave mode.

Additional signals can be used for external control and monitoring:

- CSIGTSSI: Slave select input signal
- CSIGTRYO: Ready/busy output signal (handshake signal)
- CSIGTRYI: Ready/busy input signal (handshake signal)

Data transmission is bit-wise and serial and synchronous to the transmission clock.

The following table shows the most important registers for setting up CSIG.

Table 15.9 Main Registers of CSIG

Register	Function
CSIGnCTL0	Provides and stops operating clock and enables/disables data transmission and data reception.
CSIGnCTL1	Controls options such as interrupt timing, extended data length, data consistency check, loop-back mode, handshake, etc.
CSIGnCTL2	Selects master or slave mode, and the transfer clock frequency of the built-in baud rate generator (BRG) in master mode.
CSIGnCFG0	Configures the communication protocol.

15.2.3 Block Diagram

The following block diagram shows the main components of the CSIG.

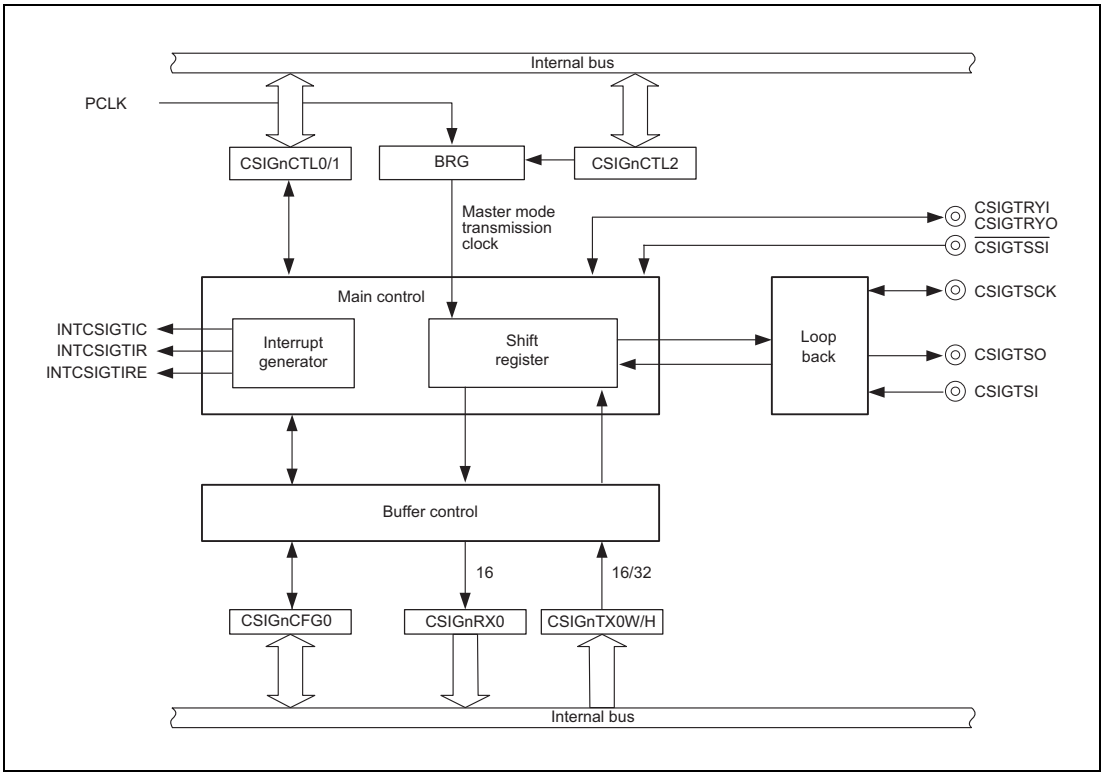


Figure 15.1 CSIG Block Diagram

In master mode, the transmission clock CSIGTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is supplied by an external source.

15.3 Registers

15.3.1 List of Registers

CSIG registers are listed in the following table.

For details on <CSIGn_base>, see **Section 15.1.2, Register Base Address**.

Table 15.10 List of Registers

Module Name	Register Name	Symbol	Address
CSIGn	CSIGn control register 0	CSIGnCTL0	<CSIGn_base> + 0000_H
CSIGn	CSIGn control register 1	CSIGnCTL1	<CSIGn_base> + 0010_H
CSIGn	CSIGn control register 2	CSIGnCTL2	<CSIGn_base> + 0014_H
CSIGn	CSIGn status register 0	CSIGnSTR0	<CSIGn_base> + 0004_H
CSIGn	CSIGn status clear register 0	CSIGnSTCR0	<CSIGn_base> + 0008_H
CSIGn	CSIGn Receive-only mode control register 0	CSIGnBCTL0	<CSIGn_base> + 1000_H
CSIGn	CSIGn configuration register 0	CSIGnCFG0	<CSIGn_base> + 1010_H
CSIGn	CSIGn transmission register 0 for word access	CSIGnTX0W	<CSIGn_base> + 1004_H
CSIGn	CSIGn transmission register 0 for half word access	CSIGnTX0H	<CSIGn_base> + 1008_H
CSIGn	CSIGn reception register 0	CSIGnRX0	<CSIGn_base> + $100C_H$
CSIGn	CSIGn emulation register	CSIGnEMU	<CSIGn_base> + 0018_H

15.3.2 CSIGnCTL0 — CSIGn Control Register 0

This register controls the operation clock, and enables or disables transmission/reception.

Access: This register can be read or written in 1-bit and 8-bit units.

Address: <CSIGn_base> + 0000_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CSIGnPWR	CSIGnTXE	CSIGnRXE	—	—	—	—	CSIGnMBS
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R/W

Table 15.11 CSIGnCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIGnPWR	Controls operation clock. 0: Stops operation clock. 1: Supplies operation clock. Clearing CSIGnPWR to 0 resets the internal circuits, stops operation, and sets the CSIG to standby state. Clock supply to internal circuits stops. If CSIGnPWR is cleared during communication, ongoing communication is aborted. In this case, communication setting must be started from the beginning.
6	CSIGnTXE	Enables/disables transmission. 0: Transmission disabled 1: Transmission enabled
5	CSIGnRXE	Enables/disables reception. 0: Reception disabled 1: Reception enabled
4 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CSIGnMBS	This bit must be set to 1 (the value after reset is 0).

CAUTION

When configuring this register, see **Table 15.24, List of Cautions when Configuring the Registers**.

15.3.3 CSIGnCTL1 — CSIGn Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It also enables/disables extended data length control, data consistency check, loop-back mode, handshake function, and slave select function.

Access: This register can be read or written in 32-bit units.

Address: <CSIGn_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CSIGn CKR	CSIGn SLIT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CSIGn EDLE	—	CSIGn DCS	—	CSIGn LBM	CSIGn SIT	CSIGn HSE	CSIGn SSE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 15.12 CSIGnCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	CSIGnCKR	CSIGTSCK clock inversion function 0: Default level of CSIGTSCK is high. 1: Default level of CSIGTSCK is low. The CSIGnCKR bit is used in combination with the CSIGnCFG0.CSIGnDAP bit. For details, see **Section 15.3.8, CSIGnCFG0 — CSIGn Configuration Register 0**.
16	CSIGnSLIT	Selects the timing of interrupt INTCSIGTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: Interrupt generation when CSIGnTX0W/H is empty and available for storing the next data. For details, see **15.4.2, INTCSIGTIC (Communication Status Interrupt)**
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	CSIGnEDLE	Enables/disables extended data length (EDL) mode. 0: Extended data length mode disabled 1: Extended data length mode enabled For details, see **15.5.5.2, Data Length Selection with Extended Data Length**.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CSIGnDCS	Enables/disables data consistency check. 0: Data consistency check disabled 1: Data consistency check enabled For details, see **15.5.10.1, Data Consistency Check**.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	CSIGnLBM	Controls loop-back mode (LBM). 0: Loop-back mode deactivated 1: Loop-back mode activated Loop-back mode can be set only in master mode. Set this bit to 0 in slave mode. For details, see **Section 15.5.9, Loop-Back Mode**.

Table 15.12 CSIGnCTL1 Register Contents (2/2)

Bit Position	Bit Name	Function
2	CSIGnSIT	Selects interrupt delay mode. 0: No delay 1: Half clock delay for all interrupts This bit is only valid in master mode. In slave mode, no delay is generated. For details, see **Section 15.4.1, Interrupt Delay**.
1	CSIGnHSE	Enables/disables handshake function. 0: Handshake function disabled 1: Handshake function enabled For details, see **Section 15.5.8, Handshake Function**.
0	CSIGnSSE	Enables/disables slave select function. 0: Input signal $\overline{\text{CSIGTSSI}}$ disabled. 1: Input signal $\overline{\text{CSIGTSSI}}$ enabled. If the slave select function is not used, this bit must be set to 0 (see also **Section 15.5.2, Master/Slave Connections**).

Details about CSIGnCTL1.CSIGnSSE:

Table 15.13 Operation of the Slave Select Function during Reception

CSIGnCTL0. CSIGnRXE	CSIGnCTL1. CSIGnSSE	$\overline{\text{CSIGTSSI}}$	Receive Operation
0	—	—	Reception disabled
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

Table 15.14 Operation of the Slave Select Function during Transmission

CSIGnCTL0. CSIGnTXE	CSIGnCTL1. CSIGnSSE	$\overline{\text{CSIGTSSI}}$	Transmit Operation
0	—	—	Transmission disabled
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

CAUTION

When setting this register, see **Table 15.24, List of Cautions when Configuring the Registers**.

15.3.4 CSIGnCTL2 — CSIGn Control Register 2

This register selects the communication clock.

Access: This register can be read or written in 16-bit units.

Address: <CSIGn_base> + 0014_H

Value after reset: $E000_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnPRS[2:0]			—	CSIGnBRS[11:0]											
Value after reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.15 CSIGnCTL2 Register Contents

Bit Position	Bit Name	Function
15 to 13	CSIGnPRS [2:0]	Selects the value of the prescaler.
12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CSIGnBRS [11:0]	Selects the transfer clock frequency. Settings of the CSIGnBRS[11:0] bits are valid only in master mode. They are ignored in slave mode.

CSIGn PRS2	CSIGn PRS1	CSIGn PRS0	Prescaler Output (PRSOUT)
0	0	0	PCLK (master mode)
0	0	1	PCLK / 2 (master mode)
0	1	0	PCLK / 4 (master mode)
0	1	1	PCLK / 8 (master mode)
1	0	0	PCLK / 16 (master mode)
1	0	1	PCLK / 32 (master mode)
1	1	0	PCLK / 64 (master mode)
1	1	1	External clock via CSIGTSCK (slave mode)

CSIGnBRS [11:0]	Transfer Clock Frequency of CSIGTSCK
0	BRG is stopped
1	PCLK / (2^{α} × 1 × 2)
2	PCLK / (2^{α} × 2 × 2)
3	PCLK / (2^{α} × 3 × 2)
4	PCLK / (2^{α} × 4 × 2)
...	...
4095	PCLK / (2^{α} × 4095 × 2)

Note: α = 0 to 6 (value set by CSIGnPRS[2:0])

CAUTION

When setting this register, see **Table 15.24, List of Cautions when Configuring the Registers**.

15.3.5 CSIGnSTR0 — CSIGn Status Register 0

This register indicates the status of the CSIG.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <CSIGn_base> + 0004_H

Value after reset: $0000\ 0010_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CSIGn TSF	—	—	—	CSIGn DCE	—	CSIGn PE	CSIGn OVE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.16 CSIGnSTR0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	CSIGnTSF	Transfer Status Flag 0: Idle state 1: Communication is in progress or being prepared The timing to set or clear this bit is as follows: **Master Mode** / **Timing to Set** / **Timing to Clear** Tx-only mode: Writing to transmission register: Within a half clock cycle from the last serial clock edge Tx/Rx mode: Writing to transmission register: Within a half clock cycle from the last serial clock edge Rx-only mode: Reading from reception register: Within a half clock cycle from the last serial clock edge **Slave Mode** / **Timing to Set** / **Timing to Clear** Tx-only mode: Writing to transmission register: Within a half clock cycle from the last serial clock edge Tx/Rx mode: Writing to transmission register: Within a half clock cycle from the last serial clock edge Rx-only mode: CSIGTSCK input: Within a half clock cycle from the last serial clock edge
6 to 4	Reserved	When read, the value after reset is returned.
3	CSIGnDCE	Data Consistency Check Error Flag 0: No data consistency check error detected 1: Data consistency check error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnDCEC. When setting to 1 due to data consistency error detection and clearing to 0 by CSIGnSTCR0.CSIGnDCEC occur simultaneously, setting to 1 due to data consistency error detection takes precedence. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1, or from 1 to 0.
2	Reserved	When read, the value after reset is returned.

Table 15.16 CSIGnSTR0 Register Contents (2/2)

Bit Position	Bit Name	Function
1	CSIGnPE	Parity Error Flag 0: No parity error detected 1: Parity error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnPEC. When setting to 1 due to parity error detection and clearing to 0 by writing to CSIGnSTCR0.CSIGnPEC occur simultaneously, setting to 1 due to parity error detection takes precedence. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1, or from 1 to 0.
0	CSIGnOVE	Overrun Error Flag 0: No overrun error detected 1: Overrun error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnOVEC. When setting to 1 due to overrun error detection and clearing to 0 by writing to CSIGnSTCR0.CSIGnOVEC occur simultaneously, setting to 1 due to overrun error detection takes precedence. This bit is initialized by the value of CSIGnCTL0.CSIGnPWR changes from 0 to 1, or from 1 to 0.

CAUTION

When setting this register, see **Table 15.24, List of Cautions when Configuring the Registers**.

15.3.6 CSIGnSTCR0 — CSIGn Status Clear Register 0

This register clears the status flags of the CSIGnSTR0 status register.

Access: This register can be read or written in 16-bit units.
When read, the value 0000_H is always returned.

Address: <CSIGn_base> + 0008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CSIGn DCEC	—	CSIGn PEC	CSIGn OVEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 15.17 CSIGnSTCR0 Register Contents

Bit Position	Bit Name	Function
15 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	CSIGnDCEC	Controls the data consistency error flag clear command. 0: No operation. Read value is always 0. 1: Clears the data consistency check error flag (CSIGnSTR0.CSIGnDCE).
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CSIGnPEC	Controls the parity error flag clear command. 0: No operation. Read value is always 0. 1: Clear parity error flag (CSIGnSTR0.CSIGnPE).
0	CSIGnOVEC	Controls the overrun error flag clear command. 0: No operation. Read value is always 0. 1: Clear overrun error flag (CSIGnSTR0.CSIGnOVE).

15.3.7 CSIGnBCTL0 — CSIGn Rx-Only Mode Control Register 0

This register enables/disables the data transfer in Rx-only mode.

Access: This register can be read or written in 1-bit and 8-bit units.

Address: <CSIGn_base> + 1000_H

Value after reset: 01_H

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

Table 15.18 CSIGnBCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	CSIGnSCE	Disables/enables the start of the next data reception by reading CSIGnRX0. 0: Next reception disabled 1: Next reception enabled For details, see **Section 15.5.4.2, Receive-Only Mode**.

CAUTION

When setting this register, see **Table 15.24, List of Cautions when Configuring the Registers**.

15.3.8 CSIGnCFG0 — CSIGn Configuration Register 0

This register configures the communication protocols such as data length, parity, transfer direction, clock phase, and data phase.

Access: This register can be read or written in 32-bit units.

Address: <CSIGn_base> + 1010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CSIGnPS[1:0]		CSIGnDLS[3:0]				—	—	—	—	—	CSIGn DIR	—	CSIGn DAP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R/W

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.19 CSIGnCFG0 Register Contents (1/2)

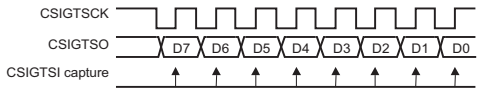
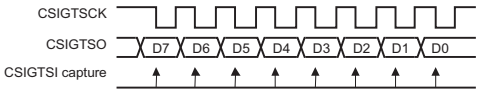
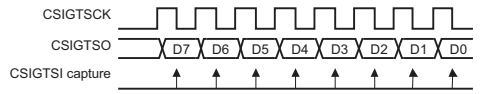
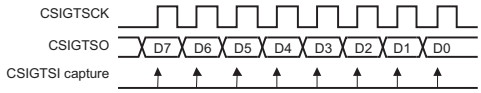
Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29, 28	CSIGnPS[1:0]	Specifies parity.

CSIGn PS1	CSIGn PS0	Transmission	Reception
0	0	Does not transmit any parity bit.	Does not wait for reception of the parity bit.
0	1	Adds a parity bit fixed to 0.	Waits for reception of the parity bit but does not evaluate it.
1	0	Adds the odd parity bit.	Waits for the odd parity bit.
1	1	Adds the even parity bit.	Waits for the even parity bit.

27 to 24	CSIGnDLS [3:0]	Specifies data length. 0: Data length is 16 bits 1: Data length is 1 bit 2: Data length is 2 bits ... 15: Data length is 15 bits **CAUTION** Do not set bits CSIGnCFG0.CSIGnDLS[3:0] to values 1 to 6 when the extended data length function is disabled with bit CSIGnCTL1.CSIGnEDLE set to 0. Transmitting two consecutive data with a data length of less than 7 bits is prohibited.
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	CSIGnDIR	Selects the serial data direction. 0: Data is transmitted/received with MSB first 1: Data is transmitted/received with LSB first
17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 15.19 CSIGnCFG0 Register Contents (2/2)

Bit Position	Bit Name	Function
16	CSIGnDAP	Data Phase Selection Used in conjunction with CSIGnCTL1.CSIGnCKR bit to select the data phase. Refer to the following table for the selectable clock phases and data phases. .

CSIGnCTL1.CSIGnCKR	CSIGnDAP	Clock and Data Phase Selection
0	0	
0	1	
1	0	
1	1	

15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

When setting this register, see **Table 15.24, List of Cautions when Configuring the Registers**.

15.3.9 CSIGnTX0W — CSIGn Transmission Register 0 for Word Access

This register stores the transmission data and specifies the extended data length.

Access: This register can be read or written in 32-bit units.

Address: <CSIGn_base> + 1004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CSIGnEDL	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnTX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.20 CSIGnTX0W Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	CSIGnEDL	Specifies the extended data length. 0: Normal operation 1: Extended data length enabled The associated data is transmitted as 16-bit data. This bit can only be set if CSIGnCTL1.CSIGnEDLE = 1.
28 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	CSIGnTX[15:0]	Data to be transmitted

CAUTION

When setting this register, see **Table 15.24, List of Cautions when Configuring the Registers**.

15.3.10 CSIGnTX0H — CSIGn Transmission Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 of CSIGnTX0W register.
The settings specified by the upper 16 bits of CSIGnTX0W are applied to the transmission.

Access: This register can be read or written in 16-bit units.

Address: <CSIGn_base> + 1008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnTX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.21 CSIGnTX0H Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIGnTX[15:0]	Data to be transmitted

CAUTION

When setting this register, see **Table 15.24, List of Cautions when Configuring the Registers**.

15.3.11 CSIGnRX0 — CSIGn Reception Register 0

This register stores the received data.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <CSIGn_base> + $100C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnRX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15.22 CSIGnRX0 Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIGnRX [15:0]	Received Data These bits are initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0. When reading, the values of these bits must be read at least 1 clock before the generation of CSIGTIR interrupt.

CAUTION

When setting this register, see **Table 15.24, List of Cautions when Configuring the Registers**.

15.3.12 CSIGnEMU — CSIGn Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read or written in 1-bit and 8-bit units.
Write to this register when EPC.SVSTOP = 0.

Address: <CSIGn_base> + 0018_H

Value after reset: 00_H

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

Table 15.23 CSIGnEMU Register Contents

Bit Position	Bit Name	Function
7	CSIGnSVSDIS	Selects whether to continue or stop transmit/receive operation during debugging. • When the EPC.SVSTOP bit is set to 0: Continues transmit/receive operation regardless of the setting of this bit. • When the EPC.SVSTOP bit is set to 1: 0: Stops transmit/receive operation. 1: Continues transmit/receive operation.
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

When setting this register, see **Table 15.24, List of Cautions when Configuring the Registers**.

15.3.13 List of Cautions

Table 15.24 List of Cautions when Configuring the Registers

Register Name	Bit Name	Cautions
CSIGnCTL0	CSIGnPWR	If this bit is cleared during communication, ongoing communication is aborted. After the communication is aborted, it is necessary to restart the communication.
CSIGnCTL0	CSIGnTXE CSIGnRXE	Do not modify any of these bits while CSIGnCTL0.CSIGnPWR = 0. (These bits can be modified simultaneously with the CSIGnCTL0.CSIGnPWR bit.) Do not modify these bits while CSIGnSTR0.CSIGnTSF = 1, because the specified operation is not guaranteed if ongoing communication is aborted.
CSIGnCTL0	CSIGnMBS	When writing, be sure to set this bit to 1. (The value after reset is “0”.) This bit must be modified simultaneously with CSIGnCTL0.CSIGnPWR bit.
CSIGnCTL1	CSIGnCKR	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnCTL1	CSIGnSLIT CSIGnEDLE CSIGnDCS CSIGnHSE	Modification of these bits is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnCTL1	CSIGnLBM	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting of this bit is prohibited in slave mode.
CSIGnCTL1	CSIGnSSE	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting this bit to 1 is prohibited in master mode.
CSIGnCTL1	CSIGnSIT	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. This bit is only valid in master mode. In slave mode, no delay is generated.
CSIGnCTL2	CSIGnPRS[2:0] CSIGnBRS[11:0]	Modification of these bits is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting of the maximum transfer clock frequency is as follows. • Master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4) • Slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)
CSIGnSTR0	CSIGnTSF	Writing to this bit is prohibited, and only reading is permitted.
CSIGnSTR0	CSIGnDCE CSIGnPE CSIGnOVE	Writing to these bits is prohibited, and only reading is permitted. These bits are initialized when CSIGnCTL0.CSIGnPWR = 0 → 1 or CSIGnCTL0.CSIGnPWR = 1 → 0.
CSIGnBCTL0	CSIGnSCE	Write to this bit before CSIGnRX0 is read. Fix the CSIGnSCE bit to 0 when the transfer mode is transmit mode or transmit/receive mode.
CSIGnCFG0	CSIGnPS[1:0] CSIGnDLS[3:0] CSIGnDIR CSIGnDAP	Modification of these bits is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnTX0W	CSIGnEDL	This bit is valid only when CSIGnCTL1.CSIGnEDLE = 1.
CSIGnTX0W CSIGnTX0H		Write access to these bits are prohibited when CSIGnCTL0.CSIGnTXE = CSIGnCTL0.CSIGnRXE = 0.
CSIGnRX0		These bits are initialized when CSIGnCTL0.CSIGnPWR = 0 → 1 or CSIGnCTL0.CSIGnPWR = 1→ 0. Read access to this bit is prohibited when CSIGnCTL0.CSIGnTXE = CSIGnCTL0.CSIGnRXE = 0.
CSIGnEMU	CSIGnSVSDIS	Modification of this bit is prohibited while SVSTOP = 1.

15.4 Interrupt Sources

CSIG can generate the following interrupts:

- INTCSIGTIC (communication status interrupt)
- INTCSIGTIR (reception status interrupt)
- INTCSIGTIRE (communication error interrupt)

15.4.1 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by half cycle of the transmission clock CSIGTSCK. This function is not available in slave mode.

The delay is specified by setting CSIGnCTL1.CSIGnSIT = 1. (The setting of the CSIGnSIT bit is invalid in slave mode.)

The following example illustrates the interrupt delay function, assuming a setting of CSIGnCTL1.CSIGnSIT = 1 (interrupt delay enabled), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits).

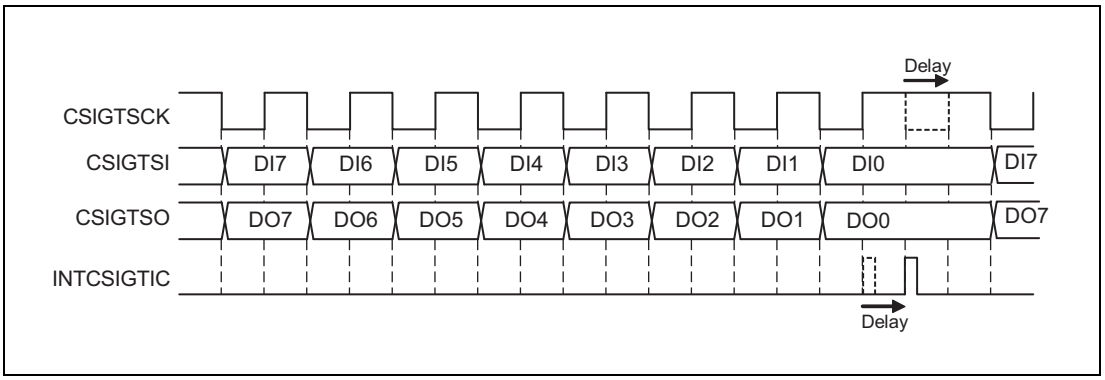


Figure 15.2 Interrupt Delay Function (CSIGnCTL1.CSIGnSIT = 1)

15.4.2 INTCSIGTIC (Communication Status Interrupt)

This interrupt is normally generated after every data transfer. It can be used to trigger a DMA for writing new transmission data to register CSIGnTX0W or CSIGnTX0H.

The following example assumes master mode and a setting of CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits), and CSIGnCTL1.CSIGnSLIT = 0 (normal interrupt timing).

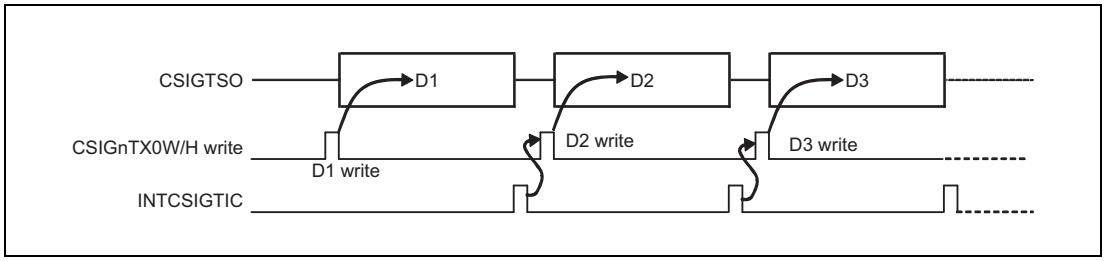


Figure 15.3 Generation of INTCSIGTIC after Communication (CSIGnCTL1.CSIGnSLIT = 0)

However, INTCSIGTIC can also be set up to occur when the CSIGnTX0W/H register is empty and available for receiving the next data. This is specified by setting CSIGnCTL1.CSIGnSLIT = 1.

This mode allows more efficient data transfers.

The effect is illustrated in the figure below.

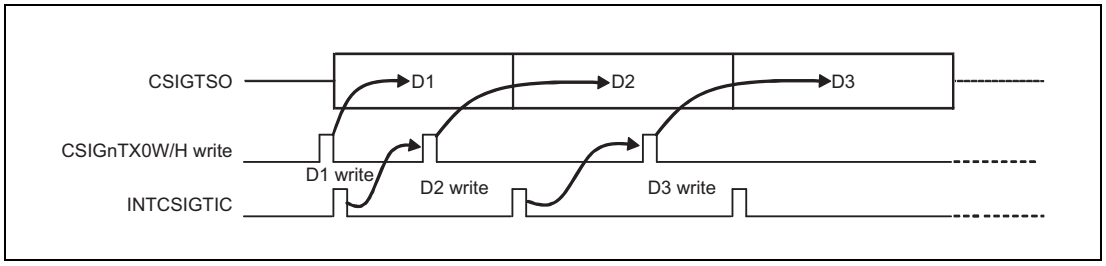


Figure 15.4 Generation of INTCSIGTIC at the Beginning of Communication

15.4.3 INTCSIGTIR (Reception Status Interrupt)

This interrupt is generated in receive-only and transmit/receive mode after data has been received and is available in the reception register. It can be used to trigger a DMA for reading the received data from CSIGnRX0 register.

The following example assumes master mode and a setting of CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits).

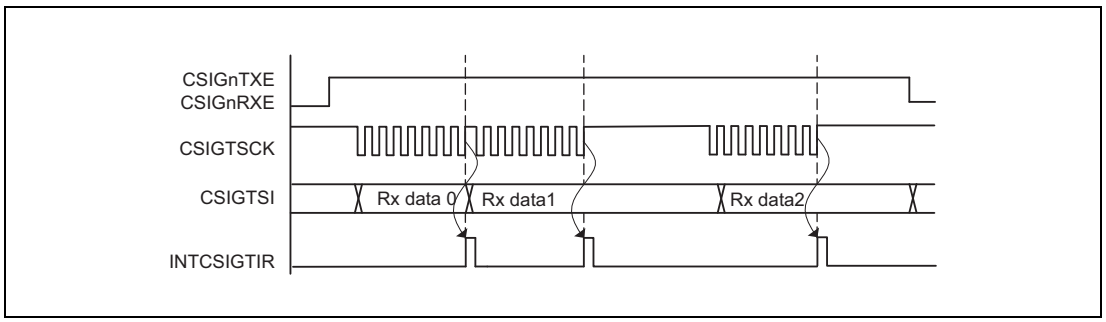


Figure 15.5 Generation of INTCSIGTIR

15.4.4 INTCSIGTIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

Table 15.25 Data Error Types

Error Type	Communication Status After Error Interrupt	Note
Parity error	Interrupt is generated and communication continues.	—
Data consistency error	Interrupt is generated and communication continues.	—
Overrun error*1	When CSIGnCTL1.CSIGnHSE = 0 (handshake function disabled) in slave mode, interrupt is generated and communication continues.	When CSIGnCTL1.CSIGnHSE = 1 (handshake function enabled) in slave mode, communication stops due to the handshake. An interrupt is not generated and an overrun error does not occur.

Note 1. In master mode, overrun errors do not occur.
In slave mode, communication cannot be stopped.

The type of error that caused the generation of INTCSIGTIRE is indicated in register CSIGnSTR0.

For details about the various error types, see **Section 15.5.10, Error Detection**.

15.5 Operation

15.5.1 Master/Slave Mode

Whether CSIG operates in master mode or in slave mode depends on the setting of bits CSIGnCTL2.CSIGnPRS[2:0]. If master mode is selected, the source of the transmission clock must be selected as well.

15.5.1.1 Master Mode

In master mode, the serial transmission clock is generated by the built-in baud rate generator (BRG) and supplied to the slave via signal CSIGTSCK.

Master mode is enabled by setting bits CSIGnCTL2.CSIGnPRS[2:0] to values other than 111_B. In master mode, the frequency of BRG can be configured by setting bits CSIGnCTL2. CSIGnPRS[2:0] and bits CSIGnCTL2.CSIGnBRS[11:0].

The default level of CSIGTSCK depends on the CSIGTSCK clock inversion function bit; it is high when CSIGnCTL1.CSIGnCKR = 0, and is low when CSIGnCTL1.CSIGnCKR = 1.

The example below shows the communication in master mode for 8-bit data, CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

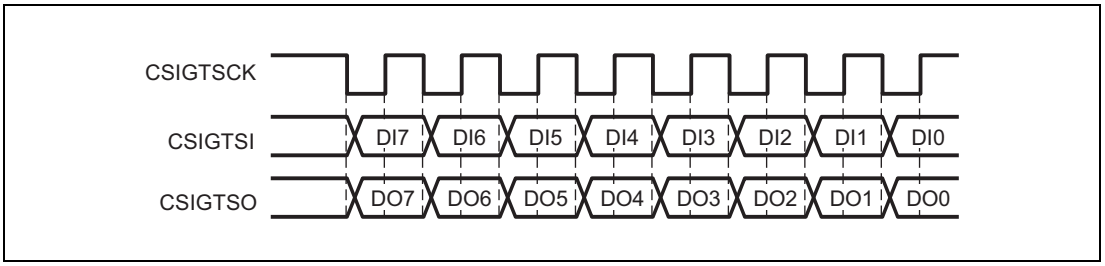


Figure 15.6 Transmission/Reception in Master Mode

15.5.1.2 Slave Mode

In slave mode, another device is the communication master. The external clock is supplied via the signal CSIGTSCK. Transmit/receive operation starts as soon as a clock signal is detected.

Slave mode is selected by setting CSIGnCTL2.CSIGnPRS[2:0] to 111_B.

NOTE

When using slave mode, disable the baud rate generator (BRG) by setting bits CSIGnCTL2.CSIGnBRS[11:0] to 000_H.

The example below shows the communication in slave mode for 8-bit data, CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

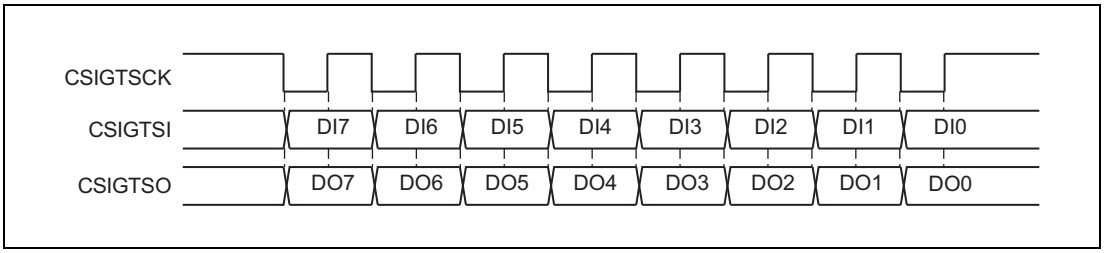


Figure 15.7 Transmission/Reception in Slave Mode

15.5.2 Master/Slave Connections

15.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

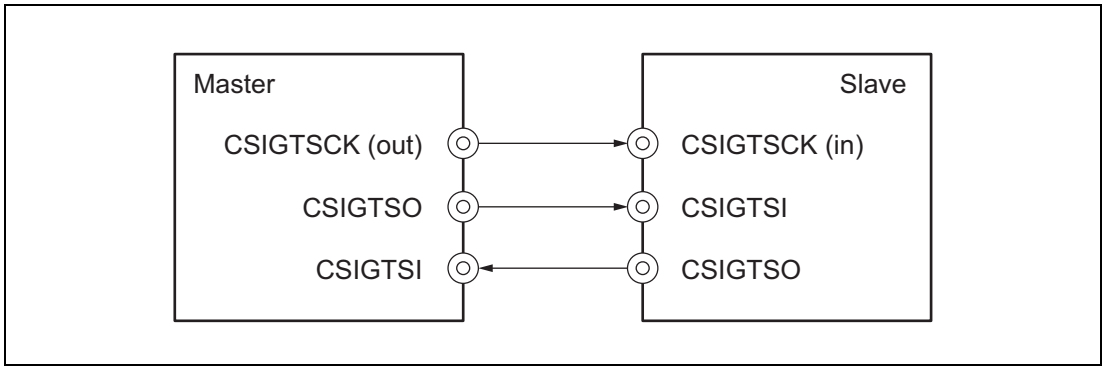


Figure 15.8 Direct Master/Slave Connection

15.5.2.2 One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this case, the master must provide one slave select (SS) signal to each of the slaves. This signal is connected to the slave select input $\overline{\text{CSIGTSSI}}$ of the slave.

The $\overline{\text{CSIGTSSI}}$ signal can be enabled or disabled by the CSIGnCTL1.CSIGnSSE bit.

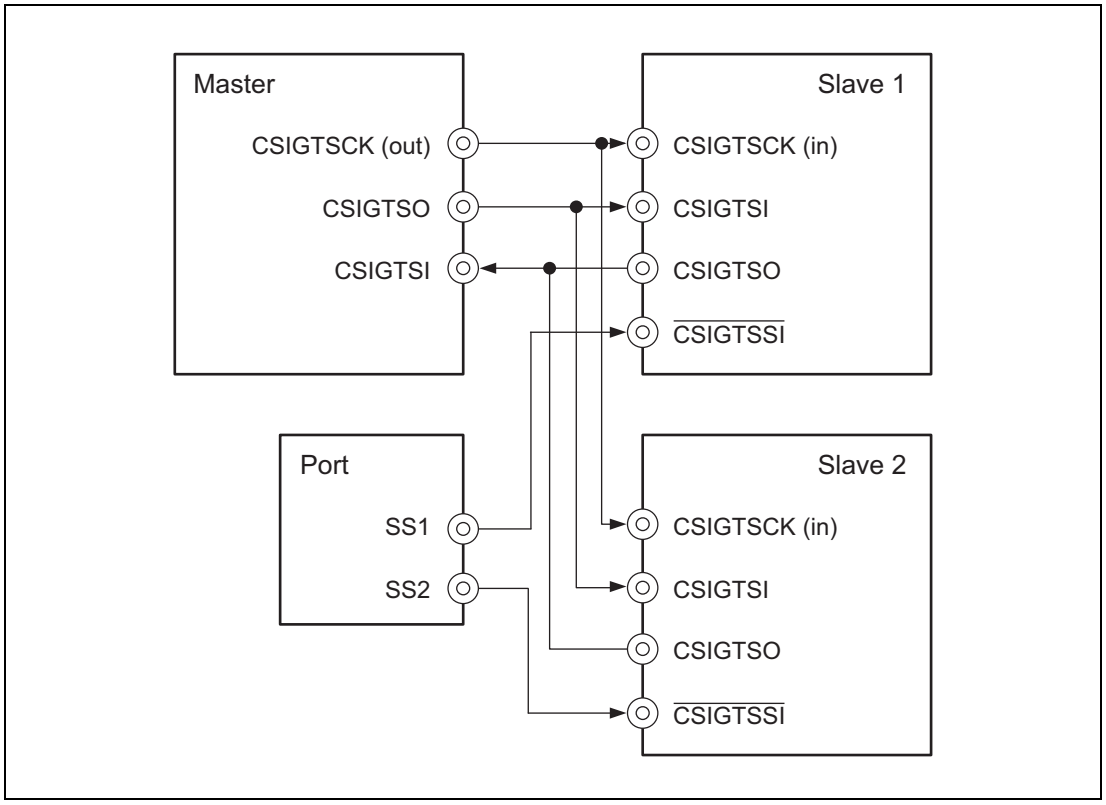


Figure 15.9 Master to Multiple Slaves Connection

A slave is selected (enabled) when its $\overline{\text{CSIGTSSI}}$ signal is low.

If it is not selected, the slave will neither receive nor transmit data. In addition, when transmit-only mode or transmit/receive mode is set (CSIGnCTL0.CSIGnTXE = 1), the CSIGTSO output buffer of the slave which is not selected is disabled and set to input mode in order to avoid interference with the outputs of other selected slaves.

15.5.3 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the CSIGnPRS[2:0] and CSIGnBRS[11:0] bits in the CSIGnCTL2 register.

The following figure shows a block diagram of the BRG.

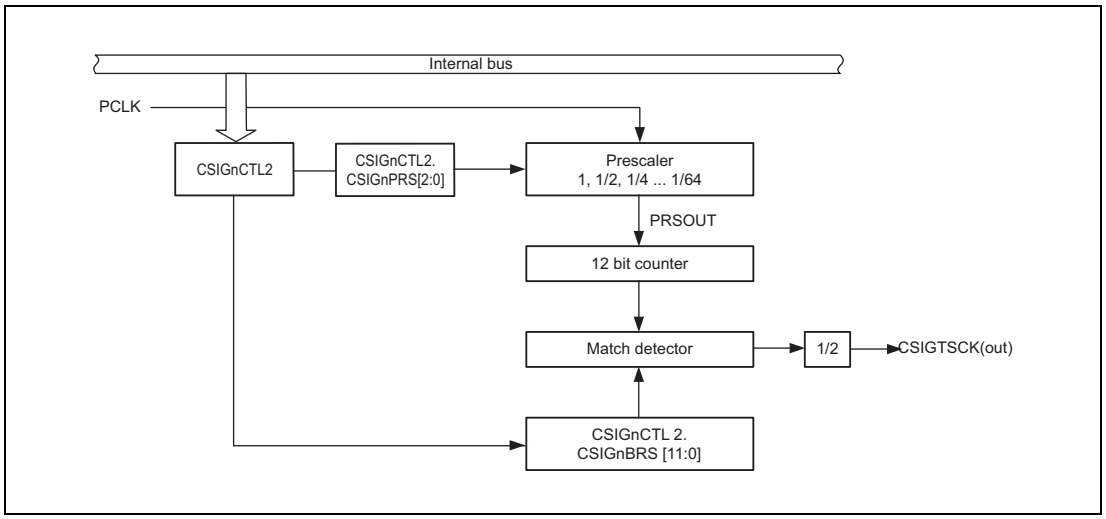


Figure 15.10 BRG Block Diagram

Setting CSIGnCTL2.CSIGnBRS[11:0] to 000_H disables the BRG.

Transfer clock frequency calculation

The transfer clock frequency in master mode is calculated as:

Transfer clock frequency (CSIGTSCK) = PCLK / (division ratio of PCLK) = PCLK / ($2^{\alpha} \times k \times 2$)
where:

α = CSIGnCTL2.CSIGnPRS[2:0] = 0 to 6

k = CSIGnCTL2.CSIGnBRS[11:0] = 1 to 4095

Transfer clock frequency upper and lower limits

When specifying the transfer clock frequency, please note the following:

- For the maximum transfer clock frequency of this product in master mode or slave mode, refer to the CSIG timing shown in the electrical characteristics. In addition, in either mode, specify a frequency within the defined range.
- The minimum transfer clock frequency in master mode and slave mode is PCLK/524160.
- The maximum transfer clock frequency is as follows:
  - In master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)

15.5.4 Data Transfer Modes

15.5.4.1 Transmit-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 0 places CSIG in transmit-only mode. Transmission starts when data to be transmitted is written in the CSIGnTX0W or CSIGnTX0H register.

CAUTION

In case transmit-only mode has been entered after any reception mode, the data in the CSIGnRX0 buffer becomes undefined after completion of the first transmission. Consequently the reception register CSIGnRX0 has to be read before changing to transmit-only mode.

15.5.4.2 Receive-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 0 and CSIGnCTL0.CSIGnRXE = 1 places CSIG in receive-only mode.

In master mode, reception starts when dummy data is read from the CSIGnRX0 register.
All subsequent receptions are triggered by reads from the CSIGnRX0 register, as long as CSIGnBCTL0.CSIGnSCE = 1.

Moreover, CSIGnBCTL0.CSIGnSCE has to be set to 0 before reading the last received data from CSIGnRX0.

The recommended procedure is:

1. Set CSIGnBCTL0.CSIGnSCE = 1.
2. Read CSIGnRX0 (dummy data).
3. Wait for the reception interrupt INTCSIGTIR.
4. Read CSIGnRX0 (received data).
   In case more data receptions follow at step 3, continue to read until all data is received.
   Before reading the last received data from CSIGnRX0, set CSIGnBCTL0.CSIGnSCE = 0.

In slave mode, reception starts when the communication clock CSIGTSCK from the master is received. In this case, it is not necessary to read data from the CSIGnRX0 register of the slave.

NOTE

In slave mode, any previously received data must be read from the reception register CSIGnRX0 in order to avoid any overwrite situation.

15.5.4.3 Transmit/Receive Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 1 places CSIG in transmit/receive mode.

Data transfer (transmission and reception) starts when data to be transmitted is written to the CSIGnTX0W or CSIGnTX0H register.

15.5.5 Data Length Selection

15.5.5.1 Data Length Selection without Extended Length

Transmission data length is selectable from 7 to 16 bits using the CSIGnDLS[3:0] bits in the CSIGnCFG0 register. The examples below show the communication with MSB first (CSIGnCFG0.CSIGnDIR = 0):

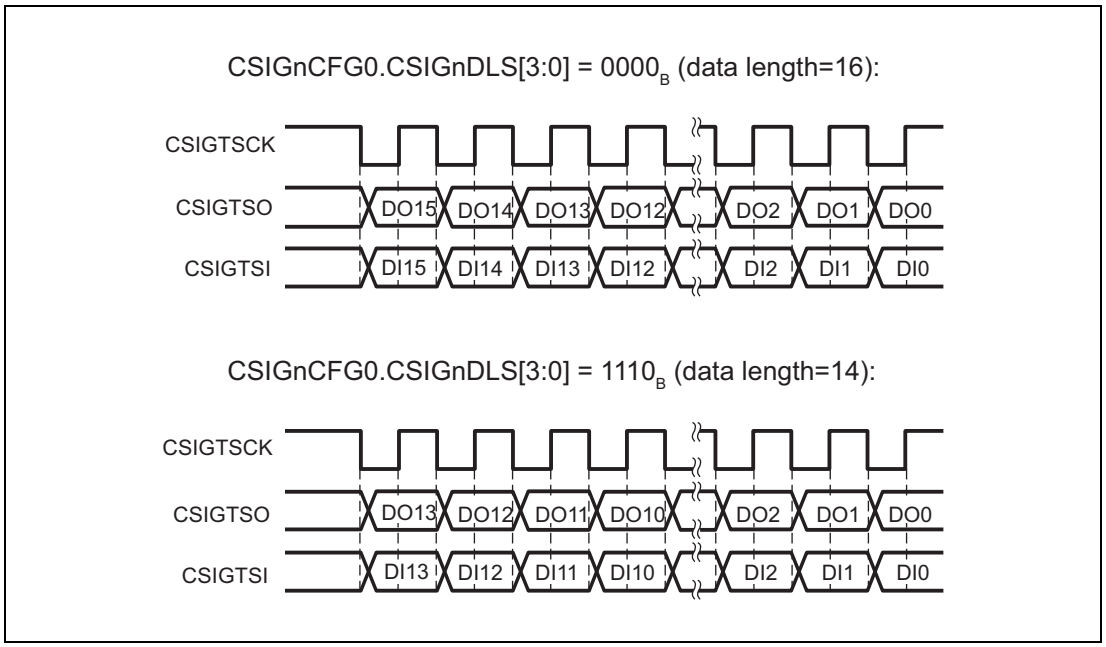


Figure 15.11 Data Length Selection Function

15.5.5.2 Data Length Selection with Extended Data Length

If the data to be sent/received exceeds 16 bits, the extended data length (EDL) function can be used.

The EDL function is enabled by setting the CSIGnCTL1.CSIGnEDLE bit to 1.

The following describes how the EDL function works and how to specify the EDL setting.

- The data has to be broken into 16-bit blocks plus remainder. For example, data of 42 bits would be broken into two 16-bit blocks plus 10 bits.
- The bit length of the remainder is set as “data length” in CSIGnCFG0.CSIGnDLS[3:0] bits.
- Set the CSIGnTX0W.CSIGnEDL bit to 1 to transmit the 16-bit blocks. In this case, the data written to the CSIGnTX0W register is sent as 16-bit data regardless of the setting of the CSIGnCFG0.CSIGnDLS[3:0] bits.
- The transfer completes after the data with the specified data length (the remainder when CSIGnTX0W.CSIGnEDL = 0) has been sent.

Example

Example of sending 40 bits of data, $123456789A_H$:

40 bits are split into 2 × 16 bits plus 8 bits.

- Initialize CSIGnCFG0.CSIGnDLS[3:0] = 8_D.
- To transmit the data $123456789A_H$ with MSB first, write the following sequence to CSIGnTX0W:
  - $2000\ 1234_H$ (CSIGnTX0W.CSIGnEDL = 1)
  - $2000\ 5678_H$ (CSIGnTX0W.CSIGnEDL = 1)
  - $0000\ 009A_H$ (CSIGnTX0W.CSIGnEDL = 0)

The following figure illustrates the timing.

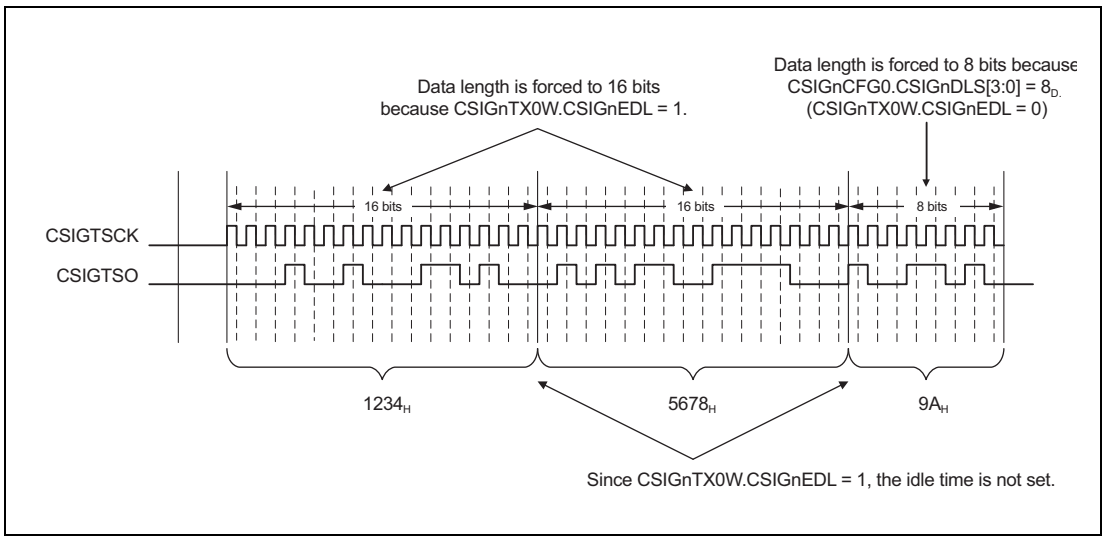


Figure 15.12 EDL Timing Diagram

NOTES

1. Data length with less than 7 bits can be set only when EDL mode is used.
2. It is not possible to transmit two consecutive data with a data length of less than 7 bits.
3. If parity is enabled, the parity bit is added after the last bit.
4. Example for setting the data direction:
   - Data to be transmitted: 123456_H
   - MSB first:
     Set CSIGnCFG0.CSIGnDIR to 0.
     Write $2000\ 1234_H$ to CSIGnTX0W (EDL bit = 1).
     Write $0000\ 0056_H$ to CSIGnTX0W (EDL bit = 0).
   - LSB first:
     Set CSIGnCFG0.CSIGnDIR to 1.
     Write $2000\ 3456_H$ to CSIGnTX0W (EDL bit = 1).
     Write $0000\ 0012_H$ to CSIGnTX0W (EDL bit = 0).
5. EDL mode cannot be used in the slave mode configured for receive-only mode. (CSIGnCTL2.CSIGnPRS[2:0] = 111_B, CSIGnCTL0.CSIGnTXE = 0, CSIGnCTL0.CSIGnRXE = 1)

15.5.6 Serial Data Direction Selection Function

The serial data direction is selectable using the CSIGnDIR bit in the CSIGnCFG0 register. The examples below show the communication for 8-bit data (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B):

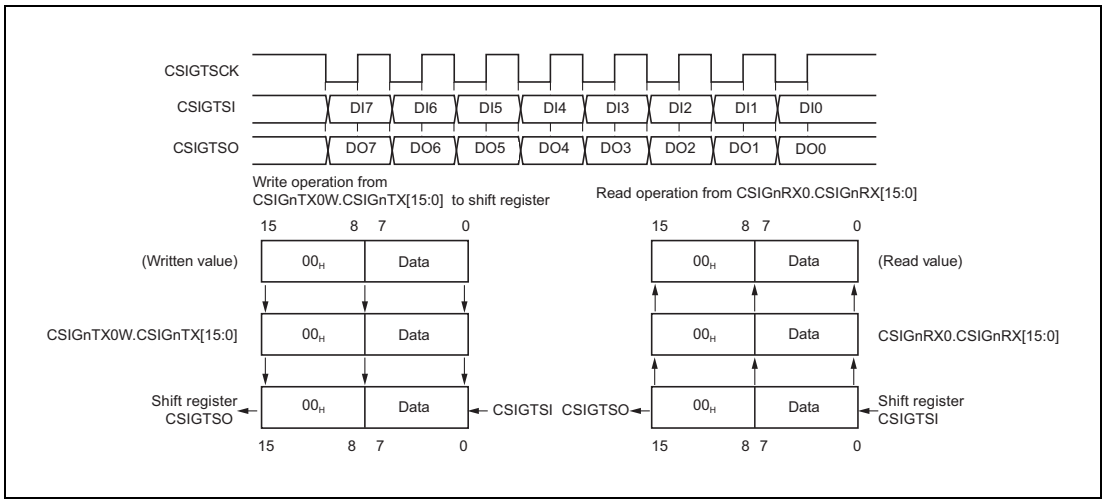


Figure 15.13 Serial Data Direction Select Function — MSB First (CSIGnDIR = 0)

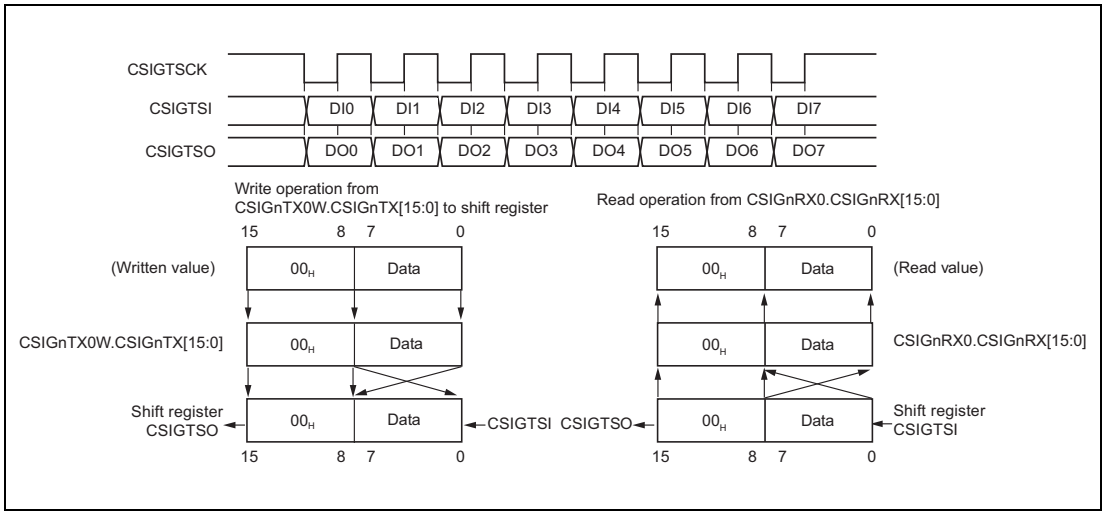


Figure 15.14 Serial Data Direction Select Function — LSB First (CSIGnDIR = 1)

15.5.7 Communication in Slave Mode

The following figure illustrates the communication signals and timing in slave mode.

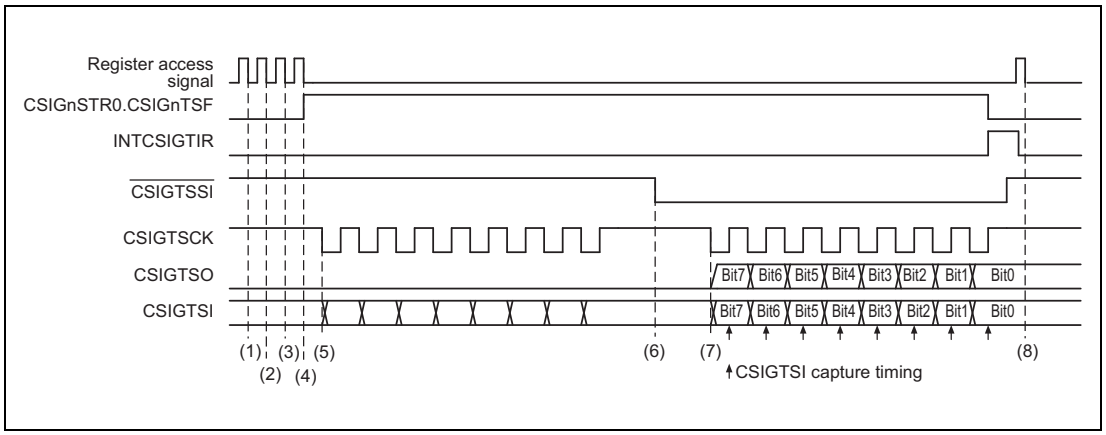


Figure 15.15 Rx/Tx Communication Timing in Slave Mode

1. Slave mode is selected (CSIGnCTL2.CSIGnPRS[2:0] = 111_B), the $\overline{\text{CSIGTSSI}}$ signal is enabled (CSIGnCTL1.CSIGnSSE = 1) and the clock phase is set to the high level (CSIGnCTL1.CSIGnCKR = 0).
2. Data length is 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B).
   Data direction is set to MSB first (CSIGnCFG0.CSIGnDIR = 0).
3. CSIG is set to transmit/receive mode (CSIGnCTL0.CSIGnPWR = 1, CSIGnCTL0.CSIGnTXE = 1, CSIGnCTL0.CSIGnRXE = 1).
4. When transfer data is written to the transmission register CSIGnTX0H, the transfer status flag CSIGnSTR0.CSIGnTSF is automatically set and the CSIG waits until signal $\overline{\text{CSIGTSSI}}$ goes low.
5. While signal $\overline{\text{CSIGTSSI}}$ is high, transmission/reception is not started even if the serial clock is input. CSIGTSO retains the values and input at CSIGTSI is ignored.
6. As soon as $\overline{\text{CSIGTSSI}}$ falls to low level, CSIGTSO is enabled.
7. If the serial clock is input to the CSIG while $\overline{\text{CSIGTSSI}}$ is low, transfer data is sent to CSIGTSO in synchronization with the serial clock, and simultaneously, data is received from CSIGTSI.
8. The register CSIGnRX0 is read.

15.5.8 Handshake Function

CSIG features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by the CSIGnCTL1.CSIGnHSE bit. For handshake, the signals CSIGTRYI and CSIGTRYO are used.

The busy timing depends on the setting of the data phase selection bit CSIGnCFG0.CSIGnDAP.

15.5.8.1 Slave Mode

If CSIGnCTL1.CSIGnHSE = 1, a low-level CSIGTRYO signal is output when the slave becomes busy. This happens when previously received data is still in the CSIGnRX0 register, and the new data cannot be copied from the shift register to CSIGnRX0 (CSIGnRX0 full condition).

The following examples assume a data length of 8 bits.

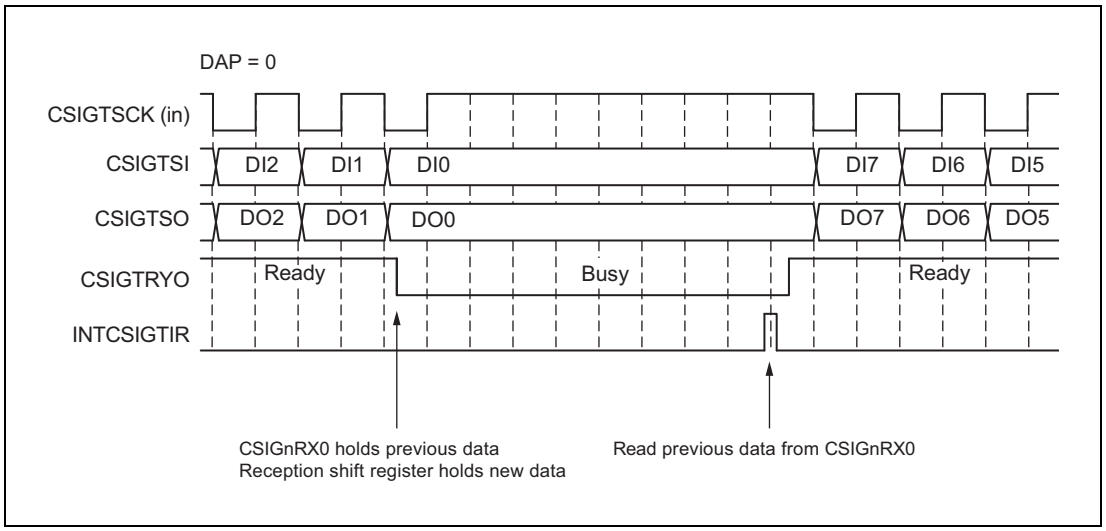


Figure 15.16 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 0)

While the slave is busy, the master has to wait (i.e. suspend the transmission clock). The slave sets CSIGTRYO to high (“ready”) as soon as the read from the reception register CSIGnRX0 completes.

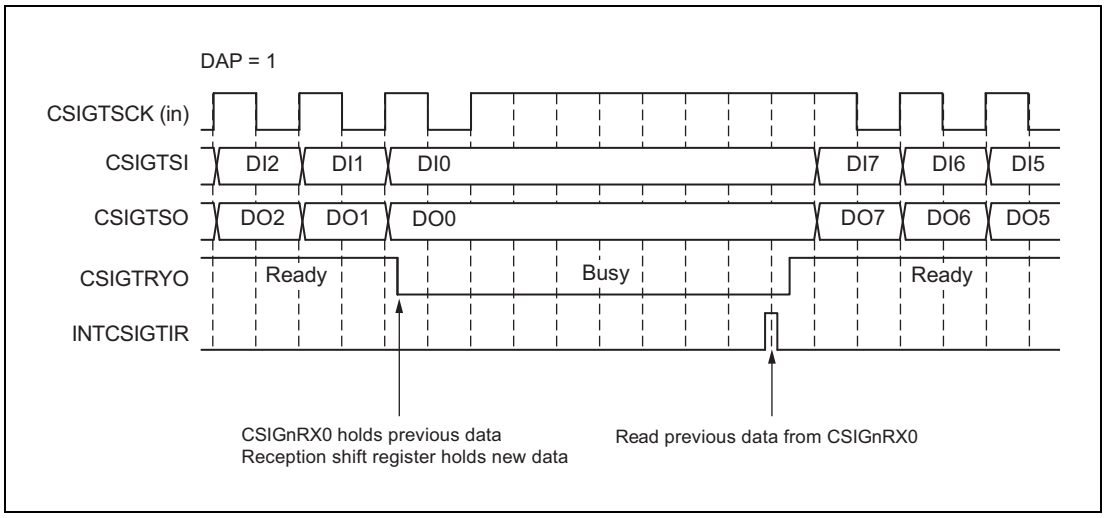


Figure 15.17 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 1)

15.5.8.2 Master Mode

When the master detects low level of the CSIGTRYI while CSIGnCTL1.CSIGnHSE = 1, subsequent transfers are put on hold, and the master goes into wait state and suspends the CSIGTSCK clock.

The CSIGTRYI level is checked at each half clock cycle of CSIGTSCK.

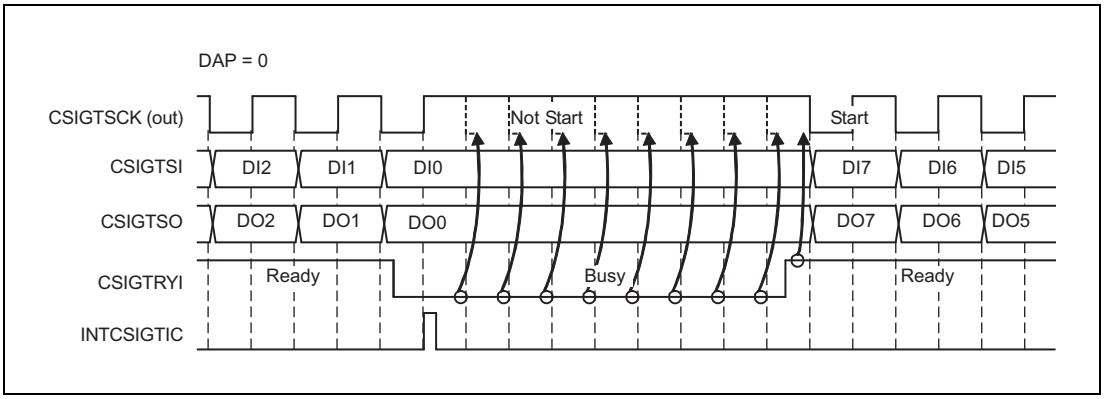


Figure 15.18 Master's Reaction to CSIGTRYI (CSIGnCFG0.CSIGnDAP = 0)

If the CSIGTRYI low signal from the slave is received while data transfer is in progress, the serial clock is suspended after the transfer is complete.

The master resumes the communication as soon as CSIGTRYI becomes high (slave is "ready").

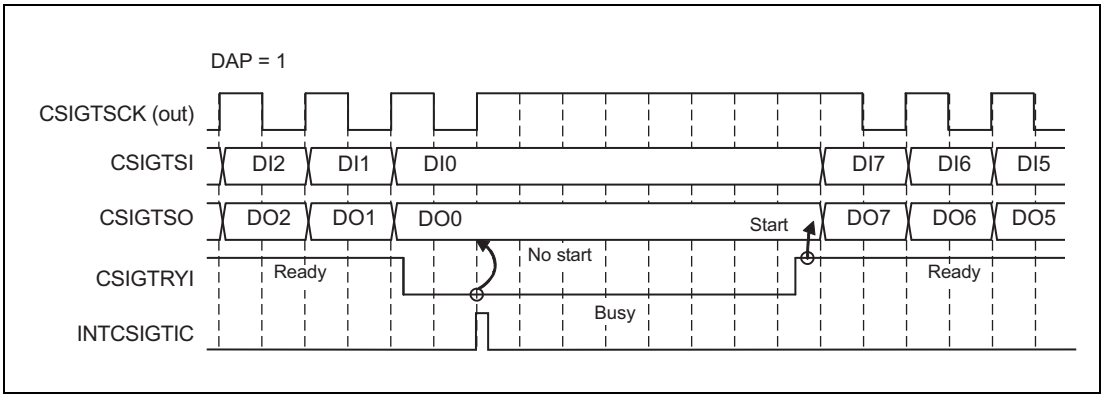


Figure 15.19 Master's Reaction to CSIGTRYI (CSIGnCFG0.CSIGnDAP = 1)

CAUTION

If multiple slaves are connected, the master must only detect the CSIGTRYI signal of the slave it has selected for communication.

CSIGTRYI of the master must be pulled down by the slave before the next transfer starts. Even if the signal is pulled down by the slave during the transfer, the transfer will continue until it completes.

15.5.9 Loop-Back Mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active (CSIGnCTL1.CSIGnLBM = 1), the transmit and receive signals are internally connected, as shown in the figures below. The signals CSIGTSCK, CSIGTSO, and CSIGTSI are disconnected from the ports. In addition, the CSIGTSO output level is fixed to low, and CSIGTSCK is set to reset level (High). The rest of CSIG works as in normal operation.

In order to test the CSIG, set the loop-back mode and carry out normal transfer operations. Then check that the received data is the same as the transmitted data.

Table 15.26 Output Level of Pins

Pin	Output Level
CSIGTSCK(out)	High level
CSIGTSO	Low level (not dependent on the previous value)
Interrupt	Normal function
CSIGTRYO	Normal function (Low level)

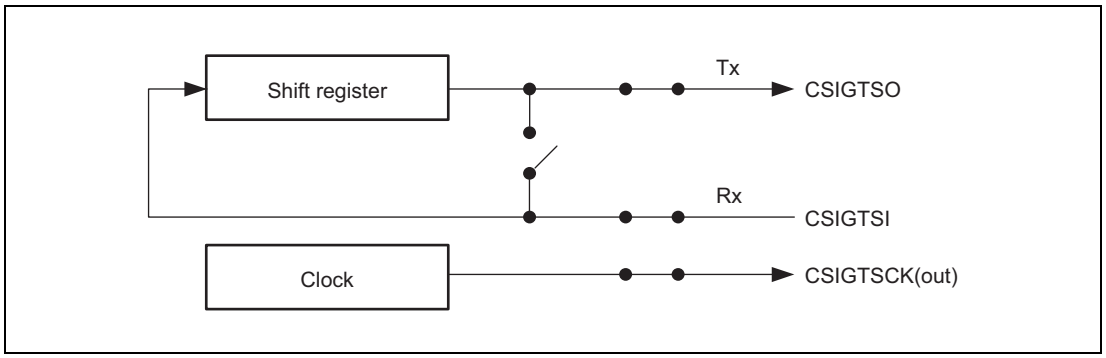


Figure 15.20 Normal Operation

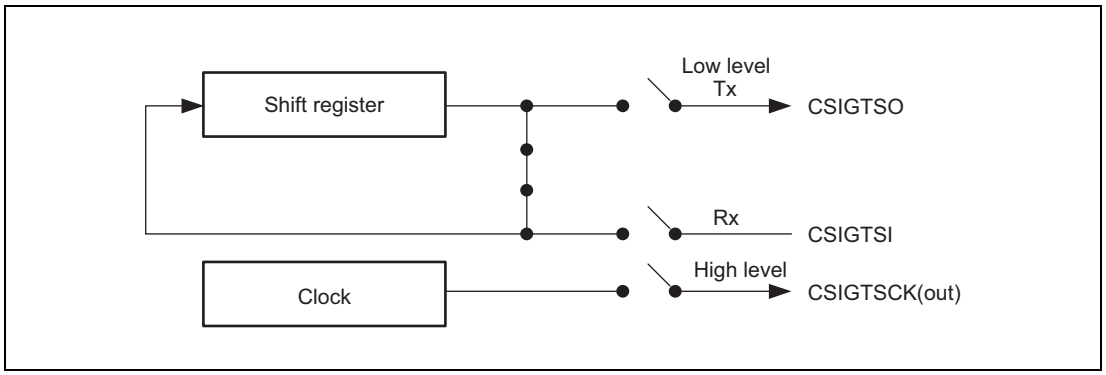


Figure 15.21 Operation in Loop-Back Mode

15.5.10 Error Detection

CSIG can detect three error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error (received data)

Data consistency error and parity error check functions can be individually enabled or disabled.

If one of these errors is detected, the interrupt INTCSIGTIRE is generated.

15.5.10.1 Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent as an output signal is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by the CSIGnCTL1.CSIGnDCS bit (when checking data consistency, make sure that PIPCn.PIPCn_m = 1 for CSIGTSO). It will not be enabled if data transmission is disabled (CSIGnCTL0.CSIGnTXE = 0).

When the data consistency check is enabled, the data transferred from CSIGnTX0W or CSIGnTX0H to the shift register is copied to a separate register. In addition, the physical levels of CSIGTSO are captured and the logical interpretation is written to the corresponding shift register.

After completion of the transmission, the data sent is compared with the original transmission data.

Mismatch is considered as a data consistency error and:

- Interrupt INTCSIGTIRE is generated.
- The CSIGnSTR0.CSIGnDCE bit is set.

The data consistency check function is illustrated in the following block diagram.

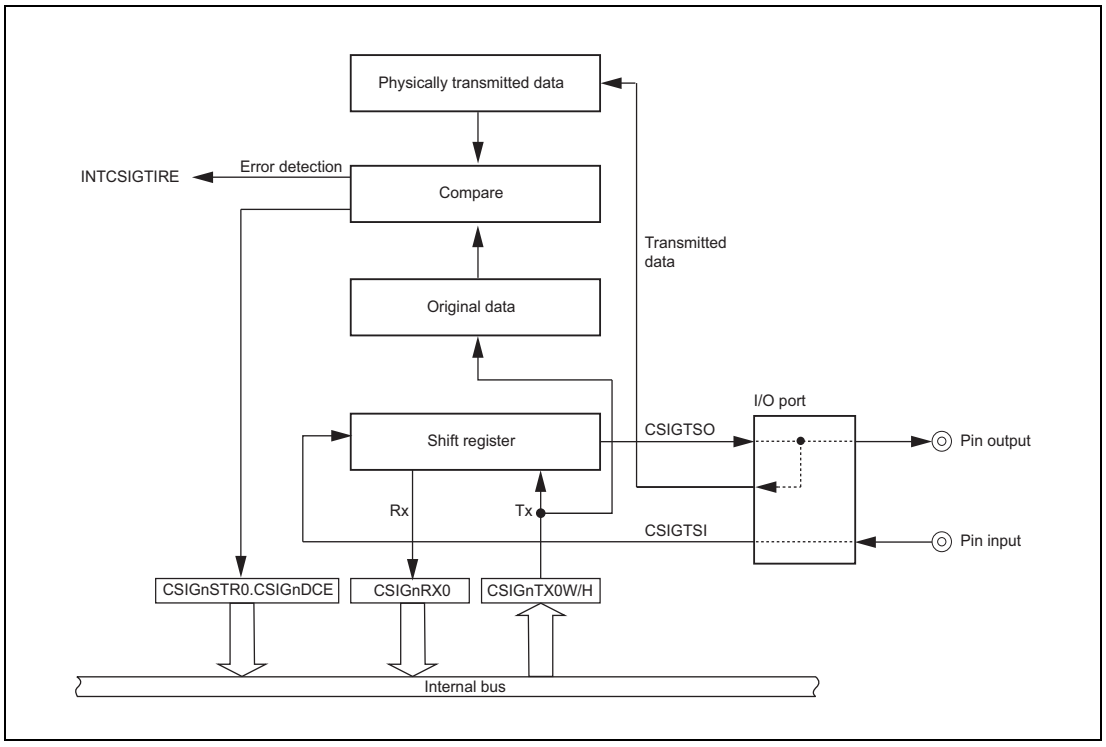


Figure 15.22 Functional Block Diagram of the Data Consistency Check

15.5.10.2 Parity Check

Parity is a common mean to detect a single bit error during data transmission. CSIG can append a parity bit after the last data bit (even if extended data length is used).

The use and type of parity is specified by CSIGnCFG0.CSIGnPS[1:0].

Parity check is enabled if CSIGnCFG0.CSIGnPS[1] = 1.

The parity bit is checked after reception is complete. If a parity error occurs:

- Interrupt INTCSIGTIRE is generated.
- The CSIGnSTR0.CSIGnPE bit is set.

The following figure shows an example.

Data length is 8 bits. The data transmitted is 05_H and 35_H. Parity type is odd.

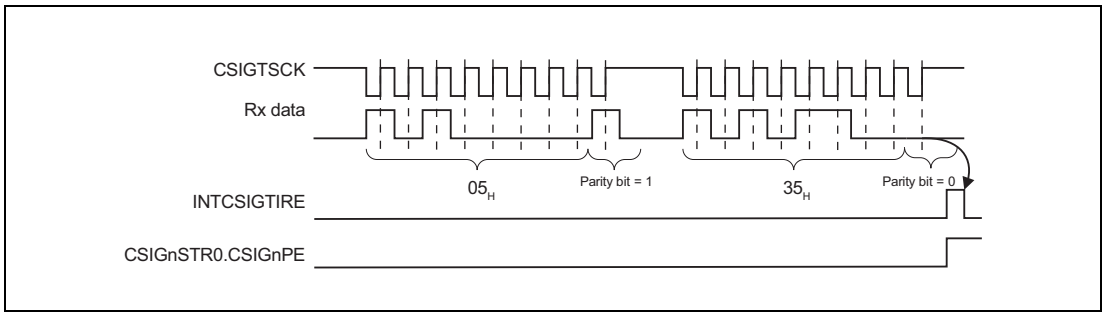


Figure 15.23 Parity Check Example

For the first 8 bits, the parity bit is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

For the second 8 bits, the parity bit is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If the EDL (extended data length) function is used, the parity bit is added after the last data bit.

15.5.10.3 Overrun Error

This error occurs when previously received data still resides in the reception register CSIGnRX0 because it has not been read, and new data is received.

The overrun error is not generated if data reception is disabled (CSIGnCTL0.CSIGnRXE = 0).

If overrun occurs:

- Interrupt INTCSIGTIRE is generated
- The CSIGnSTR0.CSIGnOVE bit is set
- Data in the CSIGnRX0 register is overwritten and communication continues.

The following figure illustrates the overrun error detection function.

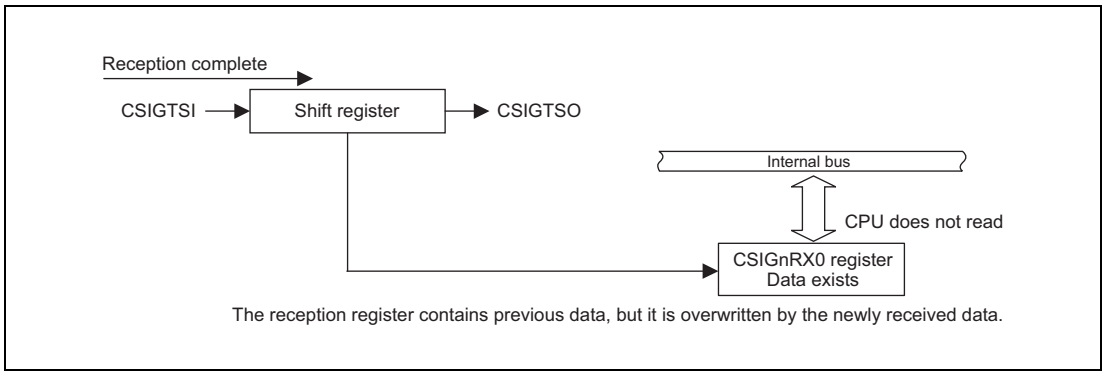


Figure 15.24 Overrun Error Detection

The following figure illustrates an example where:

- Rx data 3 is not read
- Rx data 4 is received, and data is overwritten.

Thus an overrun error occurs.

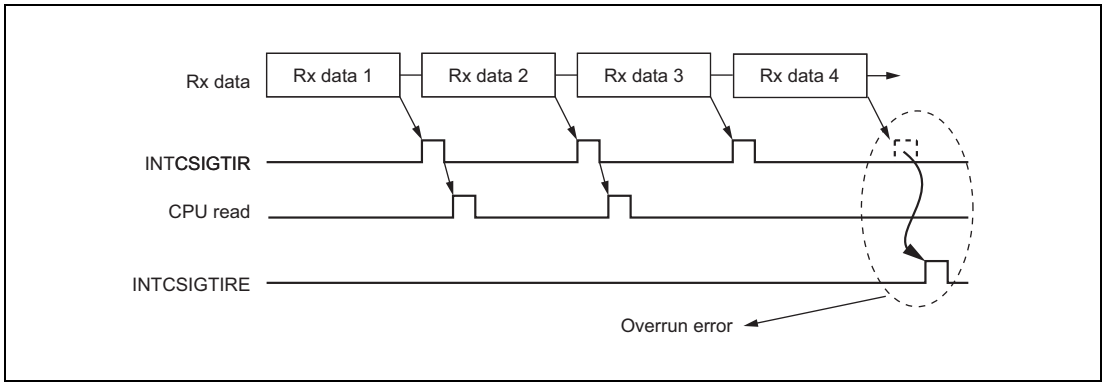


Figure 15.25 Overrun Error Detection - Example

NOTE

An overrun error can be avoided by using the handshake.

When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

For details see **Section 15.5.8, Handshake Function**.

15.6 Operating Procedures

15.6.1 Master Mode Transmission/Reception by DMA

This section describes an example of performing the transmission/reception in master mode in combination with a DMA.

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B)
- MSB is transmitted first (CSIGnCFG0.CSIGnDIR = 0)
- INTCSIGTIC interrupt is generated at the end of the transfer (CSIGnCTL1.CSIGnSLIT = 0)
- Normal clock and data phase (CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0)
- The number of data is 10 (0 to 9)

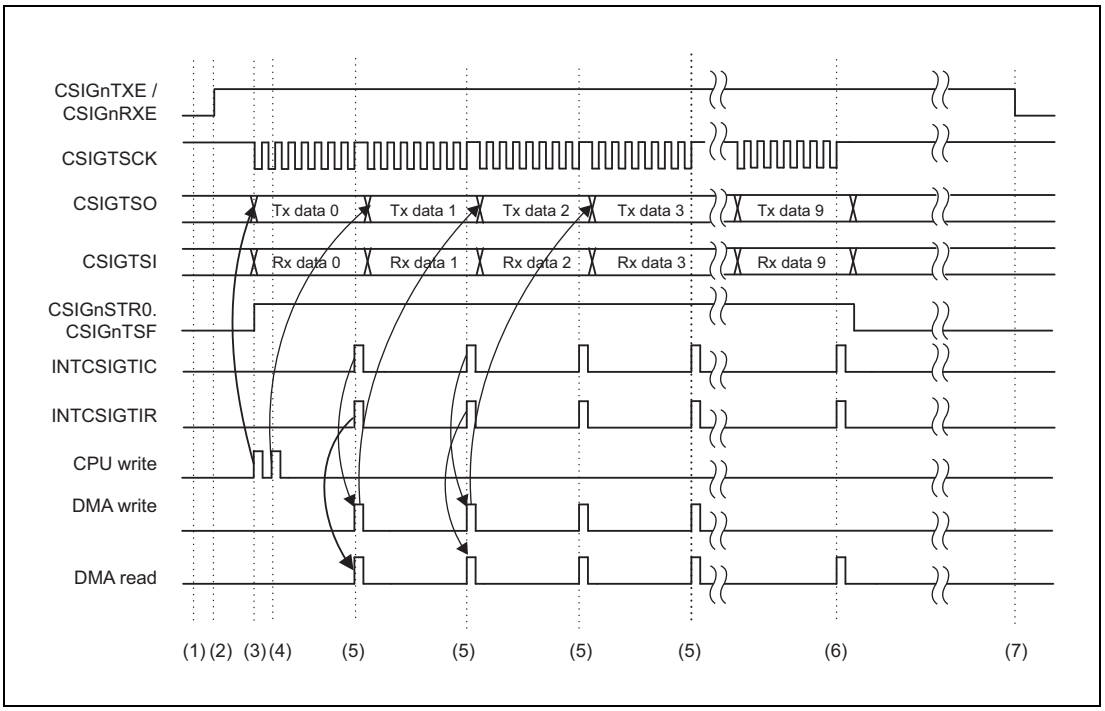


Figure 15.26 Communication in Master Mode

Procedure:

1. Configure the communication protocol in register CSIGnCFG0. Specify the interrupt timing and operating mode by setting the corresponding bits of the CSIGnCTL1 register and CSIGnCTL2 register.
2. In the CSIGnCTL0 register, set CSIGnPWR = 1 (enable clock), CSIGnTXE = 1 (enable transmission), and CSIGnRXE = 1 (enable reception).
3. Write the first data to be sent to the transmission register CSIGnTX0H. Transmission starts automatically when the first data becomes available.
4. Write the second data to CSIGnTX0H. Writing the second data immediately after the first one avoids unnecessary delays between the data.

5. After the transmission or reception of each data, INTCSIGTIC or INTCSIGTIR interrupt is generated. INTCSIGTIC indicates that the next data can be written to CSIGnTX0H. INTCSIGTIR indicates that the reception register CSIGnRX0 must be read.
   In this example, CPU write and DMA write are assumed as equivalent.
6. No more write action is required after the transmission of data 8 completes. Data 9 (the last data) has been written after the transmission of data 7. However, the reception register CSIGnRX0 must be read after the reception of data 8 and 9 completes.
7. Finally, to disable the transmit/receive operation, clear CSIGnCTL0.CSIGnTXE and CSIGnCTL0.CSIGnRXE. When no communication is taking place, set CSIGnCTL0.CSIGnPWR to “0” to minimize the power consumption of the CSIGn.

Section 16 Clocked Serial Interface H (CSIH)

This section contains a generic description of the Clocked Serial Interface H (CSIH).

The first part in this section describes the features specific to RH850/F1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of CSIH.

16.1 Features of RH850/F1M CSIH

16.1.1 Number of Units

This microcontroller has the following number of CSIH units.

Each CSIH unit has one channel interface.

Table 16.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	4		
Name	CSIHn (n = 0 to 3)		

Table 16.2 Indices

Index	Description
n	Throughout this section, the individual CSIH units are identified by the index “n” (n = 0 to 3): for example, CSIHnCTL0 is the CSIHn control register 0.
x	CSIHn has a maximum of 8 chip select signals. Throughout this section, the individual chip select signals are identified by the index “x”: that is, CSx denotes a non-specified chip select signal.
y	A variable used for explanation is identified by the index “y”: for example, CSIHnBRSy is the baud rate setting register of CSIHn.

The following table shows values indicated by the indices of each product.

Table 16.3 Indices of Products

Indices of Each Product		
144 pins	**176 pins**	**233 pins**
For the value of x, see **Table 16.4, Number of Chip Select Signals**.		
y = 0 to 3	y = 0 to 3	y = 0 to 3

The numbers of chip select signals for each of the CSIH units are listed in the following table.

Table 16.4 Number of Chip Select Signals

Unit Name	Chip Select Index		
	144 pins	176 pins	233 pins
CSIH0	CSx (x = 0 to 7)	CSx (x = 0 to 7)	CSx (x = 0 to 7)
CSIH1	CSx (x = 0 to 5)	CSx (x = 0 to 5)	CSx (x = 0 to 5)
CSIH2	CSx (x = 0 to 5)	CSx (x = 0 to 5)	CSx (x = 0 to 5)
CSIH3	CSx (x = 0 to 3)	CSx (x = 0 to 3)	CSx (x = 0 to 3)

16.1.2 Register Base Address

CSIH base addresses are listed in the following table.

CSIH register addresses are given as offsets from the base addresses.

Table 16.5 Register Base Addresses

Base Address Name	Base Address
<CSIH0_base>	FFD8 0000_H
<CSIH1_base>	FFD8 2000_H
<CSIH2_base>	FFD8 4000_H
<CSIH3_base>	FFD8 6000_H

16.1.3 Clock Supply

The CSIH clock supply is shown in the following table.

Table 16.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
CSIHn	PCLK	CKSCLK_ICSI
	Register access	CKSCLK_ICSI

16.1.4 Interrupt Requests

CSIH interrupt requests are listed in the following table.

Table 16.7 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
CSIH0			
INTCSIHTIC	Communication status interrupt	29	70
INTCSIHTIR	Receive status interrupt	30	71
INTCSIHTIRE	Communication error interrupt	31	—
INTCSIHTIJC	Job completion interrupt	20, 32	72
CSIH1			
INTCSIHTIC	Communication status interrupt	16, 116	28
INTCSIHTIR	Receive status interrupt	17, 117	29
INTCSIHTIRE	Communication error interrupt	27, 118	—
INTCSIHTIJC	Job completion interrupt	28, 119	30
CSIH2			
INTCSIHTIC	Communication status interrupt	8, 132	89
INTCSIHTIR	Receive status interrupt	10, 133	90
INTCSIHTIRE	Communication error interrupt	11, 134	—
INTCSIHTIJC	Job completion interrupt	12, 135	91
CSIH3			
INTCSIHTIC	Communication status interrupt	9, 158	41
INTCSIHTIR	Receive status interrupt	13, 159	42
INTCSIHTIRE	Communication error interrupt	14, 160	—
INTCSIHTIJC	Job completion interrupt	15, 161	43

16.1.5 Reset Sources

CSIH reset sources are listed in the following table. CSIH is initialized by these reset sources.

Table 16.8 Reset Sources

Unit Name	Reset Source
CSIHn	All reset sources (ISORES)

16.1.6 External Input/Output Signals

External input/output signals of CSIH are listed below.

Table 16.9 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
CSIH0		
CSIHTSCK	Serial clock signal	CSIH0SC*2
CSIHTSI	Serial data input signal	CSIH0SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH0SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH0RYI
CSIHTSO	Serial data output signal	CSIH0SO*2
CSIHTRYO	Ready/busy output signal	CSIH0RYO
CSIHTCSS[7:0]*1	Chip select signal	CSIH0CSS[7:0]*1
CSIH1		
CSIHTSCK	Serial clock signal	CSIH1SC*2
CSIHTSI	Serial data input signal	CSIH1SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH1SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH1RYI
CSIHTSO	Serial data output signal	CSIH1SO*2
CSIHTRYO	Ready/busy output signal	CSIH1RYO
CSIHTCSS[5:0]*1	Chip select signal	CSIH1CSS[5:0]*1
CSIH2		
CSIHTSCK	Serial clock signal	CSIH2SC*2
CSIHTSI	Serial data input signal	CSIH2SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH2SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH2RYI
CSIHTSO	Serial data output signal	CSIH2SO*2
CSIHTRYO	Ready/busy output signal	CSIH2RYO
CSIHTCSS[5:0]*1	Chip select signal	CSIH2CSS[5:0]*1
CSIH3		
CSIHTSCK	Serial clock signal	CSIH3SC*2
CSIHTSI	Serial data input signal	CSIH3SI
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH3SSI}}$
CSIHTRYI	Ready/busy input signal	CSIH3RYI
CSIHTSO	Serial data output signal	CSIH3SO*2
CSIHTRYO	Ready/busy output signal	CSIH3RYO
CSIHTCSS[3:0]*1	Chip select signal	CSIH3CSS[3:0]*1

Note 1. For the number of chip select signals, see **Table 16.4, Number of Chip Select Signals**.

Note 2. For the port pins that are used as CSIHnSO and CSIHnSC, set the output driving ability to high (PDSCn_m = 1).

CAUTION

When the P0_0 pin is used as CSIH0SSI, the P0_0 pin outputs a low-level $\overline{\text{RESETOUT}}$ signal while a reset is asserted and continues to output a low level after the reset is deasserted.

For details, see **Section 2.11.1.1, P0_0: $\overline{\text{RESETOUT}}$**.

16.1.7 Data Consistency Check

The port and the alternative function for data consistency check of CSIHnSO (CSIHTSO) output are shown in the following table. See **Section 16.5.12, Error Detection** for details on data consistency checking.

Table 16.10 Port Pins for Data Consistency Checking

Unit Signal Name	Port Pin Name	Alternative Function
CSIH0		
CSIHTSO	P0_3	ALT_OUT4
CSIH1		
CSIHTSO	P0_5	ALT_OUT3
	P10_2	ALT_OUT5
CSIH2		
CSIHTSO	P11_2	ALT_OUT1
CSIH3		
CSIHTSO	P11_6	ALT_OUT3

16.2 Overview

16.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Master mode or slave mode is selectable
- Multiple slaves configuration and RCB (Recessive Configuration for Broadcasting) are possible since there are eight configurable chip select output signals
- Slave select input signal ($\overline{\text{CSIHTSSI}}$) is usable
- Built-in baud rate generator
- Transfer clock frequency is adjustable in master mode, whereas it is determined by the input clock in slave mode.
- Maximum transfer clock frequency:
  - Master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - Slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)
- Clock phase and data phase are selectable.
- Data transfer with MSB first or LSB first is selectable
- Transfer data length is selectable from 2 to 16 bits in 1-bit units
- Built-in EDL (extended data length) function for transferring more than 16 bits of data
- Three selectable transfer modes:
  - transmit-only mode
  - receive-only mode
  - transmit/receive mode
- Built-in handshake function
- Built-in error detection (data consistency check, parity, time-out, overflow, and overrun)
- Support of job concept
- 128 words I/O buffer memory
- Direct access mode or memory mode (FIFO, dual buffer, and transmit-only buffer) is selectable
- Four different interrupt request signals (INTCSIHTIC, INTCSIHTIR, INTCSIHTIRE, INTCSIHTIJC)
- Built-in LBM (loop back mode) function for self-test
- CPU-controlled high-priority communication function
- Enforced chip select idle setting
- Built-in RCB (recessive configuration for broadcasting) bit
- Built-in JOB enable control bit for AUTOSAR

16.2.2 Functional Overview Description

The CSIH uses three signals for communication:

- Transmission clock CSIHTSCK (output in master mode, input in slave mode)
- Data output signal CSIHTSO
- Data input signal CSIHTSI

Additional signals are available for external control and monitoring:

- $\overline{\text{CSIHTSSI}}$: Slave select input signal
- CSIHTRYO: Ready/busy output signal (handshake signal)
- CSIHTRYI: Ready/busy input signal (handshake signal)
- CSIHTCSS[7:0]: Chip select signals

Data transmission is bit-wise and serial, and performed synchronously with the transmission clock.

The following table shows the most important registers for setting up the CSIH.

Table 16.11 Main Registers of CSIH

Register	Function
CSIHnCTL0	Enables/disables serial clock, and permits/ prohibits data transmission and data reception. Defines end-of-job behavior and enables/disables (bypasses) buffering.
CSIHnCTL1	Controls options such as interrupt timing, extended data length, job feature, data consistency check, loop-back mode, handshake, etc.
CSIHnCTL2	Selects master or slave mode, and the transfer clock frequency of the built-in baud rate generator (BRG) in master mode.
CSIHnBRSy	Specifies the transfer clock frequency for each chip select signal.
CSIHnMCTL0	Selects memory mode and specifies the time-out value.
CSIHnMCTL1	Controls the memory in FIFO mode.
CSIHnMCTL2	Controls the memory in dual buffer mode.
CSIHnCFGx	Configures the communication protocol for each chip select signal.

16.2.3 Block Diagram

The following block diagram shows the main components of the CSIH.

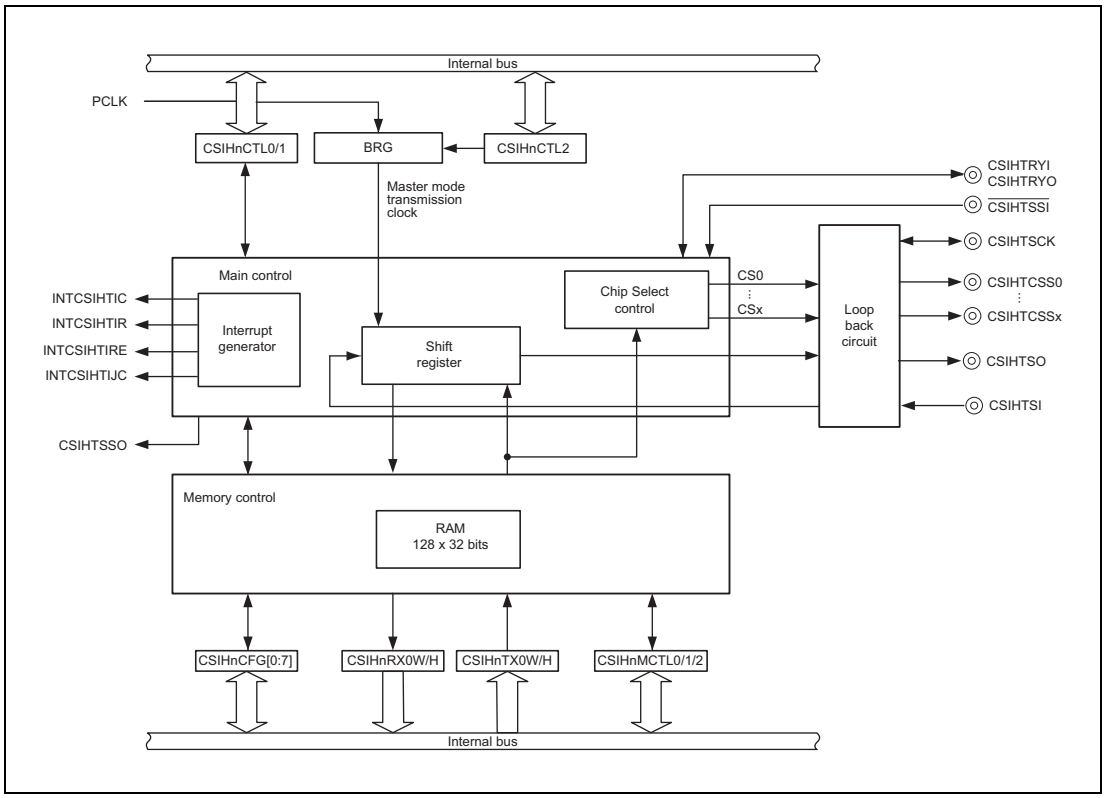


Figure 16.1 CSIH Block Diagram

In master mode, the transmission clock CSIHTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is supplied by an external source.

The built-in memory can be configured as FIFO, dual buffer (separate transmit and receive buffers), or transmit-only buffer. It can also be bypassed for data transmission and reception without buffering.

The loop back circuit disconnects the CSIH completely from the ports and supports internal self-test.

NOTE

This section describes the following modes:

- The “operating mode” is either master mode or and slave mode. In this context, only a master can control and communicate with several slaves (for details, see **Section 16.5.1, Operating Modes (Master/Slave)**).
- The “job mode” is related to the AUTOSAR job concept (for details, see **Section 16.5.3.3, Job Concept**).
- The “memory mode” accords with the various configurations of the associated buffer memory (for details, see **Section 16.5.6, CSIH Buffer Memory**).
- The “data transfer mode” specifies the type of communication – transmit-only, receive-only, or transmit/receive (for details, see **Section 16.5.7, Data Transfer Modes**).

16.3 Registers

16.3.1 List of Registers

CSIH registers are listed in the following table.

For details about <CSIHn_base>, see **Section 16.1.2, Register Base Address**.

Table 16.12 List of Registers

Module Name	Register Name	Symbol	Address
CSIHn	CSIHn control register 0	CSIHnCTL0	<CSIHn_base> + 0000_H
CSIHn	CSIHn control register 1	CSIHnCTL1	<CSIHn_base> + 0010_H
CSIHn	CSIHn control register 2	CSIHnCTL2	<CSIHn_base> + 0014_H
CSIHn	CSIHn status register 0	CSIHnSTR0	<CSIHn_base> + 0004_H
CSIHn	CSIHn status clear register 0	CSIHnSTCR0	<CSIHn_base> + 0008_H
CSIHn	CSIHn memory control register 0	CSIHnMCTL0	<CSIHn_base> + 1040_H
CSIHn	CSIHn memory control register 1	CSIHnMCTL1	<CSIHn_base> + 1000_H
CSIHn	CSIHn memory control register 2	CSIHnMCTL2	<CSIHn_base> + 1004_H
CSIHn	CSIHn memory read/write pointer register 0	CSIHnMRWP0	<CSIHn_base> + 1018_H
CSIHn	CSIHn configuration register 0	CSIHnCFG0	<CSIHn_base> + 1044_H
CSIHn	CSIHn configuration register 1	CSIHnCFG1	<CSIHn_base> + 1048_H
CSIHn	CSIHn configuration register 2	CSIHnCFG2	<CSIHn_base> + $104C_H$
CSIHn	CSIHn configuration register 3	CSIHnCFG3	<CSIHn_base> + 1050_H
CSIHn	CSIHn configuration register 4	CSIHnCFG4	<CSIHn_base> + 1054_H
CSIHn	CSIHn configuration register 5	CSIHnCFG5	<CSIHn_base> + 1058_H
CSIHn	CSIHn configuration register 6	CSIHnCFG6	<CSIHn_base> + $105C_H$
CSIHn	CSIHn configuration register 7	CSIHnCFG7	<CSIHn_base> + 1060_H
CSIHn	CSIHn transmit data register 0 for word access	CSIHnTX0W	<CSIHn_base> + 1008_H
CSIHn	CSIHn transmit data register 0 for half word access	CSIHnTX0H	<CSIHn_base> + $100C_H$
CSIHn	CSIHn receive data register 0 for word access	CSIHnRX0W	<CSIHn_base> + 1010_H
CSIHn	CSIHn receive data register 0 for half word access	CSIHnRX0H	<CSIHn_base> + 1014_H
CSIHn	CSIHn emulation register	CSIHnEMU	<CSIHn_base> + 0018_H
CSIHn	CSIHn baud rate setting register 0	CSIHnBRS0	<CSIHn_base> + 1068_H
CSIHn	CSIHn baud rate setting register 1	CSIHnBRS1	<CSIHn_base> + $106C_H$
CSIHn	CSIHn baud rate setting register 2	CSIHnBRS2	<CSIHn_base> + 1070_H
CSIHn	CSIHn baud rate setting register 3	CSIHnBRS3	<CSIHn_base> + 1074_H

16.3.2 CSIHnCTL0 — CSIHn Control Register 0

This register controls the operation clock, enables/disables transmission/reception, and enables/disables the memory allocated for transmission and/or reception. It forces the stop of communication at the end of the current job.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <CSIHn_base> + 0000_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CSIHnPWR	CSIHnTXE	CSIHnRXE	—	—	—	CSIHnJOBE	CSIHnMBS
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R/W	R/W

Table 16.13 CSIHnCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIHnPWR	Controls the operation clock. 0: Stops operation clock. 1: Supplies operation clock. Clearing CSIHnPWR to 0 resets the internal circuits, stops operation, and sets CSIH to standby state. Clock supply to internal circuits stops. If CSIHnPWR is cleared to 0 during communication, ongoing communication is immediately aborted. In this case, the communication setting must be reconfigured.
6	CSIHnTXE	Permits or prohibits transmission. 0: Prohibits transmission. 1: Permits transmission.
5	CSIHnRXE	Permits or prohibits reception. 0: Prohibits reception. 1: Permits reception.
4 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CSIHnJOBE	Stops communication at the end of the current job (communication ends if data is written to the transmission buffer when CSIHnTX0W.CSIHnEOJ = 1 (job completion)). 0: Communication stop is not requested. 1: Stops communication. This bit can be used to abort an ongoing job. This bit is cleared to 0 automatically. Even if this bit is set to 1, the read value is always 0. In FIFO mode, the pointer must be cleared by setting CSIHnSTCR0.CSIHnPCT = 1 before the next communication is started
0	CSIHnMBS	Bypasses the memory for transmission and/or reception data. 0: Memory mode CSIH memory is used for transmission and/or reception data. 1: Direct access mode CSIH memory is bypassed.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.3 CSIHnCTL1 — CSIHn Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It also enables/disables extended data length control, data consistency check, loop-back mode, handshake functionality, and job mode. It selects the active output level of each chip select signal and the behavior of the chip select signals after the transfer of the final data.

Access: This register can be read or written in 32-bit units.

Address: <CSIHn_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	CSIHn SLRS	—	—	—	—	—	CSIHn PHE	CSIHn CKR	CSIHn SLIT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn CSL7	CSIHn CSL6	CSIHn CSL5	CSIHn CSL4	CSIHn CSL3	CSIHn CSL2	CSIHn CSL1	CSIHn CSL0	CSIHn EDLE	CSIHn JE	CSIHn DCS	CSIHn CSRI	CSIHn LBM	CSIHn SIT	CSIHn HSE	CSIHn SSE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.14 CSIHnCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	CSIHnSLRS	Sets the internal synchronization timing for receive data input. 0: Rising edge of PCLK 1: Falling edge of PCLK For differences by the setting, see Data Sheet.
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	CSIHnPHE	Sets the CPU-controlled priority-based communication function. 0: The CPU-controlled high-priority communication function is disabled. 1: The CPU-controlled high-priority communication function is enabled. To enable the CPU-controlled high-priority communication function, set this bit to 1 and set CSIHnJE = 1. This bit can only be set in transmit-only buffer mode.
17	CSIHnCKR	CSIHTSCK Clock Inversion Function 0: The default level of CSIHTSCK is high 1: The default level of CSIHTSCK is low For details, see **Section 16.3.11, CSIHnCFGx — CSIHn Configuration Register x**.
16	CSIHnSLIT	Selects the timing of interrupt INTCSIHTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: As soon as the contents of the CSIHnTX0W/H register are transferred to the shift register, an interrupt is generated (this function is activated only in direct access mode/transmit-only buffer mode). For details, see **Section 16.4.3, INTCSIHTIC (Communication Status Interrupt)**.
15 to 8	CSIHnCSLx	Selects the active output level of chip select signal x (CSIHTCSSx) (x = 0 to 7). 0: Chip select is active low. 1: Chip select is active high. For details, see **Section 16.5.3, Chip Selection (CS) Features**.
7	CSIHnEDLE	Enables/disables extended data length (EDL) mode. 0: Disables extended data length mode. 1: Enables extended data length mode. For details, see **Section 16.5.8.2, Data Length Greater than 16 Bits**.

Table 16.14 CSIHnCTL1 Register Contents (2/2)

Bit Position	Bit Name	Function
6	CSIHnJE	Enables/disables job mode. 0: Disables job mode. 1: Enables job mode. For details, see **Section 16.5.3.3, Job Concept**. The CSIHnCTL0.CSIHnJOBE, CSIHnTX0W.CSIHnEOJ, and CSIHnTX0W.CSIHnCIRE bits are enabled only when CSIHnJE = 1. Setting this bit in slave mode is prohibited. In addition, to enable the CPU-controlled high-priority communication function, set this bit to 1 as well as CSIHnPHE = 1.
5	CSIHnDCS	Enables/disables data consistency check. 0: Disables data consistency check. 1: Enables data consistency check. For details, see **Section 16.5.12.1, Data Consistency Check**.
4	CSIHnCSRI	Defines chip select signal behavior after last data transfer. 0: Chip select signal retains the active level. 1: Chip select signal returns to the inactive level. The last data is determined at the interrupt timing in direct access mode or FIFO mode. When CSIHnCTL1.CSIHnSLIT = 1, the last data is determined in direct access mode.
3	CSIHnLBM	Controls loop-back mode (LBM). 0: Deactivates loop-back mode. 1: Activates loop-back mode. For details, see **Section 16.5.13, Loop-Back Mode**.
2	CSIHnSIT	Selects interrupt delay mode. 0: No delay is generated. 1: Half clock delay is generated for all interrupts. This bit is only valid in master mode. In slave mode, no delay is generated. For details, see **Section 16.4.2, Interrupt Delay**.
1	CSIHnHSE	Enables/disables the handshake function. 0: Disables the handshake function. 1: Enables the handshake function. For details, see **Section 16.5.11, Handshake Function**.
0	CSIHnSSE	Enables/disables the slave select function. 0: Input signal $\overline{\text{CSIHTSSI}}$ is disabled. 1: Input signal $\overline{\text{CSIHTSSI}}$ is recognized. If the slave select function is not used, this bit must be set to 0 (see also **Section 16.5.2, Master/Slave Connections**).

Details about CSIHnCTL1.CSIHnSSE are shown in the following tables.

Table 16.15 Operation of the Slave Select Function during Reception

CSIHnCTL0. CSIHnRXE	CSIHnCTL1. CSIHnSSE	$\overline{\text{CSIHTSSI}}$	Receive Operation
0	—	—	Reception is prohibited
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

Table 16.16 Operation of the Slave Select Function during Transmission

CSIHnCTL0. CSIHnTXE	CSIHnCTL1. CSIHnSSE	$\overline{\text{CSIHTSSI}}$	Transmit Operation
0	—	—	Transmission is prohibited
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.4 CSIHnCTL2 — CSIHn Control Register 2

This register selects operating mode and the reference clock value, and specifies the transfer clock frequency.

For details, see **Section 16.5.5, Transmission Clock Selection**.

Access: This register can be read or written in 16-bit units.

Address: <CSIHn_base> + 0014_H

Value after reset: $E000_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnPRS[2:0]			—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 16.17 CSIHnCTL2 Register Contents

Bit Position	Bit Name	Function
15 to 13	CSIHnPRS[2:0]	These bits select the operation mode and the reference clock value.
12 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CSIHnPRS2	CSIHnPRS1	CSIHnPRS0	Selection of Reference Clock (PRSOUT)
0	0	0	PCLK (Master mode)
0	0	1	PCLK/2 (Master mode)
0	1	0	PCLK/4 (Master mode)
0	1	1	PCLK/8 (Master mode)
1	0	0	PCLK/16 (Master mode)
1	0	1	PCLK/32 (Master mode)
1	1	0	PCLK/64 (Master mode)
1	1	1	External clock via CSIHTSCK(in) (Slave mode)

In master mode, the following bits are used to set the transfer clock frequency:

CSIHnCTL2.CSIHnPRS[2:0], CSIHnCFGx.CSIHnBRSS[1:0], CSIHnBRSy.CSIHnBRS[11:0]

In addition, any of the four different transfer clock frequency settings that are specified by the CSIHnBRSy.CSIHnBRS[11:0] bits is selected for each chip select signal. To select the transfer clock frequency setting for each chip select signal, use the CSIHnCFGx.CSIHnBRSS[1:0] bits.

The following table shows the relationship between CSIHnCFGx.CSIHnBRSS[1:0] and CSIHnBRSy.CSIHnBRS[11:0].

CSIHnCFGx. CSIHnBRSS[1:0]	Transfer Clock Frequency Setting Bit to be Selected
00	CSIHnBRS0.CSIHnBRS[11:0]
01	CSIHnBRS1.CSIHnBRS[11:0]
10	CSIHnBRS2.CSIHnBRS[11:0]
11	CSIHnBRS3.CSIHnBRS[11:0]

The following table shows the relationship between the transfer clock frequency and the transfer clock frequency setting (CSIHnBRSy[11:0]) selected by the CSIHnBRSS[1:0] bits when the bit value of the CSIHnPRS[2:0] bits is α.

CSIHnBRSy[11:0]	Transfer clock frequency
0	BRG stopped
1	PCLK / (2^{α} × 1 × 2)
2	PCLK / (2^{α} × 2 × 2)
3	PCLK / (2^{α} × 3 × 2)
4	PCLK / (2^{α} × 4 × 2)
...	…
4095	PCLK / (2^{α} × 4095× 2)

When a time-out error is used in slave mode, the clock selected by this setting is used. In slave mode, the CSIHnPRS[2:0] bits are set to 111_B. In this case, the prescaler has the same setting as when the CSIHnPRS[2:0] bits are set to 000_B. When using a time-out error, set the CSIHnBRSy.CSIHnBRS[11:0] bits to a value other than 000_H.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.5 CSIHnSTR0 — CSIHn Status Register 0

This register indicates the status of CSIH.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <CSIHn_base> + 0004_H

Value after reset: $0000\ 0010_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnSRP[7:0]								CSIHnSPF[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn TMOE	CSIHn OFE	—	—	—	—	—	CSIHn HPST	CSIHn TSF	—	CSIHn FLF	CSIHn EMF	CSIHn DCE	—	CSIHn PE	CSIHn OVE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 16.18 CSIHnSTR0 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 24	CSIHnSRP[7:0]	Indicates the number of received data in FIFO mode. **CSIHnSRP[7:0]** / **Description** 00_H — Number of received data (0 to 128) … 80_H Other than the above — Undefined These bits are cleared by CSIHnSTCR0.CSIHnPCT. In direct access mode, dual buffer mode, or transmit-only buffer mode, this value is fixed to 00_H. In direct access mode, this bit is fixed to 0 because there is no pointer. In buffer mode, this bit is fixed to 0 because the number of data is managed by CSIHnMCTL2.CSIHnND[7:0].
23 to 16	CSIHnSPF[7:0]	Indicates the number of untransmitted data in FIFO mode. (The number of data written by the CPU is the number of sent data.) **CSIHnSPF[7:0]** / **Description** 00_H — Number of untransmitted data (0 to 128) … 80_H Other than the above — Undefined These bits are cleared by CSIHnSTCR0.CSIHnPCT. In direct access mode, dual buffer mode, or transmit-only buffer mode, this value is fixed to 00_H. In direct access mode, this bit is fixed to 0 because there is no pointer. In buffer mode, this bit is fixed to 0 because the number of data is managed by CSIHnMCTL2.CSIHnND[7:0].

Table 16.18 CSIHnSTR0 Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15</td><td>CSIHnTMOE</td><td>Time-out Error Flag in FIFO Mode
Indicates whether a time-out error was detected in FIFO mode.
0: No time out error was detected.
1: A time out error was detected.
For details, see Section 16.5.12.3, Time-Out Error.
This bit is cleared by CSIHnSTCR0.CSIHnTMOEC.
When setting to 1 by time-out error detection and clearing to 0 by CSIHnSTCR0.CSIHnTMOEC occur simultaneously, setting to 1 takes precedence over clearing to 0.
This bit is also initialized when CSIHnCTL0.CSIHnPWR is changed from 0 to 1 or from 1 to 0.</td></tr>
<tr><td>14</td><td>CSIHnOFE</td><td>Overflow Error Flag in FIFO mode
Indicates whether an overflow error was detected in FIFO mode.
0: No overflow error was detected.
1: An overflow error was detected.
For details, see Section 16.5.12.4, Overflow Error.
This bit is cleared by CSIHnSTCR0.CSIHnOFEC.
When setting to 1 by overflow error detection and clearing to 0 by CSIHnSTCR0.CSIHnOFEC occur simultaneously, setting to 1 takes precedence over clearing to 0.
This bit is also initialized when CSIHnCTL0.CSIHnPWR is changed from 0 to 1 or from 1 to 0.</td></tr>
<tr><td>13 to 9</td><td>Reserved</td><td>When read, the value after reset is returned.</td></tr>
<tr><td>8</td><td>CSIHnHPST</td><td>Communication Priority Indication Flag
0: Indicates low-priority communication is in progress.
1: Indicates high-priority communication is in progress.
This bit always reads 0 if CPU-controlled high-priority communication is disabled (CSIHnCTL1.CSIHnPHE = 0).</td></tr>
<tr><td>7</td><td>CSIHnTSF</td><td>Transfer Status Flag
0: Idle state
1: Communication is in progress or being prepared.
The timing to set or clear this bit is as follows:
<table>
<tr><th rowspan="2">Master Mode</th><th colspan="2">Timing to Set</th><th rowspan="2">Timing to Clear</th></tr>
<tr><th>Direct Access Mode,
FIFO Mode</th><th>Dual Buffer Mode,
Transmit-Only Buffer Mode</th></tr>
<tr><td>Transmit-only mode</td><td rowspan="3">Data is written to a transmit register (CSIHnTX0W/ CSIHnTX0H)</td><td rowspan="3">CSIHnMCTL2.CSIHnBTST Bit is set</td><td rowspan="3">Within a half clock of the last serial clock edge</td></tr>
<tr><td>Transmit/receive mode</td></tr>
<tr><td>Receive-only mode</td></tr>
</table>
<table>
<tr><th rowspan="2">Slave Mode</th><th colspan="2">Timing to Set</th><th rowspan="2">Timing to Clear</th></tr>
<tr><th>Direct Access Mode,
FIFO Mode</th><th>Dual Buffer Mode,
Transmit-Only Buffer Mode</th></tr>
<tr><td>Transmit-only mode</td><td rowspan="2">Data is written to a transmit register (CSIHnTX0W/ CSIHnTX0H)</td><td rowspan="3">CSIHnMCTL2.CSIHnBTST Bit is set</td><td rowspan="3">Within a half clock cycle of the last serial clock edge</td></tr>
<tr><td>Transmit/receive mode</td></tr>
<tr><td>Receive-only mode</td><td>Input timing of CSIHTSCK</td></tr>
</table></td></tr>
<tr><td>6</td><td>Reserved</td><td>When read, the value after reset is returned.</td></tr>
<tr><td>5</td><td>CSIHnFLF</td><td>A flag indicating that the buffer is full in FIFO mode.
0: FIFO buffer is not full.
1: FIFO buffer is full.
This bit is cleared by CSIHnSTCR0.CSIHnPCT.
The FIFO buffer might be filled with untransmitted data or received data.</td></tr>
</table>

Table 16.18 CSIHnSTR0 Register Contents (3/3)

Bit Position	Bit Name	Function
4	CSIHnEMF	A flag indicating that the buffer is empty in FIFO mode. 0: FIFO buffer is not empty. 1: FIFO buffer is empty. This bit is set to 1 by CSIHnSTCR0.CSIHnPCT. This bit is set to 1 when CSIHnSTR0.CSIHnSRP[7:0] + CSIHnSTR0.CSIHnSPF[7:0] = 00_H. The FIFO buffer might be filled with untransmitted data or received data.
3	CSIHnDCE	Data Consistency Check Error Flag 0: No data consistency error is detected. 1: Data consistency error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnDCEC. When setting to 1 by data consistency error detection and clearing to 0 by CSIHnSTCR0.CSIHnDCEC occur simultaneously, setting to 1 takes precedence over clearing to 0. This bit is initialized when CSIHnCTL0.CSIHnPWR is changed from 0 to 1 or from 1 to 0.
2	Reserved	When read, the value after reset is returned.
1	CSIHnPE	Parity Error Flag 0: No parity error is detected. 1: Parity error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnPEC. When setting to 1 due to parity error detection and clearing to 0 by CSIHnSTCR0.CSIHnPEC occur simultaneously, setting to 1 by parity error detection takes precedence over clearing to 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.
0	CSIHnOVE	Overrun Error Flag (Fixed to 0 in dual buffer mode) 0: No overrun error is detected. 1: Overrun error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnOVEC. When setting to 1 due to overrun error detection and clearing to 0 by writing to CSIHnSTCR0.CSIHnOVEC occur simultaneously, setting to 1 by overrun error detection takes precedence over clearing to 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.

Table 16.19 Behaviors in Various Memory Modes

Bit Name	Bit Position	Direct Access Mode	FIFO Mode	Transmit-Only Buffer Mode	Dual Buffer Mode
CSIHnSRP[7:0]	31 to 24	Fixed to 0	Number of received data	Fixed to 0	Fixed to 0
CSIHnSPF[7:0]	23 to 16	Fixed to 0	Number of untransmitted data	Fixed to 0	Fixed to 0
CSIHnTMOE	15	Fixed to 0	0: No error is detected. 1: An error is detected.	Fixed to 0	Fixed to 0
CSIHnOFE	14	Fixed to 0	0: No error is detected. 1: An error is detected.	Fixed to 0	Fixed to 0
CSIHnTSF	7	0: Idle state 1: Communication is in progress or being prepared			
CSIHnFLF	5	Fixed to 0	0: FIFO is not full 1: FIFO is full	Fixed to 0	Fixed to 0
CSIHnEMF	4	Fixed to 1	0: FIFO is not empty 1: FIFO is empty	Fixed to 1	Fixed to 1
CSIHnDCE	3	0: No error is detected. 1: An error is detected.			
CSIHnPE	1	0: No error is detected. 1: An error is detected.			
CSIHnOVE	0	0: No error is detected. 1: An error is detected.	0: No error is detected. 1: An error is detected.	0: No error is detected. 1: An error is detected.	Fixed to 0

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.6 CSIHnSTCR0 — CSIHn Status Clear Register 0

This register clears the status flags of the CSIHnSTR0 status register.

Access: This register can be read or written in 16-bit units.
When read, the value 0000_H is always returned.

Address: <CSIHn_base> + 0008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn TMOEC	CSIHn OFEC	—	—	—	—	—	CSIHn PCT	—	—	—	—	CSIHn DCEC	—	CSIHn PEC	CSIHn OVEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R/W	R	R	R	R	R/W	R	R/W	R/W

Table 16.20 CSIHnSTCR0 Register Contents

Bit Position	Bit Name	Function
15	CSIHnTMOEC	Controls the time-out error flag clear command. 0: No operation. The read value is always 0. 1: Clears the time out error flag (CSIHnSTR0.CSIHnTMOE).
14	CSIHnOFEC	Controls the overflow error flag clear command. 0: No operation. The read value is always 0. 1: Clears the overflow error flag (CSIHnSTR0.CSIHnOFE).
13 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	CSIHnPCT	Controls the FIFO pointer clear command. 0: No operation. The read value is always 0. 1: Clears the following FIFO buffer pointers (in FIFO mode, dual buffer mode, and transmit-only buffer mode) and status bits. **FIFO Buffer Pointer / Status Bit** CSIHnMRWP0.CSIHnTRWA[6:0] / CSIHnSTR0.CSIHnSPF[7:0] CSIHnMRWP0.CSIHnRRA[6:0] / CSIHnSTR0.CSIHnSRP[7:0] CSIHnMCTL2.CSIHnSOP[6:0] / CSIHnSTR0.CSIHnFLF CSIHnSTR0.CSIHnTSF Additionally, the CSIHnSTR0.CSIHnEMF bit is set to 1 (FIFO empty) (in FIFO mode only).
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	CSIHnDCEC	Controls the data consistency error flag clear command. 0: No operation. The read value is always 0. 1: Clears the data consistency error flag (CSIHnSTR0.CSIHnDCE).
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	CSIHnPEC	Controls the parity error flag clear command. 0: No operation. The read value is always 0. 1: Clears the parity error flag (CSIHnSTR0.CSIHnPE).
0	CSIHnOVEC	Controls the overrun error flag clear command. 0: No operation. The read value is always 0. 1: Clears the overrun error flag (CSIHnSTR0.CSIHnOVE).

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.7 CSIHnMCTL0 — CSIHn Memory Control Register 0

This register selects the memory mode and the time-out setting.

Access: This register can be read or written in 16-bit units.

Address: <CSIHn_base> + 1040_H

Value after reset: $001F_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CSIHn MMS[1:0]		—	—	—	CSIHnTO[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 16.21 CSIHnMCTL0 Register Contents

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 8	CSIHnMMS [1:0]	Selects the memory mode. **CSIHn MMS1 / CSIHn MMS0 / Description** 0 / 0 / FIFO mode 0 / 1 / Dual buffer mode 1 / 0 / Transmit-only buffer mode 1 / 1 / Prohibited After a change of the memory mode, the respective buffer pointers must be cleared by setting the CSIHnSTCR0.CSIHnPCT bit to 1. In direct access mode, the setting of these bits is ignored.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	CSIHnTO[4:0]	Selects the time-out setting in FIFO mode. **CSIHnTO[4:0] / Description** 00000_B / No time-out is detected 00001_B / Time-out is (1 × 8 × BRG output clocks) 00010_B / Time-out is (2 × 8 × BRG output clocks) ... 11111_B / Time-out is (31 × 8 × BRG output clocks) **CAUTION** Changing the time-out setting is only permitted when CSIHnCTL0.CSIHnPWR = 0. Set the CSIHnTO[4:0] bits to 00000_B in direct access mode, dual buffer mode, or transmit-only buffer mode (except FIFO mode). For details about time-out detection, see also **Section 16.5.12.3, Time-Out Error**.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.8 CSIHnMCTL1 — CSIHn Memory Control Register 1

This register selects the conditions to generate the interrupt requests INTCSIHTIC and INTCSIHTIR in FIFO mode.

Access: This register can be read or written in 32-bit units.

Address: <CSIHn_base> + 1000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CSIHnFES[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CSIHnFFS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.22 CSIHnMCTL1 Register Contents

Bit Position	Bit name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	CSIHnFES[6:0]	Selects the conditions to generate the INTCSIHTIC interrupt (transmit data empty) in FIFO mode. When the number of untransmitted data to be transmitted in FIFO (checked by the CSIHnSTR0.CSIHnSPF[7:0] bit) and CSIHnMCTL1.CSIHnFES[6:0] match, the FIFO empty flag (CSIHnSTR0.CSIHnEMF bit) is set to 1, and the INTCSIHTIC interrupt request is generated.
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	CSIHnFFS[6:0]	Selects the conditions to generate the INTCSIHTIR interrupt (receive data full) in FIFO mode. When the number of received data in FIFO (checked by the CSIHnSTR0.CSIHnSRP[7:0] bit) and (128 - CSIHnMCTL1.CSIHnFFS[6:0]) match, the INTCSIHTIR interrupt request is generated.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.9 CSIHnMCTL2 — CSIHn Memory Control Register 2

This register controls the operation of the memory in dual buffer or transmit-only buffer mode and triggers to start communication.

Access: This register can be read or written in 32-bit units.

Address: <CSIHn_base> + 1004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHn BTST	—	—	—	—	—	—	—	CSIHnND[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CSIHnSOP[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.23 CSIHnMCTL2 Register Contents (1/2)

Bit Position	Bit Name	Function
31	CSIHnBTST	Provides a start trigger for buffer transfer. 0: No operation. 1: Issues the start transfer command. The read value is always 0. **CAUTION** This bit can only be used in dual buffer mode and transmit-only buffer mode.
30 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	CSIHnND[7:0]	Specifies the number of data for each memory mode. The read value indicates the number of remaining communication data.

CSIHn ND[7:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00_H	Send 0 data	Send 0 data	No influence	No influence
01_H	Send 1 data	Send 1 data	No influence	No influence
...	...	...	No influence	No influence
$3F_H$	Send 63 data	Send 63 data	No influence	No influence
40_H	Send 64 data	Send 64 data	No influence	No influence
...	Prohibited	...	No influence	No influence
$7F_H$	Prohibited	Send 127 data	No influence	No influence
80_H	Prohibited	Send 128 data	No influence	No influence
Other than the above	Setting is prohibited.			

The values are automatically decremented after data transfer (not decremented in direct access mode).

15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 16.23 CSIHnMCTL2 Register Contents (2/2)

Bit Position	Bit Name	Function
6 to 0	CSIHnSOP[6:0]	Selects the pointer of the data to be sent. If communication is forced to stop by setting CSIHnCTL0.CSIHnPWR to 0 or CSIHnSTCR0.CSIHnPCT to 1, these bits are cleared by hardware. In FIFO mode, these bits indicate the send address.

CSIHn SOP[6:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00_H	0000_H	0000_H	0000_H	No influence
01_H	0004_H	0004_H	0004_H	No influence
...	...	...	...	No influence
$3F_H$	$00FC_H$	$00FC_H$	$00FC_H$	No influence
40_H	Prohibited	0100_H	0100_H	No influence
...	Prohibited	...	...	No influence
$7F_H$	Prohibited	$01FC_H$	$01FC_H$	No influence

CAUTION

In direct access mode, these bits are not incremented.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.10 CSIHnMRWP0 — CSIHn Memory Read/Write Pointer Register 0

This register sets the pointers for reading from and writing to the dual buffer or transmit-only buffer.

Access: This register can be read or written in 32-bit units.

Address: <CSIHn_base> + 1018_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CSIHnRRA[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CSIHnTRWA[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.24 CSIHnMRWP0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	CSIHnRRA[6:0]	Selects the read pointer of the receive buffer.

CSIHn RRA[6:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00_H	0000_H	No influence	0000_H	No influence
01_H	0004_H	No influence	0004_H	No influence
...	...	No influence	...	No influence
$3F_H$	$00FC_H$	No influence	$00FC_H$	No influence
40_H	Prohibited	No influence	0100_H	No influence
...	Prohibited	No influence	...	No influence
$7F_H$	Prohibited	No influence	$01FC_H$	No influence

These bits are automatically incremented when received data is read.
If an overrun error is generated while reading the CSIHnRX0W or CSIHnRX0H register, the read pointer is not incremented.
These bits are cleared when CSIHnSTCR0.CSIHnPCT is set to 1.
In direct access mode and transmit-only buffer mode, these bits are not incremented.
To perform write access in transmit-only buffer mode, set 0000_H.
In FIFO mode, these bits indicate the read address of the received data.

15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 16.24 CSIHnMRWP0 Register Contents (2/2)

Bit Position	Bit Name	Function
6 to 0	CSIHnTRWA [6:0]	Selects the read/write pointer of the transmit buffer.

CSIHn TRWA[6:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00_H	0000_H	0000_H	0000_H	No influence
01_H	0004_H	0004_H	0004_H	No influence
...	...	...	...	No influence
$3F_H$	$00FC_H$	$00FC_H$	$00FC_H$	No influence
40_H	Prohibited	0100_H	0100_H	No influence
...	Prohibited	...	...	No influence
$7F_H$	Prohibited	$01FC_H$	$01FC_H$	No influence

These bits are automatically incremented when the transmission data is written or read.
These bits are cleared when CSIHnSTCR0.CSIHnPCT is set to 1.
In direct access mode, these bits are not incremented.
In FIFO mode, these bits indicate the read/write address of transmission data.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.11 CSIHnCFGx — CSIHn Configuration Register x

These eight registers configure the prescaler, parity, data length, recessive configuration for broadcasting, serial data direction, clock phase and data phase, idle enforcement configuration, idle time, hold time, inter-data time, and setup for each chip select signal, CSIHTCSSx.

Slave mode

In slave mode, the transmission protocol setting of the CSIHnCFG0 register is effective.

- CSIHnPSx[1:0]: parity usage
- CSIHnDLSx[3:0]: data length selection
- CSIHnDIRx: data direction
- CSIHnCKPx, CSIHnDAPx: clock phase and data phase

In slave mode, set all bits other than above in the CSIHnCFG0 register, and the CSIHnCFG1 to CSIHnCFG7 registers to 0.

Access: This register can be read or written in 32-bit units.

Address: CSIHnCFG0: <CSIHn_base> + 1044_H
CSIHnCFG1: <CSIHn_base> + 1048_H
CSIHnCFG2: <CSIHn_base> + $104C_H$
CSIHnCFG3: <CSIHn_base> + 1050_H
CSIHnCFG4: <CSIHn_base> + 1054_H
CSIHnCFG5: <CSIHn_base> + 1058_H
CSIHnCFG6: <CSIHn_base> + $105C_H$
CSIHnCFG7: <CSIHn_base> + 1060_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnBRSSx[1:0]		CSIHnPSx[1:0]		CSIHnDLSx[3:0]				—	—	—	—	CSIHnRCBx	CSIHnDIRx	CSIHnCKPx	CSIHnDAPx
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnIDLx	CSIHnIDx[2:0]			CSIHnHDx[3:0]				CSIHnINx[3:0]				CSIHnSPx[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

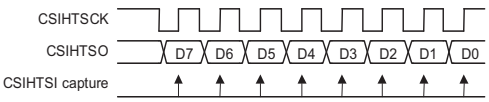
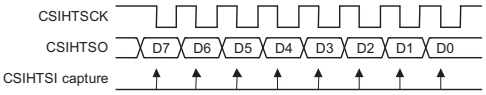
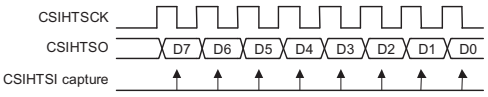
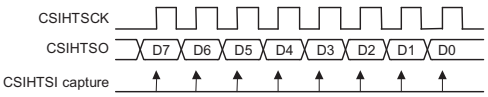
Table 16.25 CSIHnCFGx Register Contents (1/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>31, 30</td><td>CSIHnBRSSx [1:0]</td><td>These bits select the baud rate setting register (CSIHnBRSy).
<table>
<tr><th>CSIHn BRSSx1</th><th>CSIHn BRSSx0</th><th>Baud Rate Setting Register Selection</th></tr>
<tr><td>0</td><td>0</td><td>The transfer clock frequency is set according to the CSIHnBRS0 setting.</td></tr>
<tr><td>0</td><td>1</td><td>The transfer clock frequency is set according to the CSIHnBRS1 setting.</td></tr>
<tr><td>1</td><td>0</td><td>The transfer clock frequency is set according to the CSIHnBRS2 setting.</td></tr>
<tr><td>1</td><td>1</td><td>The transfer clock frequency is set according to the CSIHnBRS3 setting.</td></tr>
</table>
The maximum value for setting the transfer clock frequency, in accordance with the CSIHnCTL2.CSIHnPRS[2:0] setting, must be as follows:
Master mode: PCLK/4
Slave mode: PCLK/6</td></tr>
<tr><td>29, 28</td><td>CSIHnPSx[1:0]</td><td>Selects the parity for transmitting or receiving chip select signal x.
<table>
<tr><th>CSIHn PSx1</th><th>CSIHn PSx0</th><th>Transmission</th><th>Reception</th></tr>
<tr><td>0</td><td>0</td><td>Does not transmit any parity bit.</td><td>Does not wait for reception of the parity bit.</td></tr>
<tr><td>0</td><td>1</td><td>Adds a parity bit fixed to 0.</td><td>Waits for reception of the parity bit but does not evaluate it.</td></tr>
<tr><td>1</td><td>0</td><td>Adds an odd parity bit.</td><td>Waits for the odd parity bit.</td></tr>
<tr><td>1</td><td>1</td><td>Adds an even parity bit.</td><td>Waits for the even parity bit.</td></tr>
</table></td></tr>
<tr><td>27 to 24</td><td>CSIHnDLSx [3:0]</td><td>Selects the data length for chip select signal x.
<table>
<tr><th>CSIHnDLSx[3:0]</th><th>Data Length</th></tr>
<tr><td>0000_B</td><td>16 bits</td></tr>
<tr><td>0001_B</td><td>1 bit</td></tr>
<tr><td>0010_B</td><td>2 bits</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>1111_B</td><td>15 bits</td></tr>
</table>
CAUTION
When CSIHnTX0W.CSIHnEDL = 1, the setting of this bit has no effect. When CSIHnTX0W.CSIHnEDL = 0 (the data length is 16 bits), the setting of this bit is valid. Setting "1 bit" is only enabled if the previous transmit data was 16 bits with CSIHnEDL = 1.</td></tr>
<tr><td>23 to 20</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
<tr><td>19</td><td>CSIHnRCBx</td><td>Selects the recessive configuration for broadcasting for chip select signal x.
0: Dominant (higher priority)
1: Recessive (lower priority)
For details, see Section 16.5.3.1, Configuration Registers</td></tr>
<tr><td>18</td><td>CSIHnDIRx</td><td>Selects the serial data direction of chip select signal x.
0: Data is transmitted/received with MSB first.
1: Data is transmitted/received with LSB first.
For details, see Section 16.5.9, Serial Data Direction Selection.</td></tr>
</table>

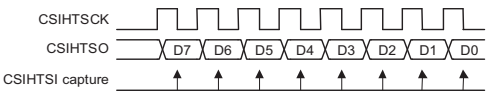
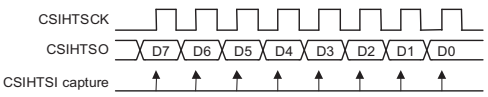
Table 16.25 CSIHnCFGx Register Contents (2/4)

Bit Position	Bit Name	Function
17	CSIHnCKPx	CSIHnCKPx: Clock phase selection bit CSIHnDAPx: Data phase selection bit
16	CSIHnDAPx	

- CSIHnCTL1.CSIHnCKR = 0

CSIHn CKPx	CSIHn DAPx	Clock Phase and Data Phase Selection
0	0	
0	1	
1	0	
1	1	

- CSIHnCTL1.CSIHnCKR = 1

CSIHn CKPx	CSIHn DAPx	Clock Phase and Data Phase Selection
0	0	
0	1	
1	—	Setting prohibited

15	CSIHnIDLx	Selects the idle enforcement configuration for chip select signal x. 0: If the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers are different, an idle state is inserted between two transfers. If the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers are the same, an idle state is not inserted between two transfers. 1: Regardless of the CSIHnTX0W.CSIHnCSx settings of two consecutive transfers, an idle state is inserted between two transfers. This bit is only available in master mode. For details about the enforced idle state, see **Section 16.5.15, Enforced Chip Select Idle Setting**.

Table 16.25 CSIHnCFGx Register Contents (3/4)

Bit Position	Bit Name	Function
14 to 12	CSIHnIDx[2:0]	Selects the idle time for chip select signal x. These bits are only available in master mode.

CSIHn IDx[2:0]	Idle time
000_B	0.5 transmission clock cycles
001_B	1.0 transmission clock cycle
010_B	1.5 transmission clock cycles
011_B	2.5 transmission clock cycles
100_B	3.5 transmission clock cycles
101_B	4.5 transmission clock cycles
110_B	6.5 transmission clock cycles
111_B	8.5 transmission clock cycles

11 to 8	CSIHnHDx[3:0]	Specifies the hold time for chip select signal x in transmission clock cycles. These bits are only available in master mode.

CSIHn HDx[3:0]	Hold time when CSIHnCTL1.CSIHnSIT is 0	Hold time when CSIHnCTL1.CSIHnSIT is 1
0000_B	0.5 transmission clock cycle	1.0 transmission clock cycle
0001_B	1.0 transmission clock cycle	1.5 transmission clock cycles
0010_B	1.5 transmission clock cycles	2.0 transmission clock cycles
0011_B	2.5 transmission clock cycles	3.0 transmission clock cycles
0100_B	3.5 transmission clock cycles	4.0 transmission clock cycles
0101_B	4.5 transmission clock cycles	5.0 transmission clock cycles
0110_B	6.5 transmission clock cycles	7.0 transmission clock cycles
0111_B	8.5 transmission clock cycles	9.0 transmission clock cycles
1000_B	9.5 transmission clock cycles	10.0 transmission clock cycles
1001_B	10.5 transmission clock cycles	11.0 transmission clock cycles
1010_B	11.5 transmission clock cycles	12.0 transmission clock cycles
1011_B	12.5 transmission clock cycles	13.0 transmission clock cycles
1100_B	14.5 transmission clock cycles	15.0 transmission clock cycles
1101_B	16.5 transmission clock cycles	17.0 transmission clock cycles
1110_B	18.5 transmission clock cycles	19.0 transmission clock cycles
1111_B	20.5 transmission clock cycles	21.0 transmission clock cycles

Table 16.25 CSIHnCFGx Register Contents (4/4)

Bit Position	Bit Name	Function
7 to 4	CSIHnINx[3:0]	Specifies the inter-data time for chip select signal x in transmission clock cycles.

CSIHn INx[3:0]	Inter-Data Time when CSIHnCTL1.CSIHnSIT is 0	Inter-Data Time when CSIHnCTL1.CSIHnSIT is 1
0000_B	0.0 transmission clock cycle	0.5 transmission clock cycle
0001_B	0.5 transmission clock cycle	1.0 transmission clock cycle
0010_B	1.0 transmission clock cycle	1.5 transmission clock cycles
0011_B	2.0 transmission clock cycles	2.5 transmission clock cycles
0100_B	3.0 transmission clock cycles	3.5 transmission clock cycles
0101_B	4.0 transmission clock cycles	4.5 transmission clock cycles
0110_B	6.0 transmission clock cycles	6.5 transmission clock cycles
0111_B	8.0 transmission clock cycles	8.5 transmission clock cycles
1000_B	9.0 transmission clock cycles	9.5 transmission clock cycles
1001_B	10.0 transmission clock cycles	10.5 transmission clock cycles
1010_B	11.0 transmission clock cycles	11.5 transmission clock cycles
1011_B	12.0 transmission clock cycles	12.5 transmission clock cycles
1100_B	14.0 transmission clock cycles	14.5 transmission clock cycles
1101_B	16.0 transmission clock cycles	16.5 transmission clock cycles
1110_B	18.0 transmission clock cycles	18.5 transmission clock cycles
1111_B	20.0 transmission clock cycles	20.5 transmission clock cycles

These bits are only available in master mode.

3 to 0	CSIHnSPx[3:0]	Specifies the setup time for chip select signal x in transmission clock cycles.

CSIHnSPx[3:0]	Setup Time
0000_B	0.5 transmission clock cycle
0001_B	1.0 transmission clock cycle
0010_B	1.5 transmission clock cycles
0011_B	2.5 transmission clock cycles
0100_B	3.5 transmission clock cycles
0101_B	4.5 transmission clock cycles
0110_B	6.5 transmission clock cycles
0111_B	8.5 transmission clock cycles
1000_B	9.5 transmission clock cycles
1001_B	10.5 transmission clock cycles
1010_B	11.5 transmission clock cycles
1011_B	12.5 transmission clock cycles
1100_B	14.5 transmission clock cycles
1101_B	16.5 transmission clock cycles
1110_B	18.5 transmission clock cycles
1111_B	20.5 transmission clock cycles

These bits are only available in master mode.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.12 CSIHnTX0W — CSIHn Transmit Data Register 0 for Word Access

This register stores transmission data. In addition, it specifies the communication interrupt request, the end-of-job, the extended data length, and the chip select activation.

Access: This register can be read or written in 32-bit units.

Address: <CSIHn_base> + 1008_H

Value after reset: $X0XX\ XXXX_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnCIRE	CSIHnEOJ	CSIHnEDL	—	—	—	—	—	CSIHnCS7	CSIHnCS6	CSIHnCS5	CSIHnCS4	CSIHnCS3	CSIHnCS2	CSIHnCS1	CSIHnCS0
Value after reset	—	—	—	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnTX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.26 CSIHnTX0W Register Contents (1/2)

Bit Position	Bit name	Function
31	CSIHnCIRE	Enables the communication interrupt request INTCSIHTIC in dual buffer or transmit-only buffer mode, or the job completion interrupt request INTCSIHTIJC in FIFO mode. 0: No interrupt is requested. 1: An interrupt is requested. Generates interrupt INTCSIHTIC or INTCSIHTIJC after transmission. For details, see **Section 16.4.3, INTCSIHTIC (Communication Status Interrupt)** and **Section 16.4.6, INTCSIHTIJC (Job Completion Interrupt)**. **CAUTION** This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
30	CSIHnEOJ	Specifies the end of a job. 0: Indicates that it is not end-of-job data. The job continues. 1: Indicates end-of-job data. **CAUTION** This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). This bit must be set to 0 in slave mode.
29	CSIHnEDL	Specifies whether the associated data requires the extended data length (EDL) option. 0: Normal operation 1: Enables the extended data length. The associated data is transmitted as a 16-bit packet. No inter-data time or idle time will be inserted after the data is transmitted. If CSIHnCTL1.CSIHnEDLE = 1 and CSIHnTX0W.CSIHnEDL = 1, the subsequent data must have the same CS selection. If CS is modified for the subsequent data, the correct operation is not assured. **CAUTION** This bit is only available if CSIHnCTL1.CSIHnEDLE = 1.
28 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 16.26 CSIHnTX0W Register Contents (2/2)

Bit Position	Bit name	Function
23 to 16	CSIHnCS[7:0]	Activates one or more chip select signals. 0: Activates chip select signals x for the associated transmission. 1: Deactivates chip select signals x for the associated transmission. Setting CSIHnTX0W.CSIHnCS[7:0] = FF_H is prohibited. **CAUTION** If several chip select signals are enabled for broadcasting, the configuration of one with CSIHnCFGx.CSIHnRCBx = 0 (dominant) is used. In this case, all dominant chip select signals must be set to precisely the same value. In slave mode, set the CSIHnCS[7:0] bit to FE_H.
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.13 CSIHnTX0H — CSIHn Transmit Data Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 of register CSIHnTX0W.

The settings specified by the upper 16 bits of CSIHnTX0W are applied to the transmission. Set transmit data to CSIHnTX0W before using this register because the value of CSIHnTX0W is undefined after the reset.

Access: This register can be read or written in 16-bit units.

Address: <CSIHn_base> + $100C_H$

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnTX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.27 CSIHnTX0H Register Contents

Bit Position	Bit name	Function
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.14 CSIHnRX0W — CSIHn Receive Data Register 0 for Word Access

This register stores the received data.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <CSIHn_base> + 1010_H

Value after reset: $0XXX\ XXXX_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CSIHn RPE	CSIHn TDCE	CSIHn CS7	CSIHn CS6	CSIHn CS5	CSIHn CS4	CSIHn CS3	CSIHn CS2	CSIHn CS1	CSIHn CS0
Value after reset	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnRX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 16.28 CSIHnRX0W Register Contents

Bit Position	Bit name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	CSIHnRPE	Indicates whether a reception data parity error was detected. 0: No parity error was detected on the associated reception data. 1: A parity error was detected on the associated reception data.
24	CSIHnTDCE	Indicates whether a transmission data consistency error was detected. 0: No consistency error was detected on the associated transmission data. 1: A consistency error was detected on the associated transmission data.
23 to 16	CSIHnCSx (x = 7 to 0)	Indicates which chip select signal was activated. 0: Chip select signal x was activated for the associated reception. 1: Chip select signal x was deactivated for the associated reception.
15 to 0	CSIHnRX[15:0]	Stores the received data.

NOTE

Read this register at least one serial clock cycle before an interrupt is generated.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.15 CSIHnRX0H — CSIHn Receive Data Register 0 for Half Word Access

This register stores the received data. This register is the same as bits 15 to 0 of register CSIHnRX0W.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <CSIHn_base> + 1014_H

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnRX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 16.29 CSIHnRX0H Register Contents

Bit Position	Bit name	Function
15 to 0	CSIHnRX[15:0]	Stores the received data.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.16 CSIHnEMU — CSIHn Emulation Register

This register controls operation of SVSTOP.

Access: This register can be read or written in 8-bit or 1-bit units.
Perform write operation when (EPC.SVSTOP = 0).

Address: <CSIHn_base> + 0018_H

Value after reset: 00_H

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

Table 16.30 CSIHnEMU Register Contents

Bit Position	Bit name	Function
7	CSIHn SVSDIS	Selects whether to continue or stop transmit/receive operation during debugging. • When the EPC.SVSTOP bit is set to 0 Continues transmit/receive operation regardless of the setting of this bit. • When the EPC.SVSTOP bit is set to 1 0: Stops transmit/receive operation. 1: Continues transmit/receive operation.
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.17 CSIHnBRSy — CSIHn Baud Rate Setting Register y (y = 0 to 3)

This register sets the transfer clock frequency for each chip select signal.

With CSIHnCFG0 to 7.CSIHnBRSSx[1:0] bits, one of the four types of transfer clock frequency settings can be selected for each chip select signal. For details of transfer clock frequency setting, see **Section 16.5.5, Transmission Clock Selection**.

Access: This register can be read or written in 16-bit units.

Address: CSIHnBRS0: <CSIHn_base> + 1068_H
CSIHnBRS1: <CSIHn_base> + $106C_H$
CSIHnBRS2: <CSIHn_base> + 1070_H
CSIHnBRS3: <CSIHn_base> + 1074_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CSIHnBRS[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.31 CSIHnBRSy Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	CSIHnBRS [11:0]	0: BRG stopped 1: PCLK / (2^{α} × 1 × 2) 2: PCLK / (2^{α} × 2 × 2) 3: PCLK / (2^{α} × 3 × 2) 4: PCLK / (2^{α} × 4 × 2) . . . 4095: PCLK / (2^{α} × 4095 × 2) α is the value of CSIHnCTL2.CSIHnPRS[2:0].

CAUTION

When setting this register, see **Table 16.32, Notes on Setting Registers**.

16.3.18 List of Cautions

Table 16.32 Notes on Setting Registers (1/2)

Register Name	Bit Name	Cautions
CSIHnCTL0	CSIHnPWR	If this bit is cleared during communication, ongoing communication is aborted. After the communication is aborted, it is necessary to restart the communication.
CSIHnCTL0	CSIHnTXE CSIHnRXE	Do not modify any of these bits while CSIHnCTL0.CSIHnPWR = 0. (These bits can be modified at the same time as the CSIHnCTL0.CSIHnPWR bit.) Do not modify these bits while CSIHnSTR0.CSIHnTSF = 1, because the specified operation is not guaranteed if ongoing communication is aborted.
CSIHnCTL0	CSIHnJOBE	Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid when CSIHnCTL1.CSIHnJE = 1. Setting this bit is prohibited in slave mode.
CSIHnCTL0	CSIHnMBS	Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. (This bit can be modified at the same time as the CSIHnCTL0.CSIHnPWR bit.) Modification of this bit is only permitted while CSIHnSTR0.CSIHnTSF = 0. Do not change the mode between FIFO mode and direct access mode while CSIHnCTL0.CSIHnPWR = 1. When the CPU-controlled high-priority communication is enabled, the operation is the same as that in direct access mode regardless of the CSIHnMBS bit setting.
CSIHnCTL1	CSIHnCKR	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. When CS is not used, use this bit instead of CSIHnCFGx.CSIHnCKPx and set CSIHnCFGx.CSIHnCKPx to 0. This bit must be used in slave mode.
CSIHnCTL1	CSIHnSLIT CSIHnCSL[7:0] CSIHnEDLE CSIHnDCS CSIHnCSRI CSIHnHSE	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0.
CSIHnCTL1	CSIHnPHE CSIHnJE CSIHnLBM	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of this bit is prohibited in slave mode.
CSIHnCTL1	CSIHnSSE	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting this bit to 1 is prohibited in master mode.
CSIHnCTL1	CSIHnSIT	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid in master mode. In slave mode, no delay is generated.
CSIHnCTL2	CSIHnPRS[2:0]	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of the maximum transfer clock frequency is as follows. • Master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4) • Slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)
CSIHnSTR0	CSIHnSRP[7:0] CSIHnSPF[7:0] CSIHnFLF CSIHnEMF CSIHnTSF CSIHnHPST	Writing to these bits is prohibited, and only reading is permitted.
CSIHnSTR0	CSIHnTMOE CSIHnOFE CSIHnDCE CSIHnPE CSIHnOVE	Writing to these bits is prohibited, and only reading is permitted. These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0.
CSIHnSTCR0	CSIHnPCT	If this bit is set to 1 during communication, ongoing communication is aborted.
CSIHnMCTL0	CSIHnMMS[1:0]	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0 and CSIHnCTL0.CSIHnMBS = 0.
CSIHnMCTL0	CSIHnTO[4:0]	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. Set these bits to 0 in master mode. Set these bits to 0 in direct access, dual buffer, and transmit-only buffer mode.
CSIHnMCTL1	CSIHnFES[6:0] CSIHnFFS[6:0]	Writing to these bits while communication is ongoing is permitted.

Table 16.32 Notes on Setting Registers (2/2)

Register Name	Bit Name	Cautions
CSIHnMCTL2	CSIHnBTST CSIHnND[7:0] CSIHnSOP[6:0]	Writing to these bits is prohibited when CSIHnCTL0.CSIHnPWR = 0. Writing to these bits is prohibited when CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0. Writing to these bits is prohibited when CSIHnSTR0.CSIHnTSF = 1. Writing to these bits is prohibited in direct access or FIFO mode.
CSIHnMRWP0	CSIHnRRA[6:0]	Writing to these bits while communication is ongoing is permitted. Writing to these bits is prohibited in direct access or FIFO mode. When writing is required, set "0000_H" to these bits in transmit-only buffer mode.
CSIHnMRWP0	CSIHnTRWA[6:0]	Writing to these bits while communication is ongoing is permitted. Writing to these bits is prohibited in direct access or FIFO mode.
CSIHnCFGx x = 0 to 7	CSIHnBRSSx[1:0] CSIHnRCBx CSIHnIDLx CSIHnIDx[2:0] CSIHnHDx[3:0] CSIHnINx[3:0] CSIHnSPx[3:0]	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. These bits must be set to 0 in slave mode.
CSIHnCFGx x = 0 to 7	CSIHnPSx[1:0] CSIHnDLSx[3:0] CSIHnDIRx CSIHnDAPx	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. In slave mode, the CSIHnCFG0 setting is used for the configuration. Therefore, all the bits in CSIHnCFG1 to CSIHnCFG7 must be set to 0.
CSIHnCFGx x = 0 to 7	CSIHnCKPx	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. Set this bit to 0 as CSIHnCTL1.CSIHnCKR must be used in slave mode. If CS is not used, use the CSIHnCTL1.CSIHnCKR bit instead of this bit, and clear this bit to 0.
CSIHnTX0W	CSIHnEOJ CSIHnCIRE	These bits are only valid when CSIHnCTL1.CSIHnJE = 1. While CSIHnCTL1.CSIHnJE = 0, values of these bits are ignored even if 1 is read. Set these bits to 0 in slave mode.
CSIHnTX0W	CSIHnEDL	These bits are only valid when CSIHnCTL1.CSIHnEDLE = 1. While CSIHnCTL1.CSIHnEDLE = 0, the value of this bit is ignored even if 1 is read.
CSIHnTX0W	CSIHnCS[7:0]	In master mode, setting this bit to "FF_H" is prohibited. In slave mode, set this bit to "FE_H".
CSIHnTX0W CSIHnTX0H		Reading these bits while communication is ongoing is prohibited in FIFO mode. While CSIHnCTL0.CSIHnPWR = 0, reading and writing to these bits is prohibited in FIFO mode. While CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0, writing to these bits are prohibited in direct access mode.
CSIHnRX0W		These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1→ 0. While CSIHnCTL0.CSIHnPWR = 0, reading and writing these bits is prohibited in FIFO mode. While CSIHnCTL0.CSIHnPWR = 0, reading and writing these bits is permitted in the mode(Transmit-only buffer, Dual buffer and Direct access modes) except FIFO mode. While CSIHnCTL0.CSIHnPWR = 1, writing these bits is prohibited, and reading these bits is permitted. Writing is prohibited and only reading is permitted of these bits while CSIHnCTL0.CSIHnPWR=1.
CSIHnRX0H		These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1→ 0. While CSIHnCTL0.CSIHnPWR = 0, reading and writing these bits is prohibited in FIFO mode. Writing is prohibited and only reading is permitted of these bits while CSIHnCTL0.CSIHnPWR=1. While CSIHnCTL0.CSIHnPWR = 1, writing these bits is prohibited in the FIFO mode and reading these bits is prohibited in the FIFO mode. In spite of CSIHnCTL0.CSIHnPWR value, writing is prohibited and only reading is permitted of these bits in the mode(Transmit-only buffer, Dual buffer and Direct access modes) except FIFO mode.
CSIHnEMU	CSIHnSVSDIS	Modification of this bit is only permitted while SVSTOP = 0.
CSIHnBRSy y = 0 to 3		Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0.

16.4 Interrupt Sources

CSIH can generate the following interrupt requests:

- INTCSIHTIC (communication status interrupt)
- INTCSIHTIR (reception status interrupt)
- INTCSIHTIRE (communication error interrupt)
- INTCSIHTIJC (job completion interrupt)

16.4.1 Overview

The communication error interrupt INTCSIHTIRE is generated when an error is detected. The generation of the other interrupts depends on the memory mode, the job mode, and – in case of the job completion interrupt INTCSIHTIJC –also the operating mode.

The job completion interrupt INTCSIHTIJC is only generated when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). It is not available in slave mode.

The following table gives an overview.

Table 16.33 Interrupt Generation

Memory Mode	Interrupt	Interrupt Source	
		Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	**Job Mode Enabled CSIHnCTL1.CSIHnJE = 1**
FIFO	INTCSIHTIC	Tx data empty*1	Tx data empty*1 except job abort*4
	INTCSIHTIR	Rx data full*2 *and* CSIHnCTL0.CSIHnRXE = 1	Rx data full*2 *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*3	Not applicable	CSIHnTX0W.CSIHnCIRE = 1 (except Tx data empty), or job abort*4
Transmit-only buffer	INTCSIHTIC	End of communication	CSIHnTX0W.CSIHnCIRE = 1 *and* (CSIHnCTL0.CSIHnJOBE = 0 *or* CSIHnTX0W.CSIHnEOJ = 0)
	INTCSIHTIR	Data received *and* CSIHnCTL0.CSIHnRXE = 1	Data received *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*3	Not applicable	Job abort*4
Dual buffer	INTCSIHTIC	End of communication	CSIHnTX0W.CSIHnCIRE = 1 *and* (CSIHnCTL0.CSIHnJOBE = 0 *or* CSIHnTX0W.CSIHnEOJ = 0)
	INTCSIHTIR	End of communication *and* CSIHnCTL0.CSIHnRXE = 1	Data received *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*3	Not applicable	Job abort*4
Direct access	INTCSIHTIC	One data transfer	One data transfer except the state of job abortion*4
	INTCSIHTIR	Data received *and* CSIHnCTL0.CSIHnRXE = 1	Data received *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*3	Not applicable	Job abort*4

Note 1. “Tx data empty” refers to the FIFO fill level, defined by CSIHnMCTL1.CSIHnFES[6:0].

Note 2. “Rx data full” refers to the FIFO fill level, defined by CSIHnMCTL1.CSIHnFFS[6:0].
Note 3. INTCSIHTIJC is not available in slave mode.
Note 4. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1
During high priority communication in transmit-only buffer mode, the operation is the same as that in direct access mode.

16.4.2 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by half a cycle of the transmission clock, CSIHTSCK. This is not possible in slave mode.

The delay is specified by setting CSIHnCTL1.CSIHnSIT = 1. (The setting of the CSIGnSIT bit is invalid in slave mode.)

The following example illustrates the interrupt delay function, assuming a setting of CSIHnCTL1.CSIHnSIT = 1 (interrupt delay enabled), CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0 (clock phase and data phase), and CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B (data length 8 bits).

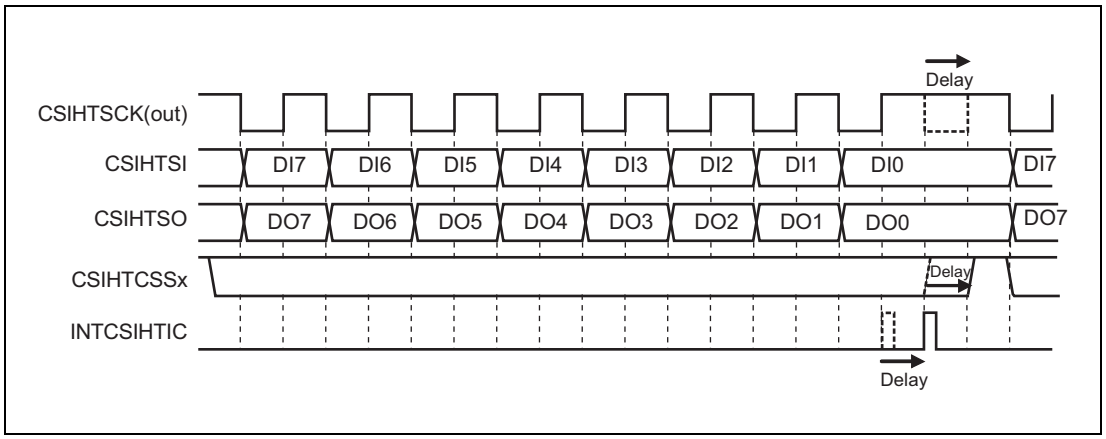


Figure 16.2 Interrupt Delay Function (CSIHnCTL1.CSIHnSIT = 1)

Setting CSIHnCTL1.CSIHnSIT = 1 adds a half cycle delay to the transmission clock. This also delays the end of the present chip select signal (CSIHTCSSx).

16.4.3 INTCSIHTIC (Communication Status Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the conditions shown in the following table.

Table 16.34 INTCSIHTIC Interrupt Generation

Memory Mode	Cause of Interrupt	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	**Job Mode Enabled CSIHnCTL1.CSIHnJE = 1**
FIFO	This interrupt is generated just before transmission data in the FIFO runs out, notifying the application that new data should be added. INTCSIHTIC is generated if the number of transmit data remaining in the FIFO (CSIHnSTR0.CSIHnSPF[7:0]) equals CSIHnMTCL1.CSIHnFES[6:0].	Similar to “when JE = 0”, an interrupt is generated when the number of transmit data remaining in the FIFO (CSIHnSTR0.CSIHnSPF[7:0]) equals CSIHnMCTL1.CSIHnFES[6:0]. However, it is not generated if a job is aborted.
Transmit-only buffer, dual buffer	Generated at the end of communication. (specified by the CSIHnMTLC2.CSIHnND[7:0] bit)	Generated when data is transmitted while CSIHnTX0W.CSIHnCIRE = 1 Note that if data and job abort*1 are transmitted while CSIHnTX0W.CSIHnCIRE = 1 the INTCSIHTIJC interrupt is generated instead of INTCSIHTIC.
Direct access	Generated after every data transfer.	Generated after every data transfer, except when the communication was aborted.

Note 1. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1.
During high priority communication in transmit-only buffer mode, the operation is the same as that in direct access mode.

16.4.3.1 INTCSIHTIC in Direct Access Mode

The examples below show the INTCSIHTIC behavior in direct access mode.

The examples assume:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

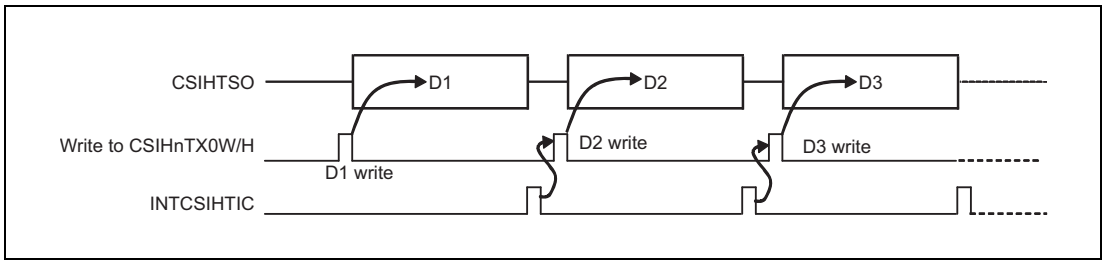


Figure 16.3 Generation of INTCSIHTIC after Transfer (CSIHnCTL1.CSIHnSLIT = 0)

If job mode is enabled (CSIHnCTL1.CSIHnJE = 1) and a job ends because data is sent with CSIHnTXOW.CSIHnEOJ = 1 and communication stop is requested (CSIHnCTL0.CSIHnJOBE = 1), then INTCSIHTIC is replaced by the job completion interrupt INTCSIHTIJC.

INTCSIHTIC can also be set up to occur as soon as the CSIHnTX0W/H register becomes empty and available for storing the next data. This is specified by setting CSIHnCTL1.CSIHnSLIT = 1.

The effect is illustrated in the figure below.

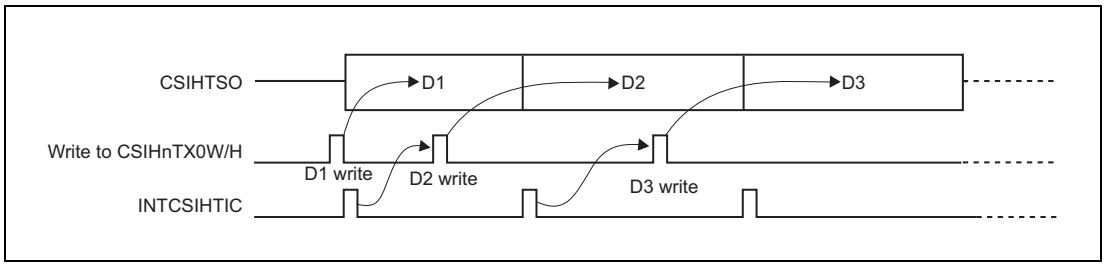


Figure 16.4 Immediate Generation of INTCSIHTIC (CSIHnCTL1.CSIHnSLIT = 1)

Thus, the new data can be written in advance.

NOTE

During high priority communication in transmit-only buffer mode, the operation is the same as that in direct access mode.

16.4.3.2 INTCSIHTIC in FIFO Mode

The example below shows the INTCSIHTIC behavior in FIFO mode.

The example assumes:

- Master mode
- FIFO mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

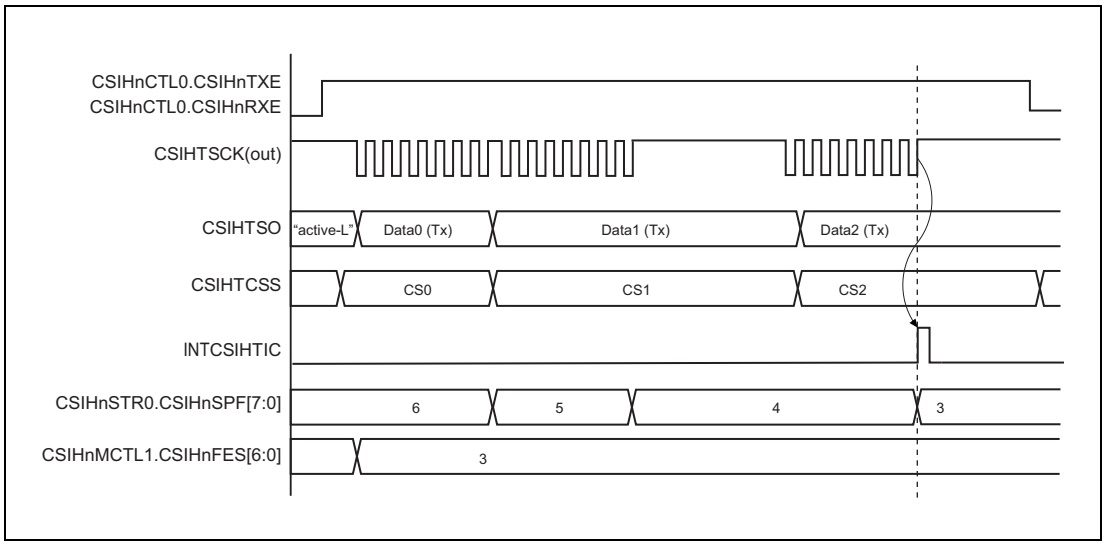


Figure 16.5 Generation of INTCSIHTIC in FIFO Memory Mode

The condition for “FIFO empty” is specified in CSIHnMCTL1.CSIHnFES[6:0]. In the example of the diagram above, the number of untransmitted data in FIFO is set to 3.

CSIHnSTR0.CSIHnSPF[7:0] indicates the number of untransmitted data. When both match, the interrupt INTCSIHTIC occurs.

16.4.3.3 INTCSIHTIC in Job Mode

The example below shows the INTCSIHTIC behavior in job mode.

The example assumes:

- Master mode
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

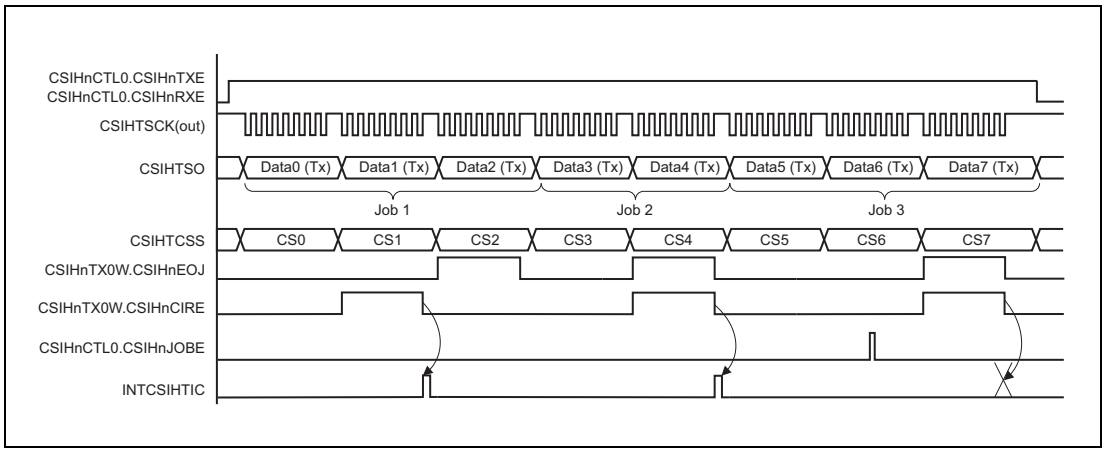


Figure 16.6 Generation of INTCSIHTIC in Job Mode

The rules for generating INTCSIHTIC in job mode are shown in the following table.

Table 16.35 Generation of INTCSIHTIC in Job Mode

CSIHnTX0W. CSIHnEOJ	CSIHnTX0W. CSIHnCIRE	INTCSIHTIC
0	0	Not generated
0	1	Generated
1	0	Not generated
1	1	CSIHnCTL0.CSIHnJOBE = 0: Generated
		CSIHnCTL0.CSIHnJOBE = 1: Not generated, replaced by interrupt INTCSIHTIJC

16.4.4 INTCSIHTIR (Reception Status Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the conditions below.

Table 16.36 INTCSIHTIR Interrupt Generation

Memory Mode	Cause of Interrupt	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	**Job Mode Enabled CSIHnCTL1.CSIHnJE = 1**
FIFO	This interrupt occurs when CSIHnCTL0.CSIHnRXE = 1 and the FIFO buffer is almost full with received data, notifying the application that the FIFO must be emptied. INTCSIHTIR is generated, if the number of received data in the FIFO (CSIHnSTR0.CSIHnSRP[7:0]) equals (128 – CSIHnMCTL1.CSIHnFFS[6:0]).	
Dual buffer	An interrupt is generated when the communication is finished (as specified by the CSIHnMCTL2.CSIHnND[7:0] bit) and CSIHnCTL0.CSIHnRXE = 1.	An interrupt is generated after every data transfer.
Transmit-only buffer, Direct access	An interrupt is generated after every data transfer.	

16.4.4.1 INTCSIHTIR in Direct Access Mode

The example below shows the INTCSIHTIR behavior in direct access mode.

The example below assumes:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

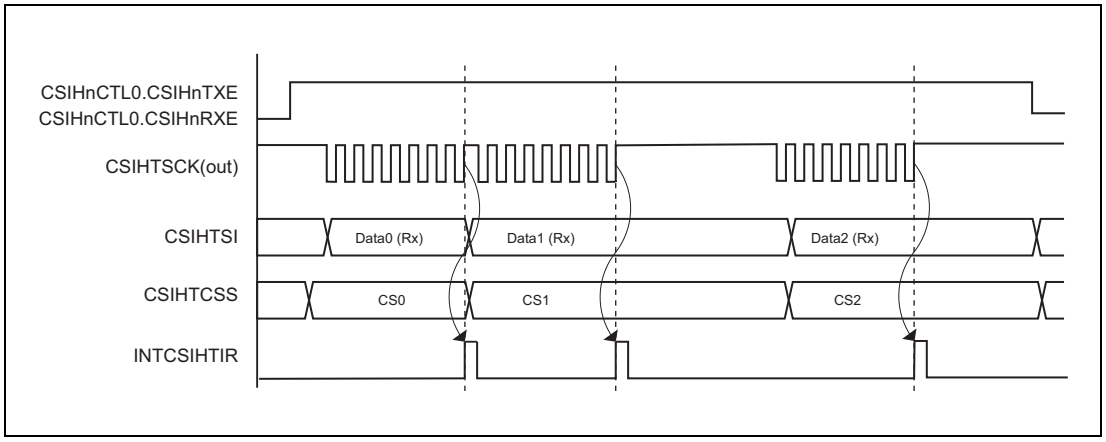


Figure 16.7 Generation of INTCSIHTIR in Direct Access Mode

16.4.4.2 INTCSIHTIR in Dual Buffer Mode

The example below shows the INTCSIHTIR behavior in dual buffer mode.

The example assumes:

- Master mode
- Dual buffer mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Default clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

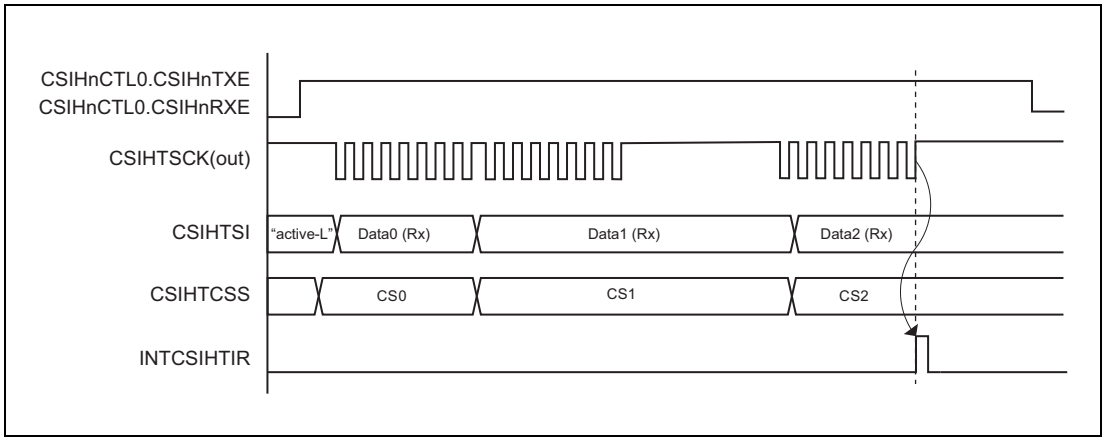


Figure 16.8 Generation of INTCSIHTIR in Dual Buffer Mode

16.4.5 INTCSIHTIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

For details about interrupt generation timing, see **Section 16.5.12, Error Detection**.

Table 16.37 Data Error Types (1/2)

Error Type	Communication Status after Error Interrupt	Note
FIFO overflow error	Communication continues even if an interrupt is generated.	The data is not written to the FIFO buffer and the data that overflowed is lost, but communications started before the error continue.
Parity error	Communication continues even if an interrupt is generated.	—
Data consistency error	Communication continues even if an interrupt is generated.	—

Table 16.37 Data Error Types (2/2)

Error Type	Communication Status after Error Interrupt	Note
Time-out error	Communication continues even if an interrupt is generated.	—
Overrun error	**Condition for errors 1:** In FIFO mode, when the number of received data is 0 and CPU reads the CSIHnRX0W/H register, an interrupt is generated and communication continues.	—
	Condition for errors 2: In slave mode, when CSIHnCTL1.CSIHnHSE = 0 (handshake function disabled): [1] In direct access mode or transmit-only buffer mode, when reception is completed while the previous received data is retained in the CSIHnRX0W/H register, an interrupt is generated, and communication continues. [2] In FIFO mode, when reception by the FIFO buffer is completed and the buffer is in the full state, an interrupt is generated. Communication continues.	In slave mode, when CSIHnCTL1.CSIHnHSE = 1 (handshake function enabled), communication is suspended due to handshake, an overrun error is not generated.

The type of error that caused the generation of INTCSIHTIRE is flagged in register CSIHnSTR0.

Additionally a parity error flag and a data consistency error flag are attached to the received data in CSIHnRX0W.

For details about the various error types, see **Section 16.5.12, Error Detection**.

16.4.6 INTCSIHTIJC (Job Completion Interrupt)

This interrupt supports the handling of jobs. For details, see **Section 16.5.3.3, Job Concept**. This interrupt is only available in master mode.

Job mode is enabled by setting CSIHnCTL1.CSIHnJE = 1. When CSIHnCTL1.CSIHnJE = 0, INTCSIHTIJC is not generated.

Depending on the memory mode, this interrupt is generated according to the condition shown in the following table.

Table 16.38 INTCSIHTIJC Interrupt Generation

Memory Mode	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	Not applicable	Indicates that the communication stopped at the end of a job after a job abort*1 is triggered If FIFO empty is not detected, INTCSIHTIJC is generated when CSIHnCIRE is 1.
Transmit-only buffer		Indicates that the communication stopped at the end of a job after a job abort*1 was triggered.
Dual buffer		
Direct access		

Note 1. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

16.5 Operation

16.5.1 Operating Modes (Master/Slave)

Whether CSIH operates in, the master or slave mode determines the source of the serial clock.

16.5.1.1 Master Mode

In master mode, the serial transmission clock is generated by the internal baud rate generator (BRG) and supplied to the slave(s) via CSIHTSCK signal.

Master mode is enabled by setting CSIHnCTL2.CSIHnPRS[2:0] to values other than 111_B. In master mode, the BRG frequency can be specified by setting the CSIHnCTL2.CSIHnPRS[2:0] bits in combination with the CSIHnBRSy.CSIHnBRS[11:0] bits.

(1) Chip select signals

In master mode, one or more chip select signals can be used. If several slaves are connected to the master, the chip select signals can be used to select one or more slaves. Only the selected slave is then enabled for communication.

The communication protocol as well as additional parameters are stored separately for each chip select signal. This makes it possible to adapt the data transfer individually to the requirements of each slave. For details, see **Section 16.5.3, Chip Selection (CS) Features**.

(2) Clock defaults

The default level of CSIHTSCK depends on the clock inversion function bit of the CSIHTSCK, and is high when CSIHnCTL1.CSIHnCKR = 0 and is low when CSIHnCTL1.CSIHnCKR = 1.

The example below shows the communication in master mode for 8-bit data, CSIHnCTL1.CSIHnCKR = 0, CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0, and MSB first.

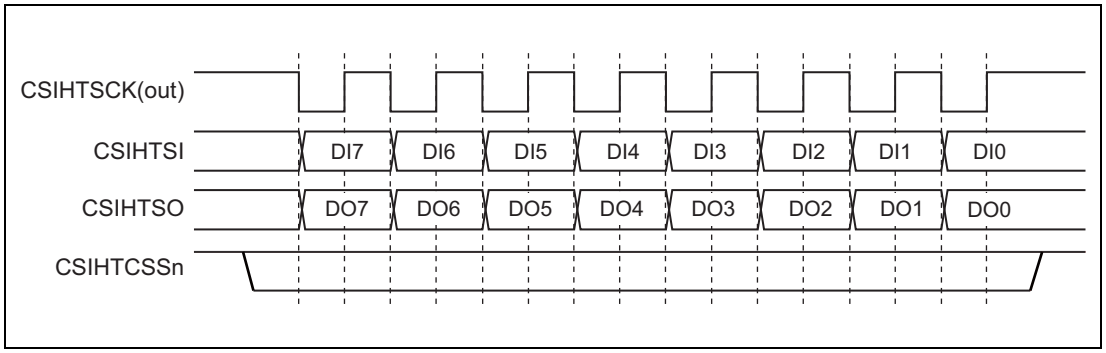


Figure 16.9 Transmission/Reception in Master Mode

16.5.1.2 Slave Mode

In slave mode, another device is the communication master and supplies the transmission clock. Normal transmit/receive operation is started as soon as a clock signal is detected.

Slave mode is selected by setting the CSIHnCTL2.CSIHnPRS[2:0] bits to 111_B.

In slave mode, the transmission protocol setting by the CSIHnCFG0 register is enabled (setting of the CSIHnCFG1-CSIHnCFG7 registers are disabled).

- CSIHnPSx[1:0]: parity usage
- CSIHnDLSx[3:0]: data length selection
- CSIHnDIRx: data direction
- CSIHnCKPx, CSIHnDAPx: clock phase and data phase

NOTE

When using slave mode, disable the baud rate generator (BRG) by setting the CSIHnBRSy.CSIHnBRS[11:0] bits to 000_H. However, if using a time-out error, set the CSIHnBRSy.CSIHnBRS[11:0] bits to a value other than 000_H.

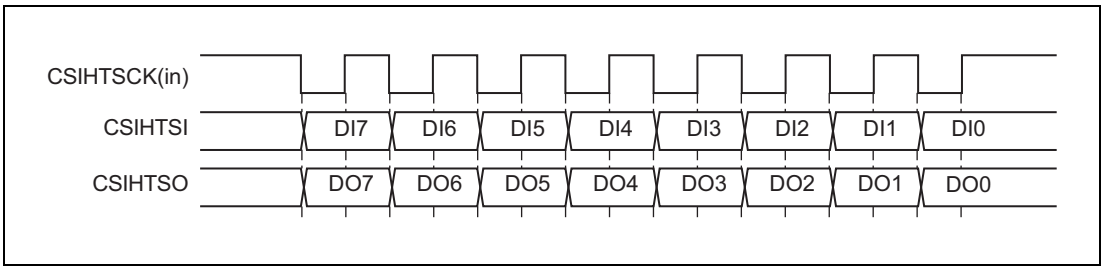


Figure 16.10 Transmission/Reception in Slave Mode

16.5.2 Master/Slave Connections

16.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

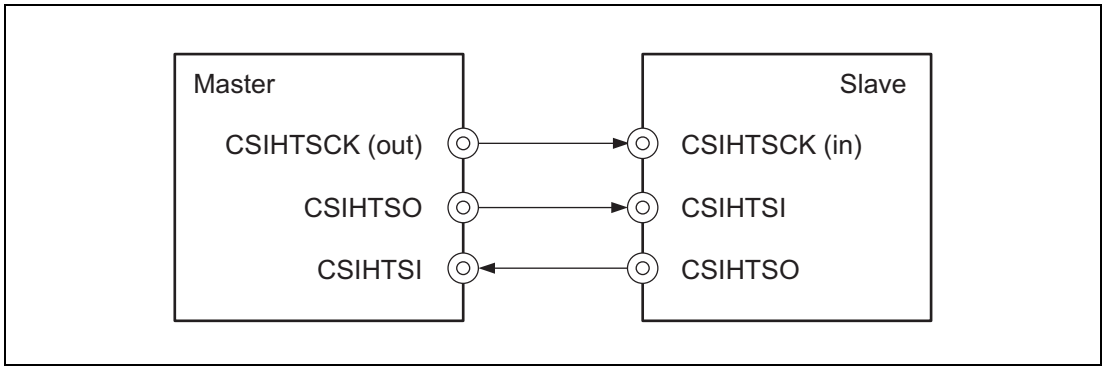


Figure 16.11 Direct Master/Slave Connection

16.5.2.2 One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this example, the master supplies one chip select (CS) signal to each of the slaves. This signal is connected to the slave select input $\overline{\text{CSIHTSSI}}$ of the slave.

The $\overline{\text{CSIHTSSI}}$ signal can be enabled/disabled by using the CSIHnCTL1.CSIHnSSE bit.

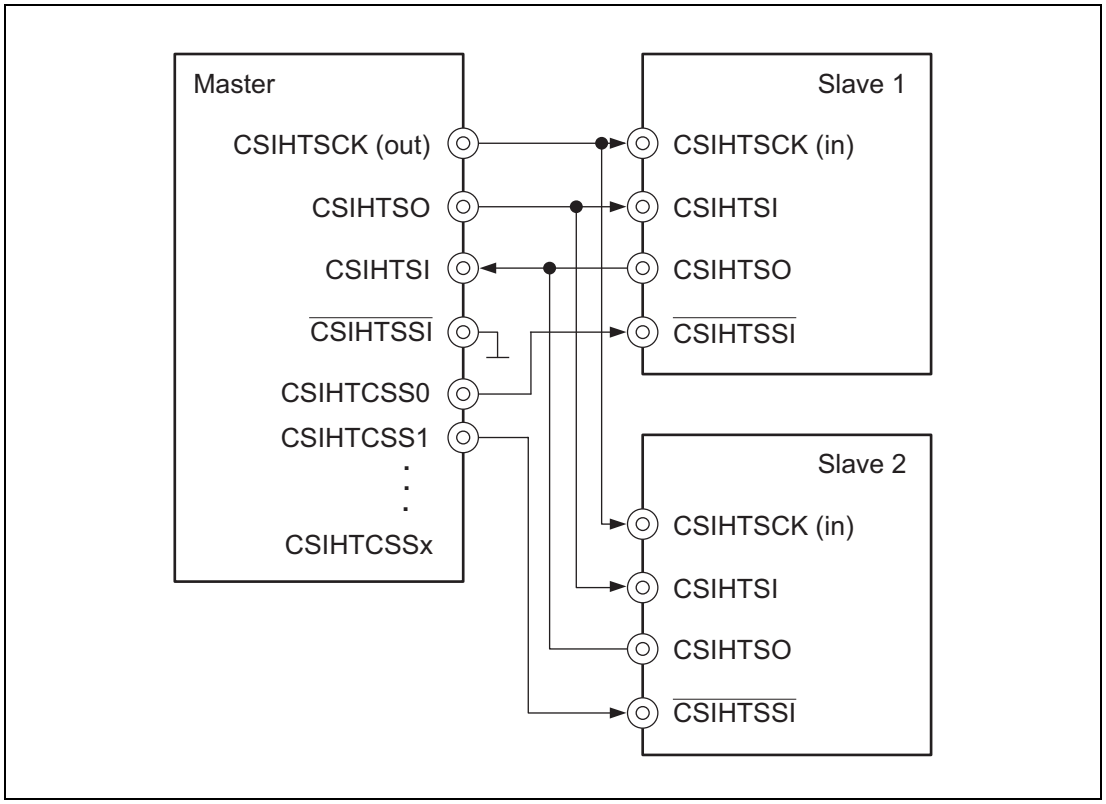


Figure 16.12 Connections between One Master and Multiple Slaves

By default, the chip select level is active low. That means, a slave is selected (enabled) as a CSIH slave when its $\overline{\text{CSIHTSSI}}$ signal is low level. However, to adapt the CS to other devices, the output level of each chip select signal can also be programmed to be active high.

If a slave is not selected, it will neither receive nor transmit data. In addition, when transmit-only mode or transmit/receive mode is set (CSIHnCTL0.CSIHnTXE = 1), the CSIHTSO output of the slaves that are not selected is disabled and set to input mode in order to avoid interference with the output of the selected slave.

16.5.3 Chip Selection (CS) Features

The chip select signal, CSIHTCSSx can be used by the master to select one or more slaves for communication.

16.5.3.1 Configuration Registers

The parameters for each chip select signal CSIHTCSSx are defined in the corresponding configuration register CSIHnCFGx. The parameters include the communication protocol and additional CS parameters.

The communication protocol specifies:

- Data length: The number of bits to be sent or received.
  (CSIHnCFGx.CSIHnDLSx[3:0])
- Transfer direction: MSB or LSB first.
  (CSIHnCFGx.CSIHnDIRx)
- Parity usage: Odd, even, 0 parity or none.
  (CSIHnCFGx.CSIHnPSx[1:0])
- Clock phase and data phase.
  (CSIHnCFGx.CSIHnCKPx, CSIHnCFGx.CSIHnDAPx)

Additional parameters for each chip select signal that are only available in master mode are:

- Individual selection of the baud rate generator prescaler for each chip select signal
  (CSIHnCFGx.CSIHnBRSSx[1:0])
- Chip select priority: Categorizes chip select signals into “dominant” and “recessive” The priority applies if two or more chip select signals with different configurations are simultaneously activated for message broadcasting. In this case, the configuration specified for dominant chip select signals is used. (CSIHnCFGx.CSIHnRCBx)

The principle is also called “Recessive Configuration for Broadcasting” (RCB).

CAUTION

When specifying multiple chip select signals as dominant, be sure to configure the same settings for all dominant signals.

- Chip select timing:
  - Setup time T_{setup}: The time from when the CS signal becomes active to the start of data output.
    (CSIHnCFGx.CSIHnSPx[3:0])
  - Inter-data time T_{inter}: The time between one data and the next data while the same CS signal is active.
    (CSIHnCFGx.CSIHnINx[3:0])
  - Hold time T_{hold}: The time during which the CS signal remains active until CS is switched.
    (CSIHnCFGx.CSIHnHDx[3:0])
  - Idle time T_{idle}: Inactive time after terminating a CS signal or after every data transfer to the same CSx. (CSIHnCFGx.CSIHnIDx[2:0])

The CS timings of the setup time, the inter-data time, the hold time, and the idle time are illustrated in the figure below. When CSIHnCFGx.CSIHnIDLx bit is set to 1, IDLE time is inserted for every transfer regardless of CS signal.

Figure 16.13 provides an example of when the default active low setting is specified for the CSIHTCSS1 and CSIHTCSS2 signals (CSIHnCTL1.CSIHnCSL1 bit = 0, CSIHnCTL1.CSIHnCSL2 bit = 0). The active level can be specified individually for each CS.

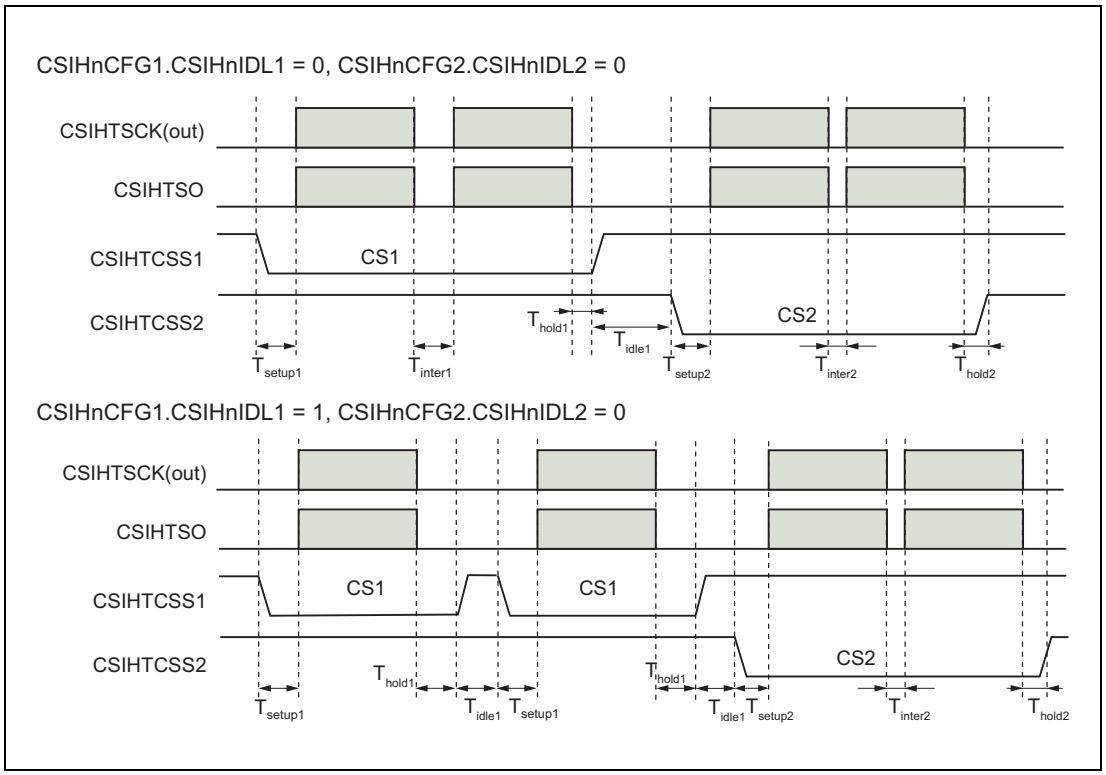


Figure 16.13 Chip Select Timings

Note that each CS signal can have a different value for the setup time, inter-data time, hold time, and idle time.

A particular chip select signal is activated by setting the appropriate bit in the transmission register CSIHnTX0W.CSIHnCSx.

CSIHnRX0W.CSIHnCSx in the reception register indicates the chip select signal associated with the received data.

CAUTION

When high priority communication function by CPU control is enabled (CSIHnCTL1.CSIHnPHE = 1), IDLE state is inserted regardless of IDLn bit settings when priority communication mode is changed from low to high and from high to low.

16.5.3.2 CS Example

The following figure shows an example of two consecutive data transmissions.

The first communication uses CS0 to communicate with one slave. The second enables CS0 and CS1 to broadcast a message to two slaves. The priority of CS0 is set to “recessive: low priority” and the priority of CS1 to “dominant: high priority”. Consequently, the second communication is conducted using the CS1 settings, which are set as dominant.

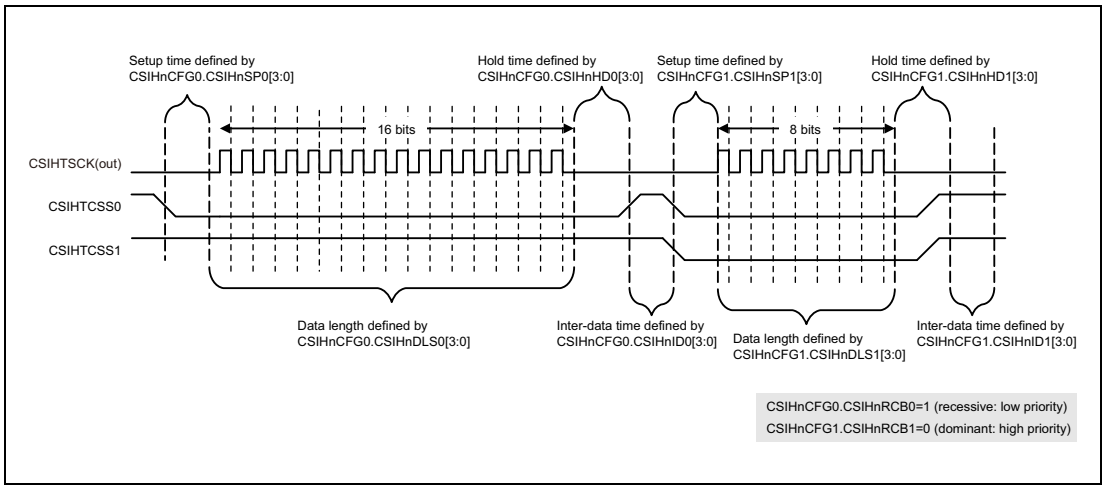


Figure 16.14 Chip Select and RCB Example

16.5.3.3 Job Concept

In CSIH, a job consists of the number of data targeted for transfer.

Job mode enable

The job mode can only be enabled in master mode. The job mode is enabled and disabled by CSIHnCTL1.CSIHnJE, while the CSIH is disabled by CSIHnCTL0.CSIHnPWR = 0.

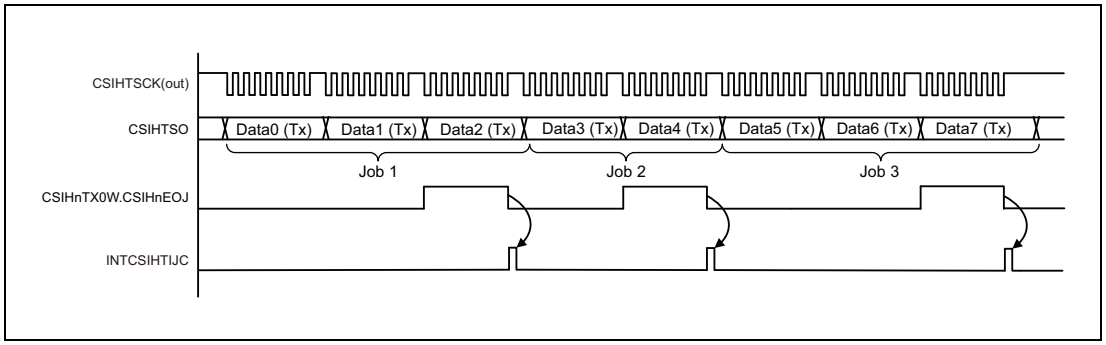


Figure 16.15 Job Examples

A job ends by transmitting data with CSIHnTX0W.CSIHnEOJ = 1.

Communication can be specified to stop when a job is finished. This is done by setting CSIHnCTL0.CSIHnJOBE. When CSIHnJOBE is set, the communication continues until the data for which the CSIHnEOJ bit is set is transmitted. After this data is sent, the communication is stopped and the job completion interrupt INTCSIHTIJC is generated.

16.5.4 Details of Chip Select Timing

16.5.4.1 Changing the Clock Phase

The serial clock level specified by CSIHnCFGx.CSIHnCKPx can be changed while communication is stopped. The minimum value of an idle time is one transmission clock (CSIHTSCK(out)) cycle.

If the idle time is set to 0.5 transmission clock cycles (in CSIHnCFGx.CSIHnIDx[2:0]) and two consecutive data is sent with different CSIHnCFGx.CSIHnCKPx configuration, the idle time is automatically extended to one CSIHTSCK(out) cycle.

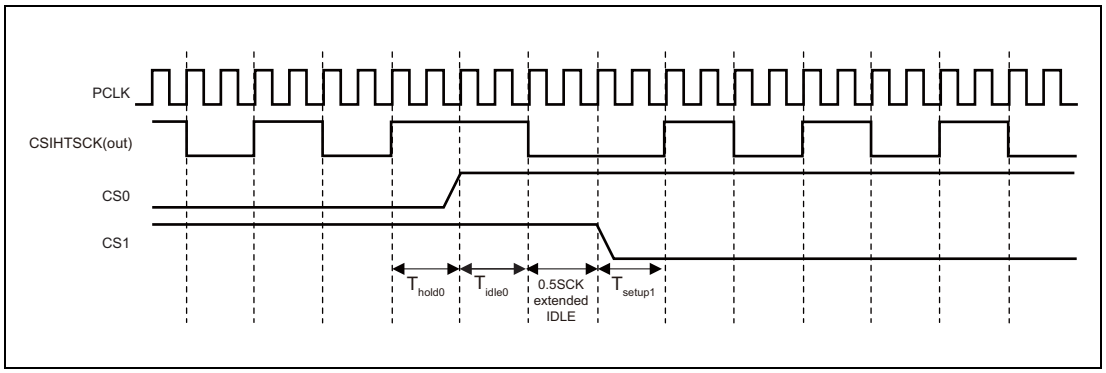


Figure 16.16 Clock Phase Timing with PCLK/4, $T_{hold}0 = T_{setup1}$ = 0.5CSIHTSCK, T_{idle0} = 0.5CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG1.CSIHnCKP1 = 1 (CSIHTCSS1)

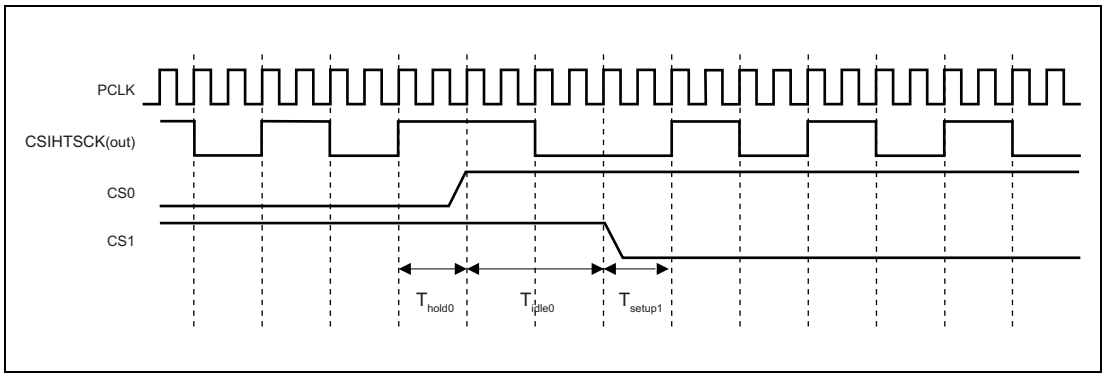


Figure 16.17 Clock Phase Timing with PCLK/4, $T_{hold}0 = T_{setup1}$ = 0.5CSIHTSCK, T_{idle0} = 1CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG1.CSIHnCKP1 = 1 (CSIHTCSS1)

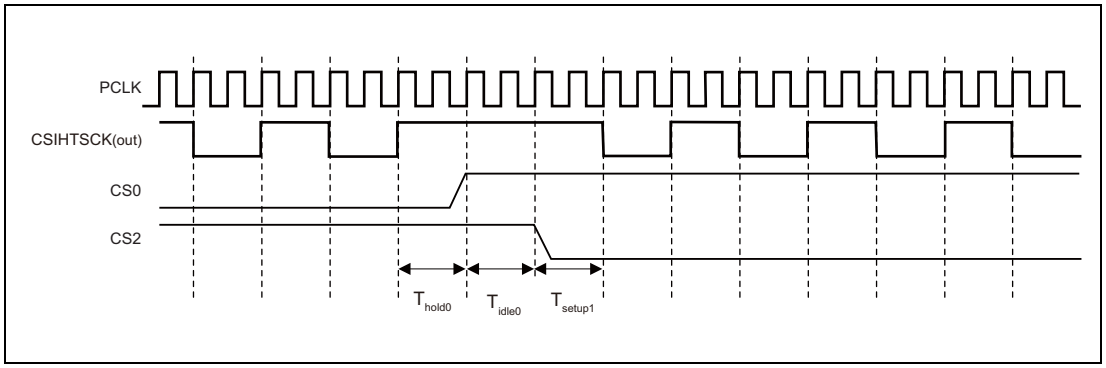


Figure 16.18 Clock Phase Timing with PCLK/4, $T_{hold}0 = T_{setup1}$ = 0.5CSIHTSCK, T_{idle0} = 0.5CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG2.CSIHnCKP2 = 0 (CSIHTCSS2)

16.5.4.2 Changing the Data Phase

The CSIHnCFGx.CSIHnDAPx bit defines the phase of the data bits relative to the clock.

The relation between the setting of the CSIHnCFGx.CSIHnDAPx bit and the hold and setup periods is as follows:

Hold time is the period from the last edge of the serial clock (CSIHTSCK) until the signals on CSIHTCSS[7:0] change to the inactive level.

Setup time is the period from when the signals on CSIHTCSS[7:0] change to the active level to when the transmission data (CSIHTSO) is output.

Therefore, there is a gap of 0.5 CSIHTSCK cycles until the edge of the serial clock signal (CSIHTSCK) is output according to the CSIHnCFGx.CSIHnDAPx setting.

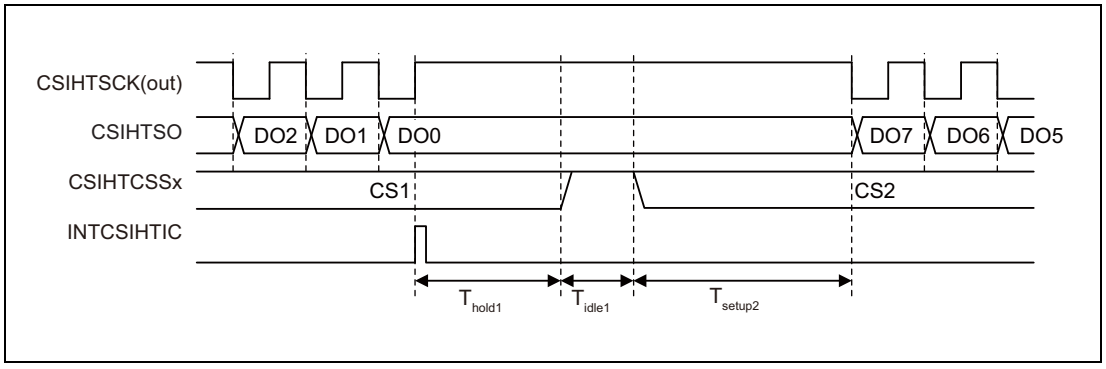


Figure 16.19 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 0

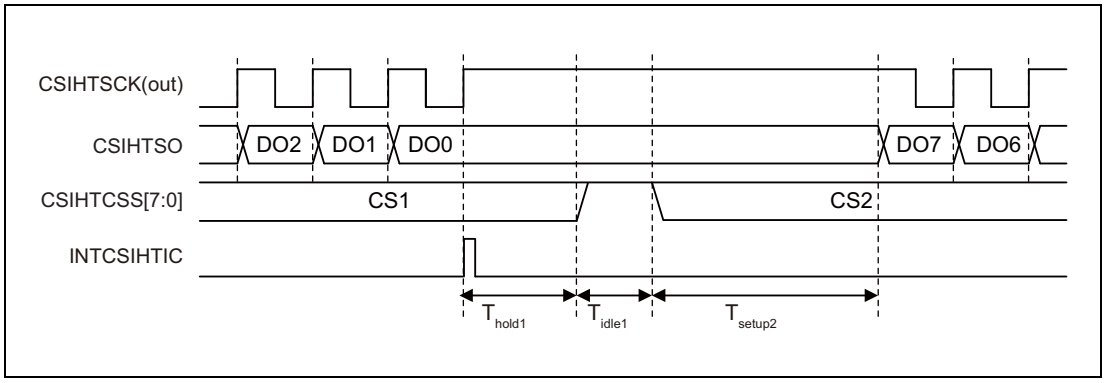


Figure 16.20 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 1, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 1

16.5.5 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the following bits:

- CSIHnCTL2.CSIHnPRS[2:0]
- CSIHnBRSy.CSIHnBRS[11:0]
- CSIHnCFGx.CSIHnBRSSx[1:0]

The transfer clock frequency of transmission clock CSIHTSCK is determined by the setting of the CSIHnCTL2.CSIHnPRS[2:0] bits and the setting of the CSIHnBRSy.CSIHnBRS[11:0] bits, but any one of CSIHnBRS3 to CSIHnBRS0 can be selected for each chip select signal by using the CSIHnCFGx.CSIHnBRSSx[1:0] bits.

The following figure shows a block diagram of the baud rate generator.

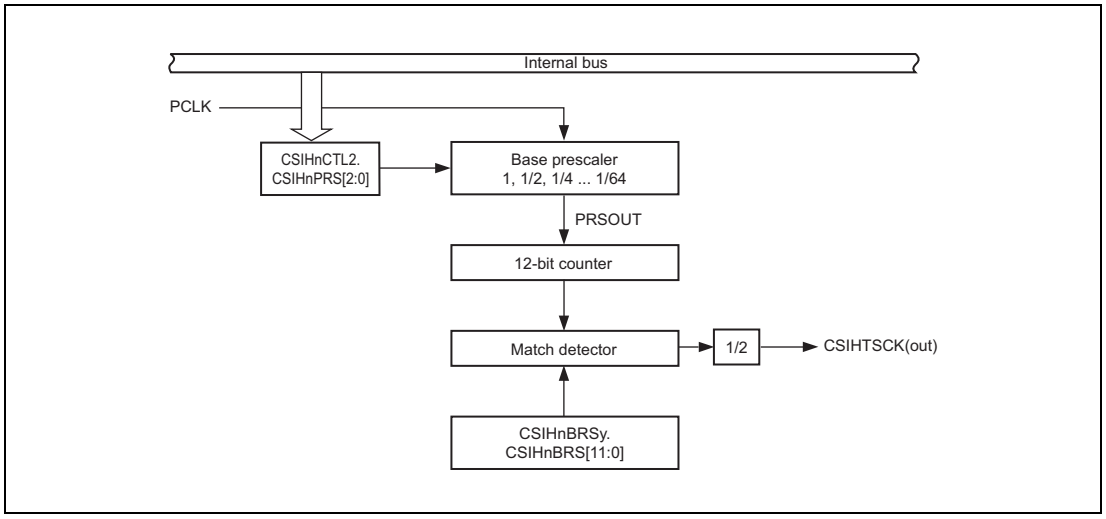


Figure 16.21 Baud Rate Generator Block Diagram

By setting CSIHnBRSy.CSIHnBRS[11:0] to 000_H, the baud rate generator is disabled and all CSIHTSCK are stopped.

Transfer clock frequency calculation

The transfer clock frequency in master mode is calculated as:

Transfer clock frequency (CSIHTSCK) = PCLK / (division ratio of PCLK) = PCLK / ($2^{\alpha} \times k \times 2$),

where:

α = CSIHnCTL2.CSIHnPRS[2:0] = 0 to 6

k = CSIHnBRS0.CSIHnBRS0[11:0] = 1 to 4095
(when CSIHnCFGx.CSIHnBRSSx[1:0] = 0)
CSIHnBRS1.CSIHnBRS1[11:0] = 1 to 4095
(when CSIHnCFGx.CSIHnBRSSx[1:0] = 1)
CSIHnBRS2.CSIHnBRS2[11:0] = 1 to 4095
(when CSIHnCFGx.CSIHnBRSSx[1:0] = 2)
CSIHnBRS3.CSIHnBRS3[11:0] = 1 to 4095
(when CSIHnCFGx.CSIHnBRSSx[1:0] = 3)

Transfer clock frequency upper and lower limits

When setting the transfer clock frequency, please note the following:

- The minimum transfer clock frequency in master mode and slave mode is PCLK/524160.
- The maximum transfer clock frequency is as follows:
  - In master mode: 10.0 MHz (however, it must be equal to or lower than PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be equal to or lower than PCLK/6)

16.5.6 CSIH Buffer Memory

The CSIH has a configurable RAM that can be used for buffered I/O. The size is 128 words. One word is comprised of 32-bit data and 7-bit ECC.

The following configurations are available:

Mode	CSIHnCTL0. CSIHnMBS	CSIHnMCTL0. CSIHnMMS[1:0]
FIFO mode	0	00_B
Dual buffer mode		01_B
Transmit-only buffer mode		10_B
Direct access mode	1	X

16.5.6.1 FIFO Mode

In FIFO mode, data can be written to the CSIHnTX0W register without waiting for completion of the transmission, and data can be received without reading the CSIHnRX0W register immediately, provided the FIFO is not full.

Data to be transmitted is stored in the FIFO memory. Transmission and reception occur simultaneously – one data is received as one data is transmitted. That means, received data overwrites the transmitted data in the FIFO.

The CSIH automatically updates the respective FIFO memory pointers when data is written to or read from the FIFO memory, or data is transmitted to or received from the FIFO memory.

Table 16.39 FIFO Mode

Pointer Description	Control Bit*1	Range
Number of untransmitted words	CSIHnSTR0.CSIHnSPF[7:0]	0 to 128
Number of words received and stored in the FIFO	CSIHnSTR0.CSIHnSRP[7:0]	0 to 128
Address for write/read of transmit data	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $01FC_H$
Address for read of received data	CSIHnMRWP0.CSIHnRRA[6:0]	0000_H to $01FC_H$
Address to be sent	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $01FC_H$

Note 1. The values are automatically updated after each read/write or data transmit/receive operation.

The CSIH status register contains two FIFO status flags:

- CSIHnSTR0.CSIHnFLF: FIFO full
- CSIHnSTR0.CSIHnEMF: FIFO empty

When this mode is started, CSIHnSTCR0.CSIHnPCT bit must be set. By doing this, only CSIHnSTR0.CSIHnEMF is set, not reset, but set.

All FIFO pointers and FIFO flags except CSIHnSTR0.CSIHnEMF are reset and CSIHnSTR0.CSIHnEMF is set.

16.5.6.2 Dual Buffer Mode

In this mode, the memory is divided into two parts of equal size ; 64 words are allocated to transmission data and 64 words to received data. In dual buffer mode, the respective buffer pointers indicate the values shown in the following table.

Table 16.40 Dual Buffer Mode

Pointer Description	Pointer*1	Range
Address of data written to and read from transmit buffer	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $00FC_H$
Address of data read from receive buffer	CSIHnMRWP0.CSIHnRRA[6:0]	0000_H to $00FC_H$
The number of transmit data remaining in the transmit buffer	CSIHnMCTL2.CSIHnND[6:0]	0 to 64
Address to which data is transmitted	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $00FC_H$

Note 1. Pointers are automatically incremented after each read/write.

16.5.6.3 Transmit-Only Buffer Mode

In this mode, the entire memory is used to save transmission data.

Received data must be read directly from CSIHnRX0W/H.

In transmit-only buffer mode, the respective buffer pointer indicates the values shown in the following table.

Table 16.41 Transmit-Only Buffer Mode

Pointer Description	Pointer*1	Range
Address of data written to and read from transmit buffer	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $01FC_H$
The number of transmit data remained in the transmit buffer	CSIHnMCTL2.CSIHnND[6:0]	0 to 128
Address to which data is transmitted	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $01FC_H$

Note 1. Pointers are automatically incremented after each read/write.

16.5.6.4 Direct Access Mode

In direct access mode, the CSIH memory is completely bypassed:

- Transmission data provided by the CPU to the transmission register CSIHnTX0W or CSIHnTX0H is directly copied to the shift register.
- Reception data is directly copied from the shift register to the reception register CSIHnRX0W or CSIHnRX0H.

16.5.7 Data Transfer Modes

16.5.7.1 Transmit-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 0 puts the CSIH in transmit-only mode. Start of transmission depends on the memory mode:

- In case of FIFO or direct access mode, transmission starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, transmission starts when the CSIHnMCTL2.CSIHnBTST bit is set.

16.5.7.2 Receive-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 0 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIH in receive-only mode.

In master mode, start of reception depends on the memory mode:

- In case of FIFO or direct access mode, reception starts when dummy data is written to the CSIHnTX0W or CSIHnTX0H register.

In slave mode, reception starts as soon as the CSIHTSCK transmission clock is received from the master. It is not necessary to write data to the CSIHnTX0W or CSIHnTX0H register of the slave.

- In case of dual buffer mode or transmit-only buffer mode, reception starts when the CSIHnMCTL2.CSIHnBTST bit is set.

16.5.7.3 Transmit/Receive Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIH in transmit/receive mode.

Start of communication (transmission and reception) depends on the memory mode:

- In case of FIFO or direct access mode, communication starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, communication starts when the CSIHnMCTL2.CSIHnBTST bit is set.

16.5.7.4 Summary

The following table summarizes this section. It shows how data transfer is started in the various memory, operating, and transfer modes.

Table 16.42 Start of Data Transfer

Memory and Operating Mode		Transfer Mode	
		Transmit-Only Transmit/Receive	Receive-Only
FIFO, direct access	Master	Writing to the CSIHnTX0W register or the CSIHnTX0H register	Writing to the CSIHnTX0W register or the CSIHnTX0H register
	Slave	Clock reception from master	Incoming clock from master
Transmit-only buffer, dual buffer	Master	CSIHnMCTL2.CSIHnBTST = 1	CSIHnMCTL2.CSIHnBTST = 1
	Slave	Incoming clock from master	Incoming clock from master

16.5.8 Data Length Selection

16.5.8.1 Data Length from 2 to 16 Bits

The length of a data packet is selectable for each chip select signal from 2 to 16 bits using CSIHnCFGx.CSIHnDLSx[3:0]. The examples below show the communication with MSB first (CSIHnCFGx.CSIHnDIRx = 0).

Data length = 16 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 0000_B):

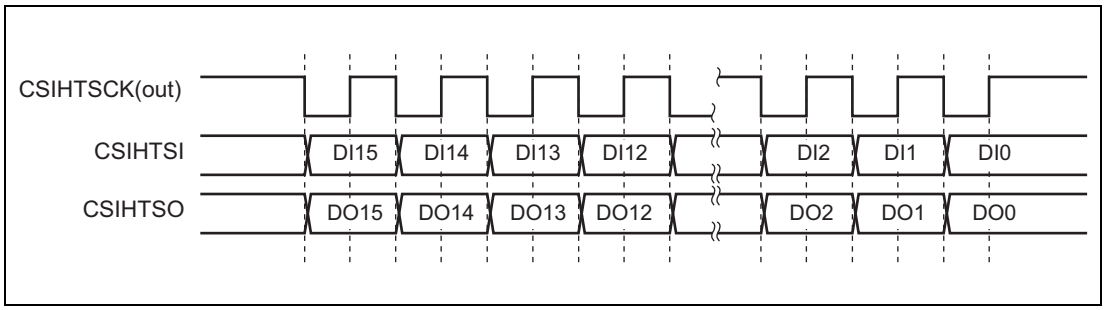


Figure 16.22 16 Bit Data Length, MSB First

Data length = 14 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1110_B):

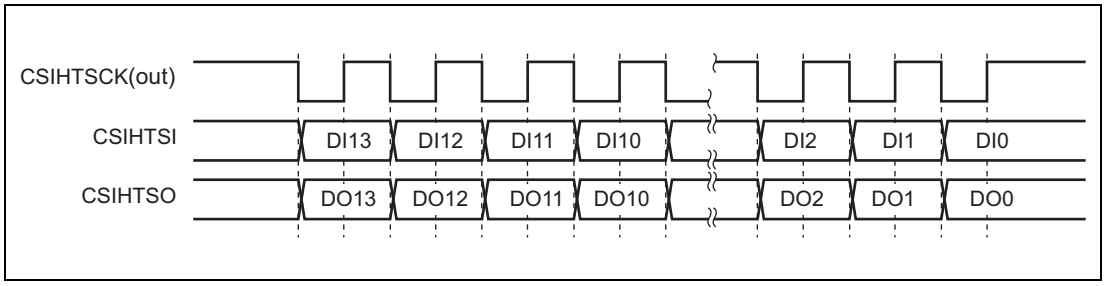


Figure 16.23 14 Bit Data Length, MSB First

16.5.8.2 Data Length Greater than 16 Bits

If the data to be sent/received exceeds 16 bits, the extended data length (EDL) function can be used.

EDL function is enabled by setting the CSIHnCTL1.CSIHnEDLE bit to 1.

EDL function works as follows:

- The data has to be broken into 16-bit blocks and remainder. For example, data of 42 bits would be broken into two 16-bit blocks and 10 bits.
- The bit length of the remainder is set as “data length” in CSIHnCFGx.CSIHnDLSx[3:0].
- For transmitting the 16-bit blocks, CSIHnTX0W.CSIHnEDL must be set to 1. In this case, the data written to CSIHnTX0W is sent as a 16-bit data length regardless of the CSIHnCFGx.CSIHnDLSx[3:0] bit setting.
- The transfer is complete after a block with the specified data length (the remainder of data specified with CSIHnTX0W.CSIHnEDL = 0) has been sent.

Example

Example of transmitting 40-bit data ($123456789A_H$) to CS0:

40 bits are split into two blocks of 16-bit blocks and 8 bits.

- Initialize to CSIHnCFG0.CSIHnDLS0[3:0] = 8.
- To transmit $123456789A_H$ with MSB first, write the following sequence to CSIHnTX0W:
  - $20FE\ 1234_H$ (CSIHnTX0W.CSIHnEDL = 1)
  - $20FE\ 5678_H$ (CSIHnTX0W.CSIHnEDL = 1)
  - $00FE\ 009A_H$ (CSIHnTX0W.CSIHnEDL = 0)

The following figure illustrates the timing.

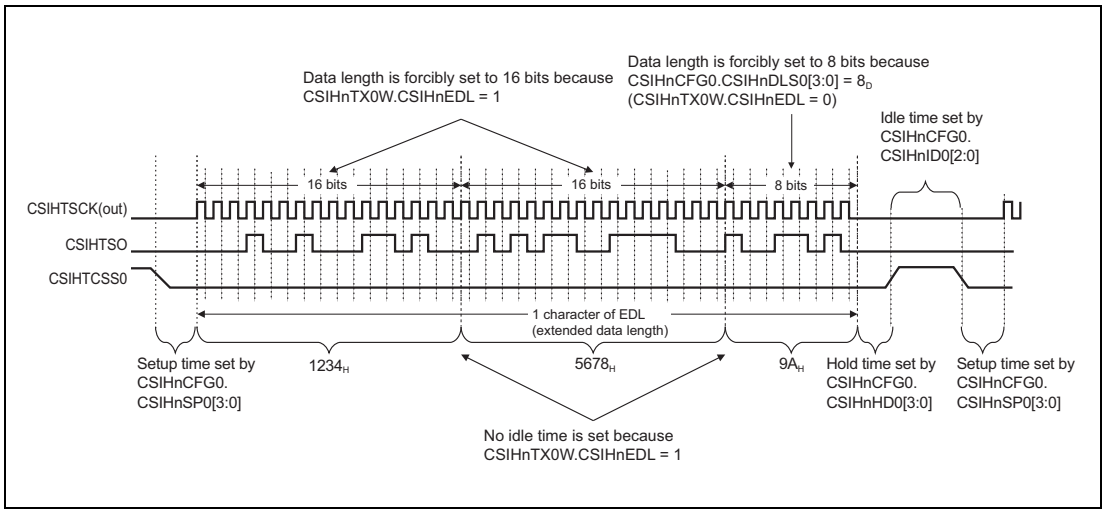


Figure 16.24 EDL Timing Diagram

NOTES

1. 1-bit data length is allowed only when using the EDL mode.
2. If parity is enabled, the parity bit is added after the last bit.
3. When data is sent using extended data length (EDL) function, use the same chip select signal.
4. Example for configuring the data direction:
   - Data to be sent:123456_H
   - MSB first:
     Set CSIHnCFGx.CSIHnDIRx = 0
     Write CSIHnTX0W = 20FE 1234_H (EDL bit = 1)
     Write CSIHnTX0W = 00FE 0056_H (EDL bit = 0)
   - LSB first:
     Set CSIHnCFGx.CSIHnDIRx = 1
     Write CSIHnTX0W = 20FE 3456_H (EDL bit = 1)
     Write CSIHnTX0W = 00FE 0012_H (EDL bit = 0)
5. Operation is not guaranteed if CSIHnTX0W.CSIHnEOJ and CSIHnTX0W.CSIHnEDL are simultaneously set to “1” while CSIHnCTL1.CSIHnJE = 1 and CSIHnCTL1.CSIHnEDLE = 1.
6. EDL mode cannot be used in receive-only mode of slave mode. (CSIHnCTL2.CSIHnPRS[2:0] = 111_B, CSIHnCTL0.CSIHnTXE = 0, CSIHnCTL0.CSIHnRXE = 1)

16.5.9 Serial Data Direction Selection

The serial data direction is selectable for each chip select signal using the CSIHnDIRx bit in the CSIHnCFGx register.

The examples below show communication for a data length of 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).

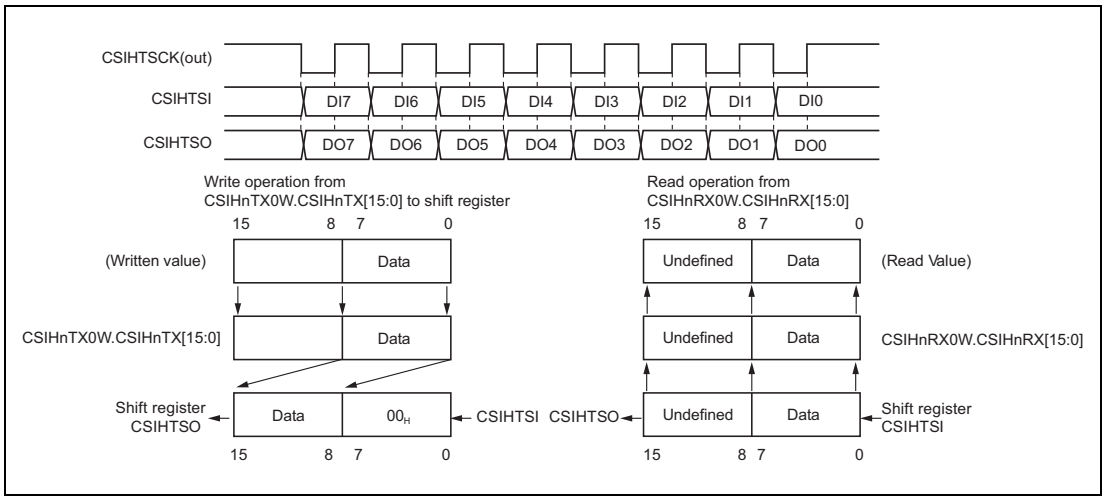


Figure 16.25 Serial Data Direction Select Function - MSB First (CSIHnDIRx = 0)

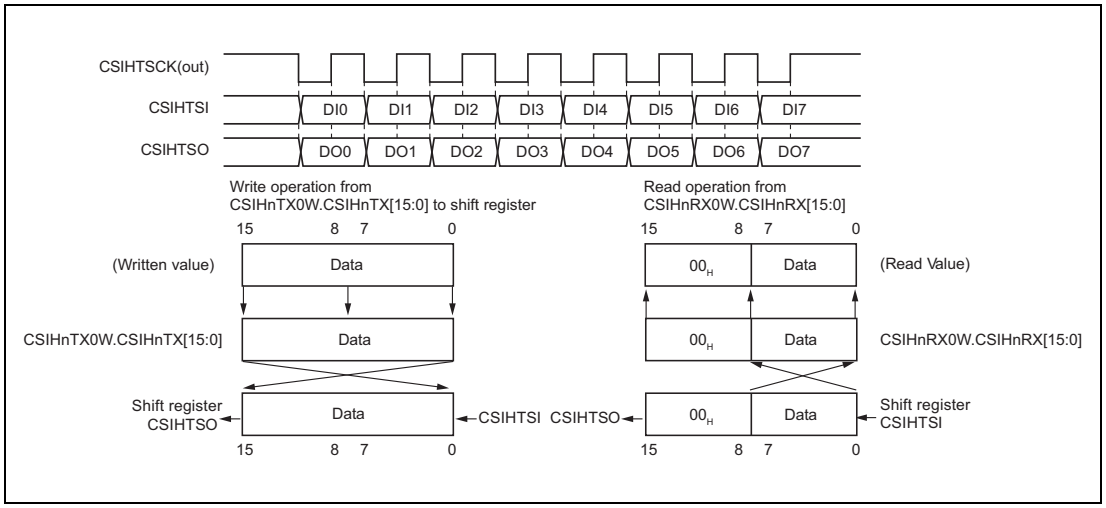


Figure 16.26 Serial Data Direction Select Function - LSB First (CSIHnDIRx = 1)

16.5.10 Slave Select (SS) Function

The Slave Select (SS) function enables communication between one master and multiple slaves.

In master mode, the master device outputs the slave select signal (CSIHTCSSx) to a slave. Communication by a device in slave mode is enabled when the slave input select signal (CSIHTSSI) is at the low level.

See the **Section 16.5.2, Master/Slave Connections**, for examples of connections using the SS function.

16.5.10.1 Communication Timing Using SS Function

The following figure illustrates the communication signal and timings using the SS function.

In slave mode, the data transfer configuration is determined by the CSIHnCFG0 register.

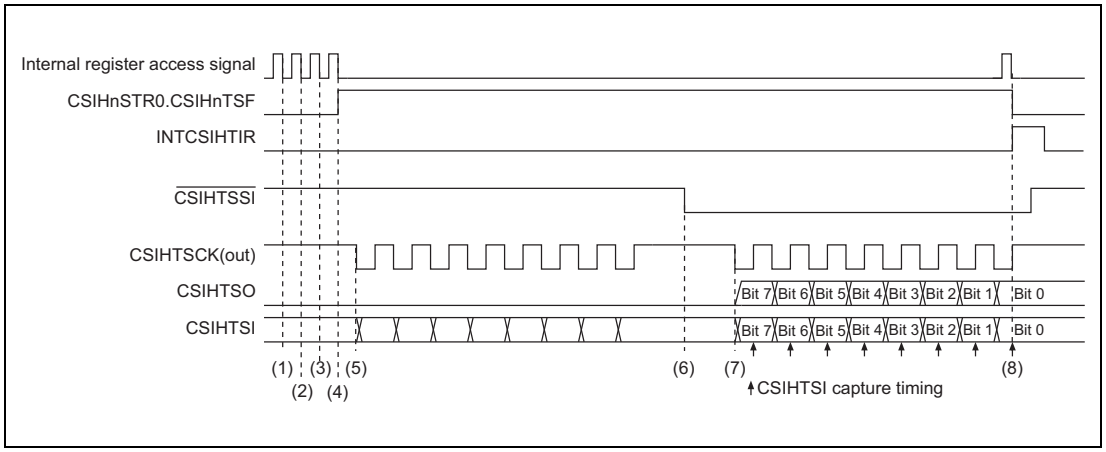


Figure 16.27 Transmission/Reception Timing of Communication Using SS Function

(1) CSIH enters slave mode by setting CSIHnCTL2.CSIHnPRS[2:0] = 111_B. CSIHnCFG0.CSIHnCKP0 and CSIHnCFG0.CSIHnDAP0 are 0.

(2) The data length is 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000_B). The data direction is MSB first (CSIHnCFG0.CSIHnDIR0 = 0).

(3) The transmit/receive mode is set (CSIHnCTL0.CSIHnTXE = 1, CSIHnCTL0.CSIHnRXE = 1, and CSIHnCTL0.CSIHnPWR = 1). Communication start is permitted.

(4) The transfer status flag CSIHnSTR0.CSIHnTSF is automatically set when transfer data is written to the CSIHnTX0W or CSIHnTX0H transmission register during direct access mode or FIFO mode.

(5) As long as signal $\overline{\text{CSIHTSSI}}$ is at the high level, transmission/reception is not started, even if an external transmission clock CSIHTSCK is input. Input to CSIHTSI is ignored.

(6) $\overline{\text{CSIHTSSI}}$ falling to low level indicates that CSIHTSO is enabled and ready for transmission.

(7) As soon as the external clock signal CSIHTSCK is detected, the slave transmits data to CSIHTSO and simultaneously captures data from CSIHTSI.

(8) Interrupt INTCSIHTIR indicates that the reception is complete. The CSIHnRX0W/H register can be read.

16.5.10.2 CSIHTSSO Operation

CSIHnPWR	CSIHnTXE	CSIHnRXE	CSIHnSSE	CSIHTSSO
0	—	—	—	H
1	—	—	0	H
	0		1	H
	1		1	Reversed value of $\overline{\text{CSIHTSSI}}$ level

The CSIHTSSO pin is a signal to control the I/O function of the chip's SO pin when using the SS function.

The CSIHTSO pin is enabled when the CSIHTSSO pin is "High" (the chip's SO pin is being driven).

The CSIHTSO pin is disabled when the CSIHTSSO pin is "Low" (the chip's SO pin is not being driven).

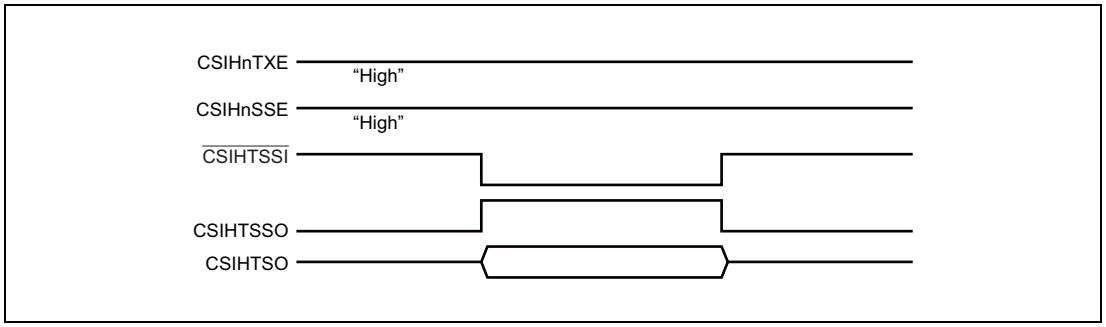


Figure 16.28 Operation of CSIHTSSO

CAUTION

If $\overline{\text{CSIHTSSI}}$ pin is changed during communication (CSIHnSTR0.CSIHnTSF = 1), current communication is not guaranteed.

16.5.11 Handshake Function

CSIH features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by the CSIHnCTL1.CSIHnHSE bit. For handshake, the signals CSIHTRYI and CSIHTRYO are used.

The busy timing depends on the data phase selection bit CSIHnCFGx.CSIHnDAPx.

16.5.11.1 Slave Mode

If CSIHnCTL1.CSIHnHSE = 1, a low-level CSIHTRYO signal is output when the slave becomes busy. This can happen in two cases:

1. When the next data to be sent is not ready:
   When the slave is in transmit-only mode or transmit/receive mode (CSIHnCTL0.CSIHnTXE = 1) and is in the states listed below, the CSIHTRYO outputs the busy state (low level).

Table 16.43 Memory Mode and Slave Transfer State

Memory Mode	Slave Transfer State
Direct access mode	When there is no more data to be sent
FIFO mode	When there is no more data to be sent (CSIHnSTR0.CSIHnEMF = 1)
Dual buffer mode	When CSIHnMCTL2.CSIHnBTST is not set to 1
Transmit-only buffer mode	

following examples assume an eight-bit data length.

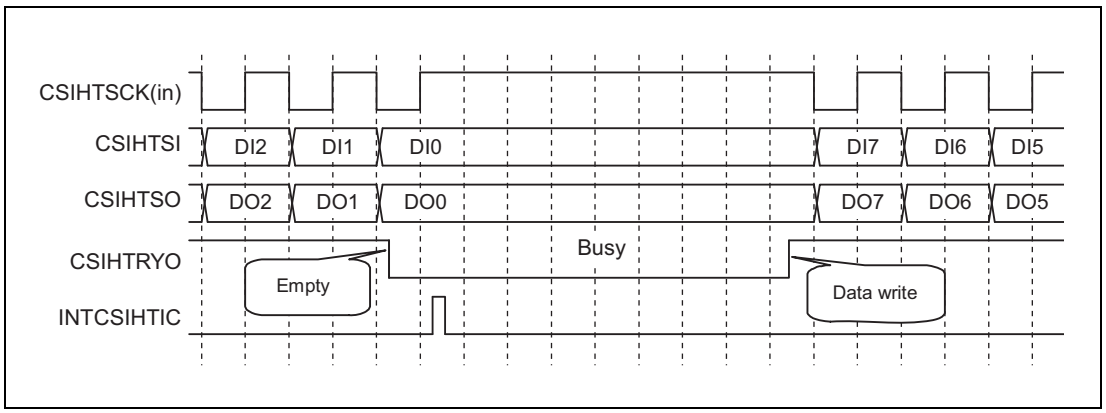


Figure 16.29 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 0)

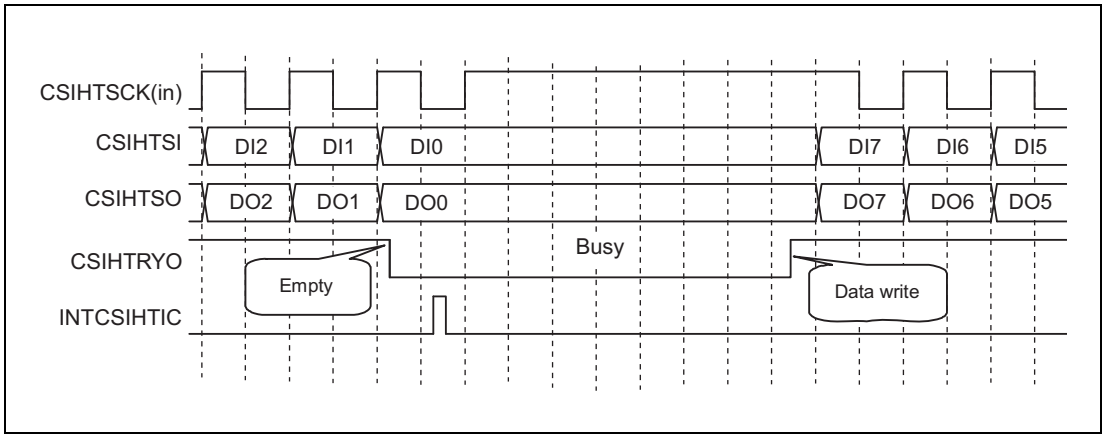


Figure 16.30 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 1)

2. When receive register is full:
   When slave is set in receive-only mode or transmit/receive mode (CSIHnCTL0.CSIHnRXE = 1), and new data cannot be copied from a shift register to CSIHnRX0W/H because the previously received data is still in the CSIHnRX0W/H register (CSIHnRX0W/H is full).
   When CSIHnCTL0.CSIHnRXE is 1 and is in any of the following states, CSIHTRYO outputs busy state (low level).

Table 16.44 Memory Mode and Slave Reception State

Memory Mode	Slave Reception State
Direct access mode	When CSIHnRX0W or CSIHnRX0H is full
FIFO mode	When received data is remaining in buffer (CSIHnSTR0.CSIHnFLF = 1)
Dual buffer mode	No applicable case
Transmit-only buffer mode	When CSIHnRX0W or CSIHnRX0H is full

The following examples assume an eight-bit data length.

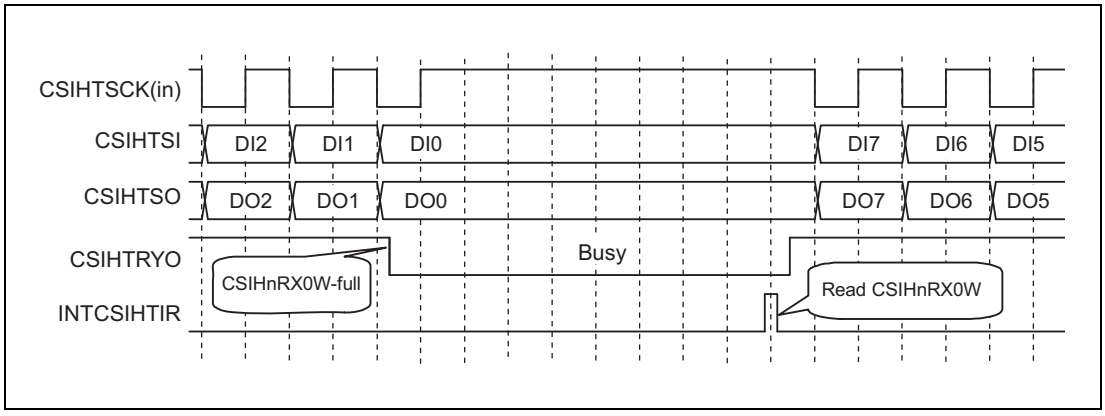


Figure 16.31 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 0)

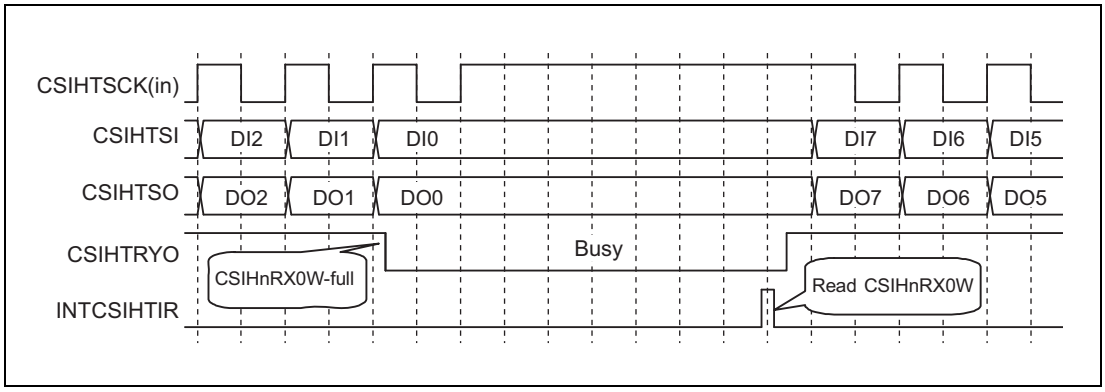


Figure 16.32 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 1)

16.5.11.2 Master Mode

When the master detects CSIHTRYI = 0 while CSIHnCTL1.CSIHnHSE = 1, the subsequent transfers are put on hold, the master goes into wait state and suspends the CSIHTSCK clock.

The CSIHTRYI level is checked at each half clock cycle of CSIHTSCK.

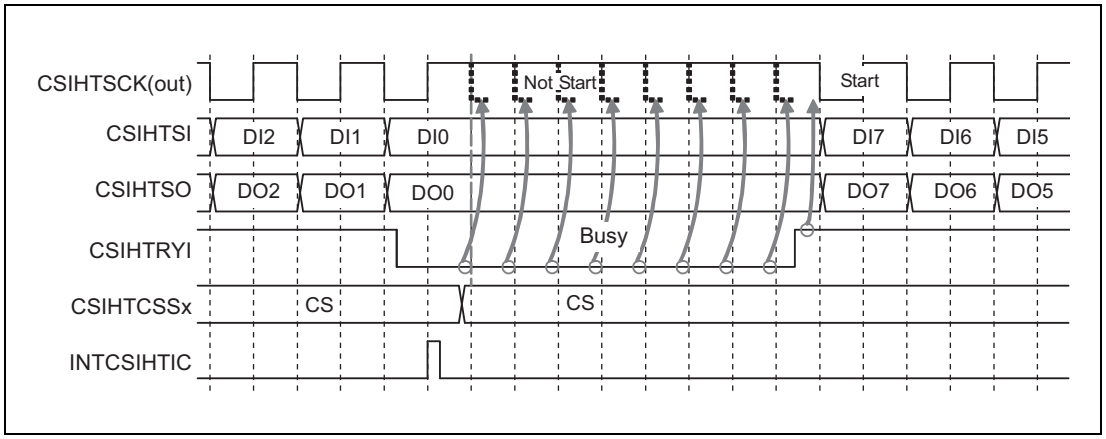


Figure 16.33 Master's Reaction to CSIHTRYI (CSIHnCFGx.CSIHnDAPx = 0)

The CSIHTRYI signal must be pulled down by the slave before the next transfer starts. If this is done while data transfer is in progress, the serial clock from the master is suspended after the transfer is complete.

The master resumes the communication as soon as CSIHTRYI becomes high (the slave is "ready").

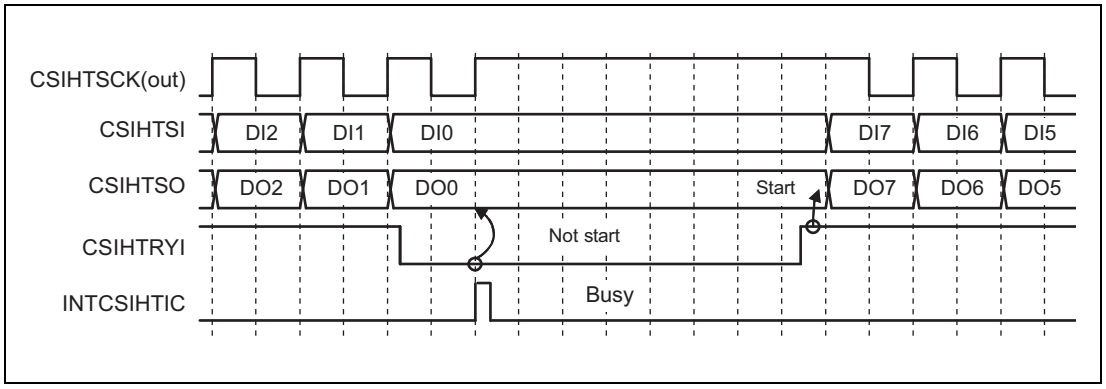


Figure 16.34 Master's Reaction to CSIHTRYI (CSIHnCFGx.CSIHnDAPx = 1)

CAUTIONS

1. **If multiple slaves are connected, the master must only detect the CSIHTRYI signal of the slave it has selected for communication.**
2. **Even when the CSIHTRYI pin of the master detects a CSIHTRYO signal from the slave during data transfer, the communication is not put on hold but continues until the data transfer is completed.**

16.5.12 Error Detection

CSIH can detect five error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error (received data)
- Time-out error (in FIFO mode)
- Overflow error (in FIFO mode)

Check for parity, data consistency and time-out errors can be enabled/disabled individually.

If any of these errors is detected, the interrupt request INTCSIHTIRE is generated and the corresponding flags are set.

16.5.12.1 Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent as output signal is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by the CSIHnCTL1.CSIHnDCS bit (when checking data consistency, make sure that PIPCn.PIPCn_m = 1). It will not be enabled if data transmission is disabled (CSIHnCTL0.CSIHnTXE = 0).

When the data consistency check is active, the data transferred from CSIHnTX0W or CSIHnTX0H to the shift register is copied to a separate register. In addition, the physical levels of CSIHTSO are read back via the CSIHTDCS signal into the corresponding shift register.

After completion of the transmission, the sent data is compared with the original transmission data.

Mismatch is considered as a data consistency error and:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnDCE bit is set.

Additionally, CSIHnRX0W.CSIHnTDCE of data that contains the error is set.

The data consistency check function is illustrated in the following block diagram.

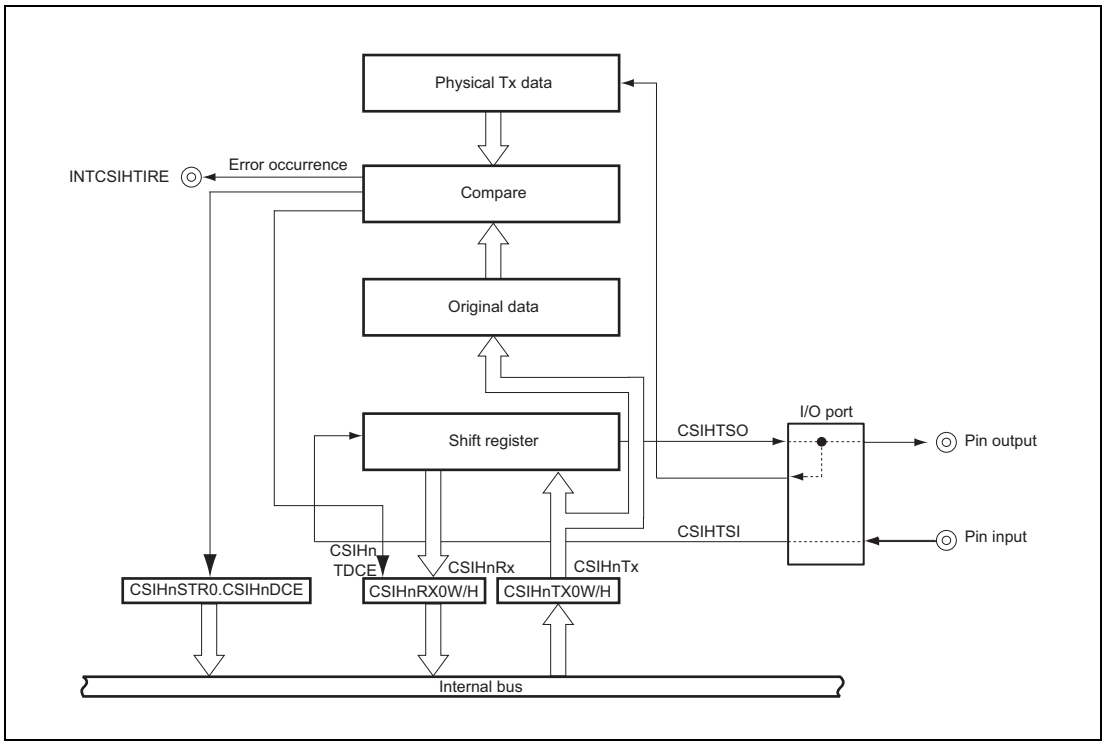


Figure 16.35 Block Diagram of Data Consistency Check Function

16.5.12.2 Parity Check

CSIH can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified in CSIHnCFGx.CSIHnPSx[1:0].

Parity check is enabled if CSIHnCFGx.CSIHnPSx[1] = 1.

The parity bit is checked after a reception is complete. If an parity error occurs, the following happen:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnPE bit is set.

Additionally, CSIHnRX0W.CSIHnRPE of data that contains the error is set.

The figure below shows an example.

- Data length is 8 bits.
- The data to be transmitted is 05_H and 35_H.
- Data direction is LSB first.
- Parity type is odd.

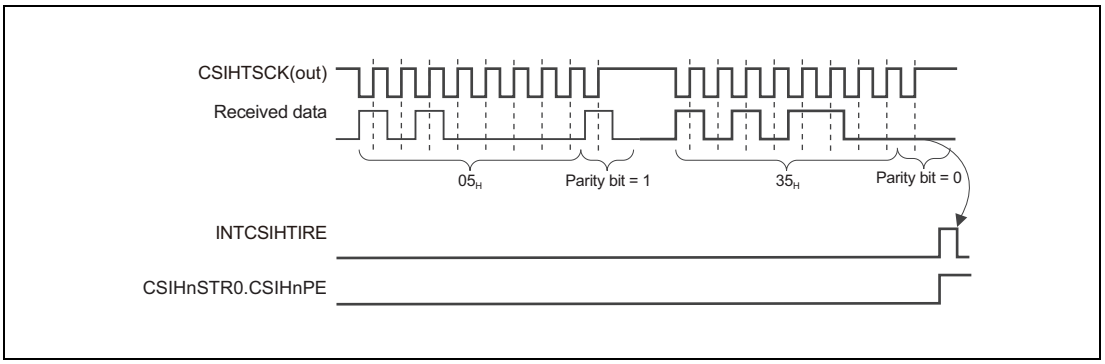


Figure 16.36 Parity Check Example

The parity bit of the first data is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

The parity bit of the second data is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If the EDL (extended data length) function is used, the parity bit is added after the last bit of the data.

16.5.12.3 Time-Out Error

Time-out errors can be checked only in FIFO mode of the slave.

This error occurs if neither of the following occurred within a certain period of time:

- Received data in FIFO is read
- FIFO receives data from CSIHTSI

The time-out time is defined in CSIHnMCTL0.CSIHnTO[4:0] in units of "8 x transmission clock CSIHTSCK". A time-out error occurs when the specified time is exceeded (the time-out time is not detected when CSIHnMCTL0.CSIHnTO[4:0] = 00000_B).

The dedicated time-out counter is set by the CSIHnCTL2.CSIHnPRS[2:0] and CSIHnBRSy.CSIHnBRS[11:0] bits.

If the value of the CSIHnBRSy.CSIHnBRS[11:0] bits is left as 000_H, the dedicated time-out counter does not operate.

The dedicated time-out counter measures the time between the last and the next read operation.

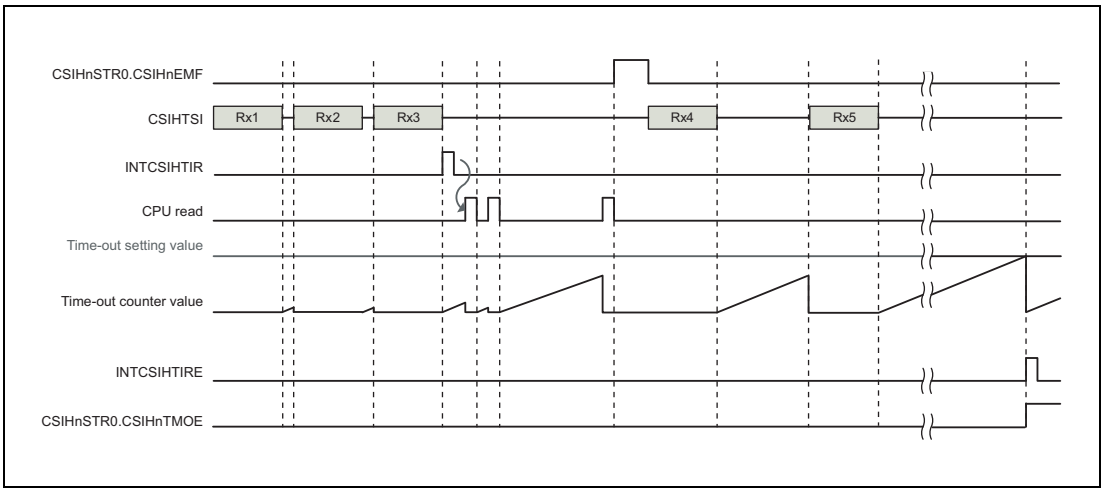


Figure 16.37 Time-Out Check Functional Timing Diagram

The start timing of the time-out counter is as follows:

- When reception is completed
- When data read from the CPU is completed
  (The counter does not start if the buffer is empty.)
- When a time-out error is detected

After a time-out error is detected, if data is still available in FIFO, the time-out counter restarts.

If the value set by bits CSIHnMCTL0.CSIHnTO[4:0] is reached again, the INTCSIHTIRE interrupt is output again.

The timeout counter continues to count until received data is read. To stop the counter, read all received data or set CSIHnSTCR0.CSIHnPCT to 1. Note that the pointer is cleared if you perform it.

The counter is reset at the following timing:

- Data is read once.
- A new data item is received.

- A timeout error is detected.
- The CSIHnSTCR0.CSIHnPCT bit is set to 1.

If a timeout error occurs, the following occur:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnTMOE bit is set.

16.5.12.4 Overflow Error

An overflow error may occur in FIFO mode. It occurs when transmission data is written to the CSIHnTX0W register while the FIFO buffer is filled with received data.

Example

100 data have been transmitted. That means, the FIFO contains 100 received data. The application starts to read the received data.

While the read operation is in progress, the application begins to write another set of 50 transmission data to the FIFO. However, only 10 received data have been read up to now, and 90 are still in the FIFO.

In this case, only 38 buffers are available for new transmission data. When the CPU tries to write the 39th data, an overflow error occurs.

This is illustrated in the following figure.

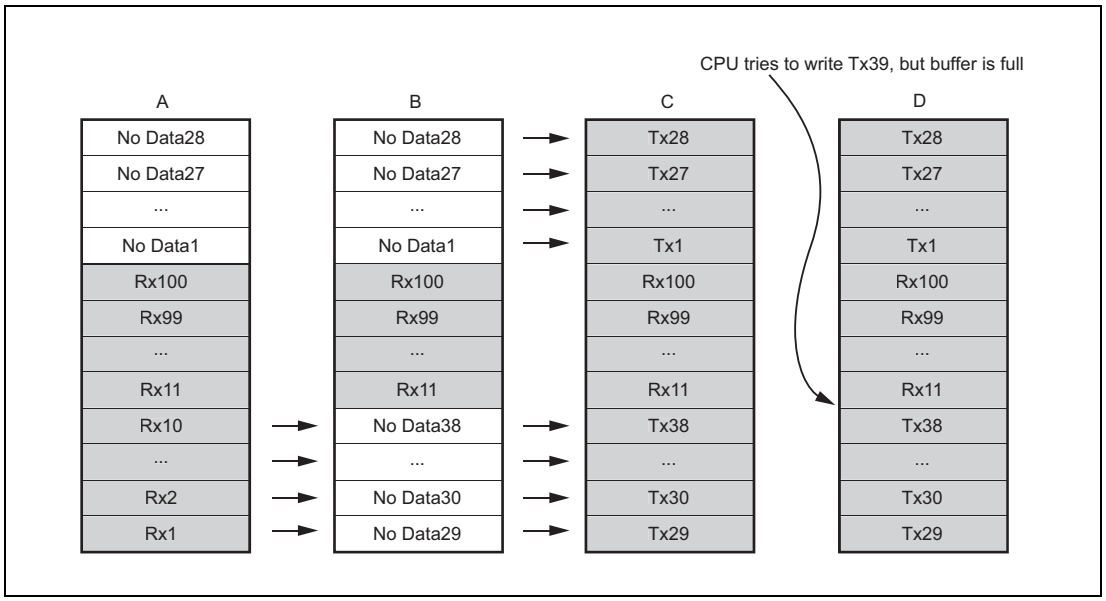


Figure 16.38 FIFO Overview

The 39th and subsequent data are discarded. The figure below shows the overflow timing.

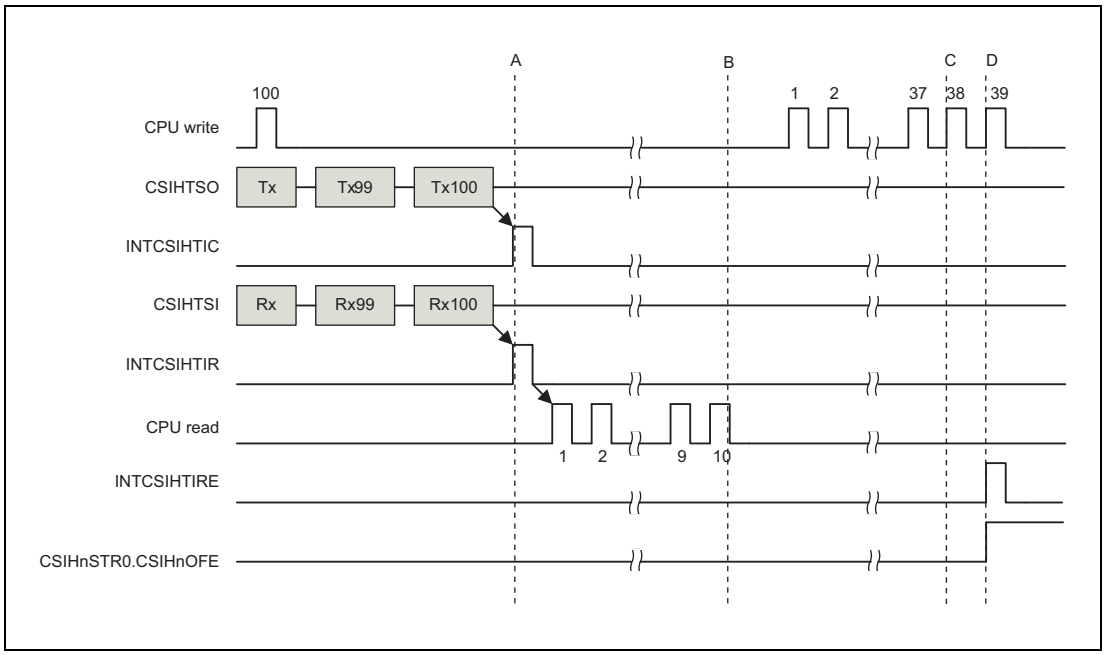


Figure 16.39 FIFO Overflow Timing

In case of overflow error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnOFE bit is set.

16.5.12.5 Overrun Error

An overrun error may occur in direct access, transmit-only buffer, and FIFO modes. It does not occur in dual buffer mode. The overrun error does not occur if data reception is disabled (CSIHnCTL0.CSIHnRXE = 0).

There are two conditions for overrun errors.

Condition for errors 1

- In FIFO mode, if the CPU reads the CSIHnRX0W/H register when the number of received data is 0.

Condition for errors 2

- In slave mode, when CSIHnCTL1.CSIHnHSE = 0 (handshake function disabled):
  - In direct access mode or transmit-only buffer mode, when reception is completed while the previously received data remains in the CSIHnRX0W/H register.
  - In FIFO mode, when reception is completed while the FIFO buffer is still full of receive data.

(1) Direct access/transmit-only buffer

In direct access and transmit-only buffer modes, this error occurs when newly received data cannot be transferred from the shift register to the reception register CSIHnRX0W/H. This happens when CSIHnRX0W/H was not read and therefore contains previously received data.

The following figure illustrates the overrun error detection function.

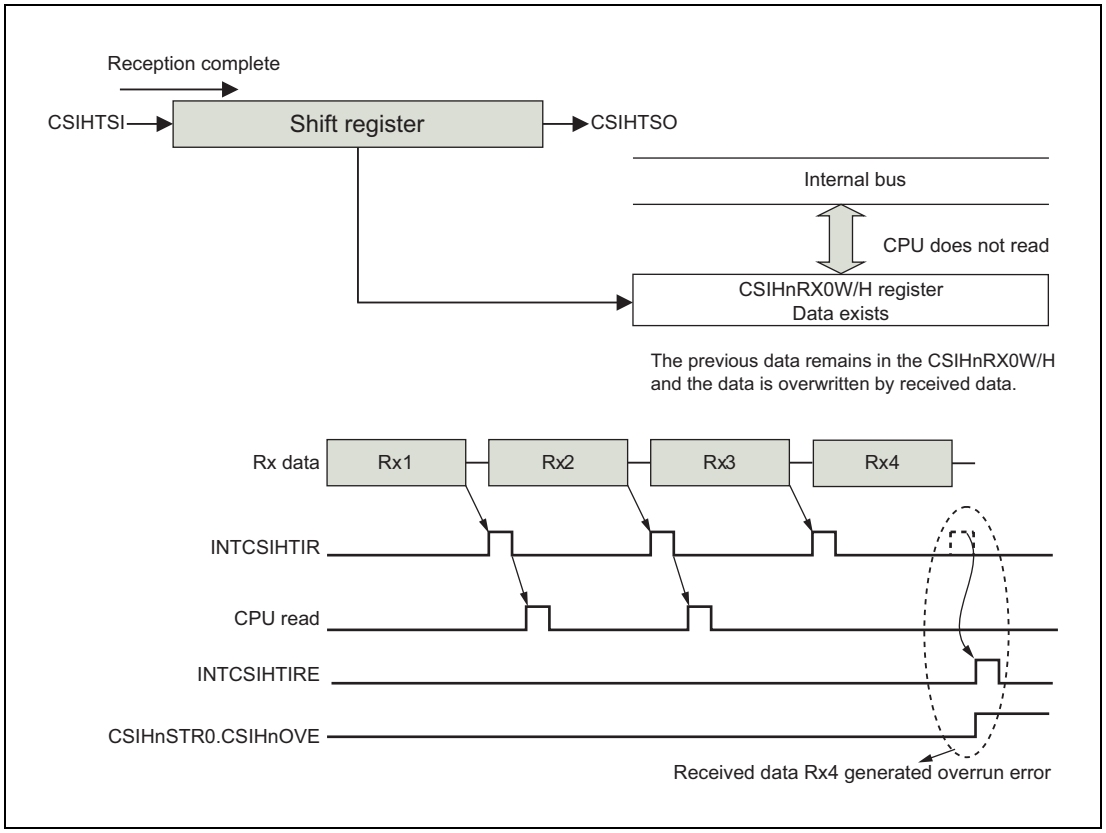


Figure 16.40 Overrun Error Detection in Direct Access and Transmit-Only Buffer Modes

NOTE

An overrun error can be avoided in slave mode by using handshake function. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

(2) FIFO mode

In FIFO mode, this error occurs if:

1. Newly received data cannot be transferred from the shift register to the FIFO because the FIFO is full.

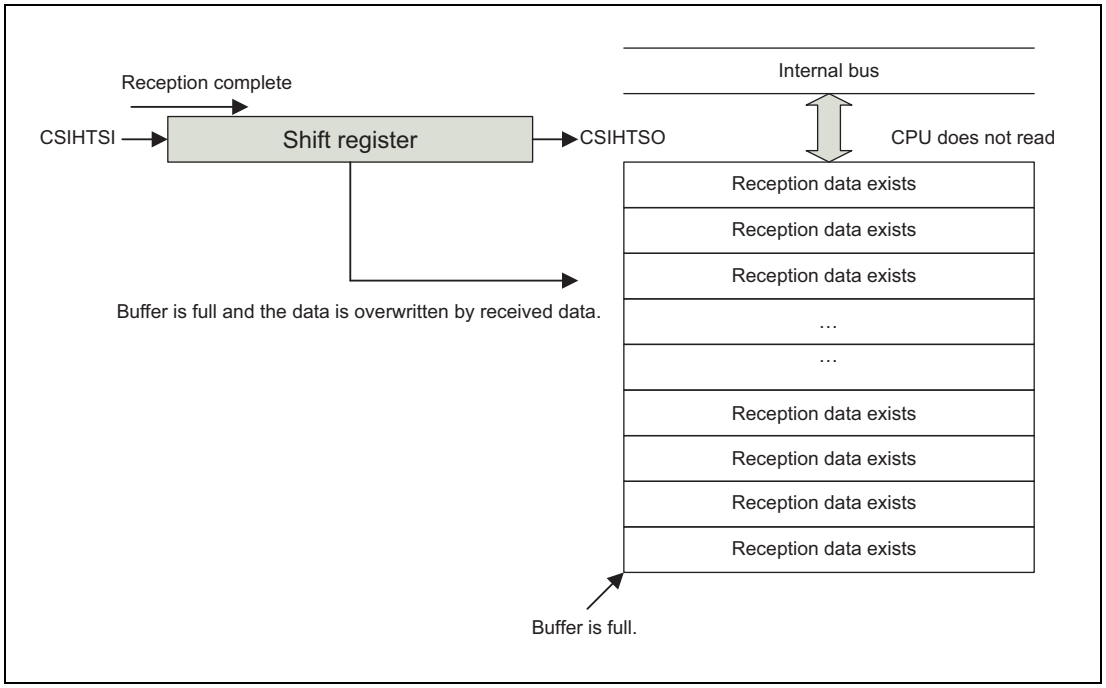


Figure 16.41 Overrun Error Detection in FIFO Mode (FIFO Full)

NOTE

An overrun error can be avoided in slave mode by using the handshake function. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver reads its reception register and becomes ready again.

2. The CPU attempts to read non-existent receive data.

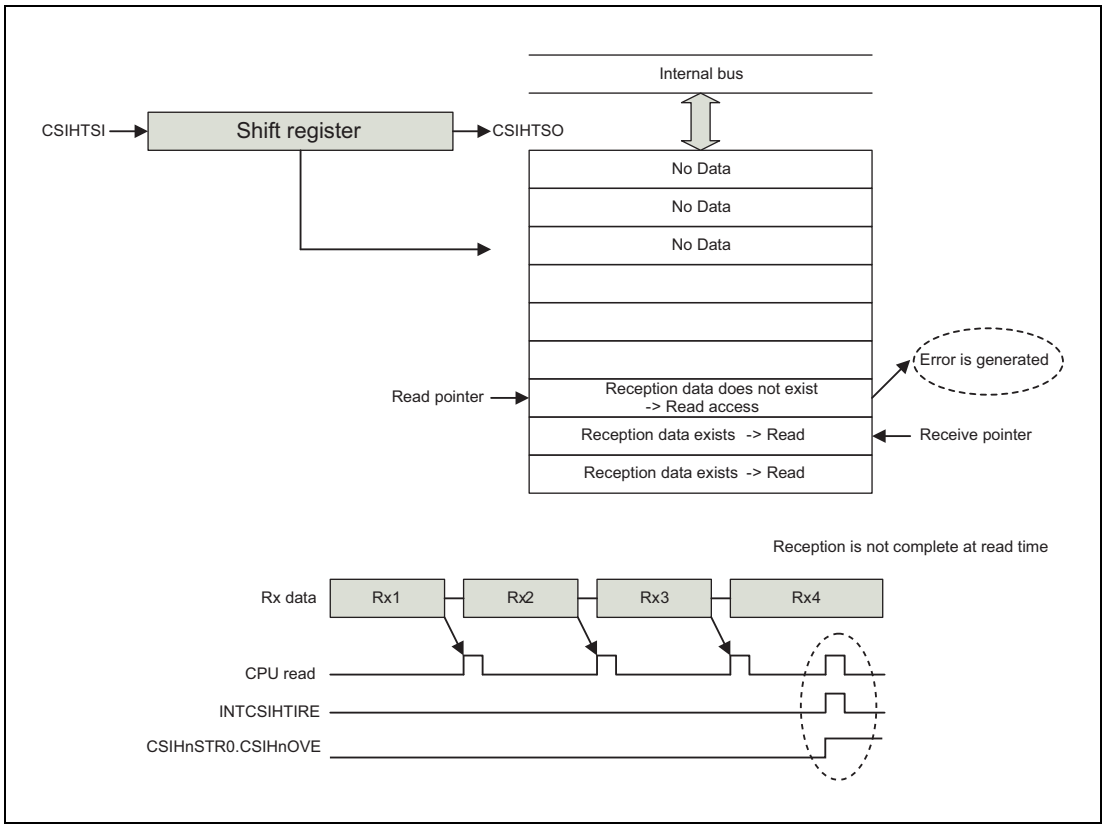


Figure 16.42 Overrun Error Detection in FIFO Mode (No Data)

In case of overrun error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnOVE bit is set.
- Received data is overwritten and the communication continues.
  (When the CPU tries to read non-existent data, it waits until reception is completed and then resumes reading.)

For details, see **Section 16.5.11, Handshake Function**.

16.5.13 Loop-Back Mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active (CSIHnCTL1.CSIHnLBM = 1), CSIHTCSSx is fixed to the inactive level (the active level is defined by the CSIHnCTL1.CSIHnCSLx value). The transmit and receive signals are internally connected, as shown in the figures below. The signals CSIHTSCK, CSIHTSO, CSIHTSI, and CSIHTCSSx are disconnected from the ports. In addition, the CSIHTSO output level is fixed to low, and CSIHTSCK is set to reset level (High) regardless of the value of the CSIHnCFGx.CSIHnCKPx. The rest of CSIH works as in normal operation.

In order to test CSIH, put it in loop-back mode and carry out normal transfer operations. Then check that the received data is the same as the transmitted data. Any connected device remains unaffected by the loop-back test.

Table 16.45 Pin Output Level in Loop-Back Mode

Pin Name	Output level
CSIHTSCK(out)	High level
CSIHTCSS[7:0]	Inactive level
CSIHTSO	Low level (not dependent on the previous value)
Interrupt	Normal function
CSIHTRYO	Normal function (Low level)

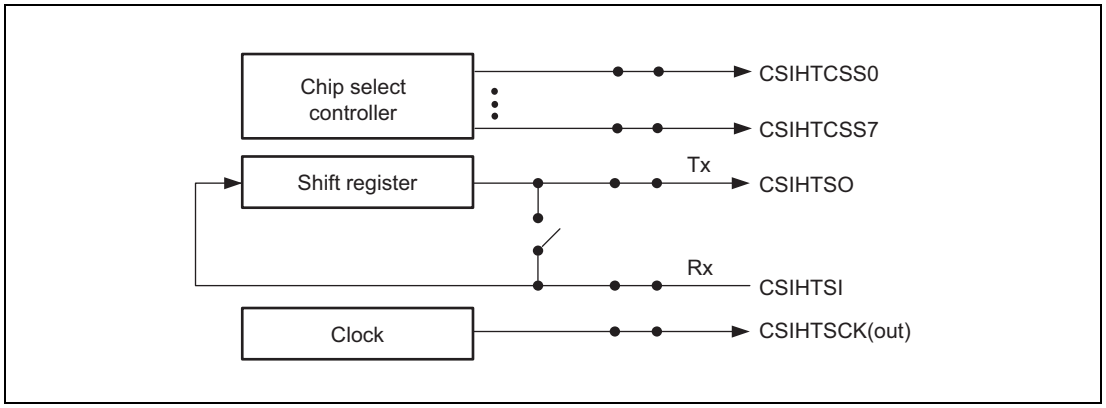


Figure 16.43 Normal Operation

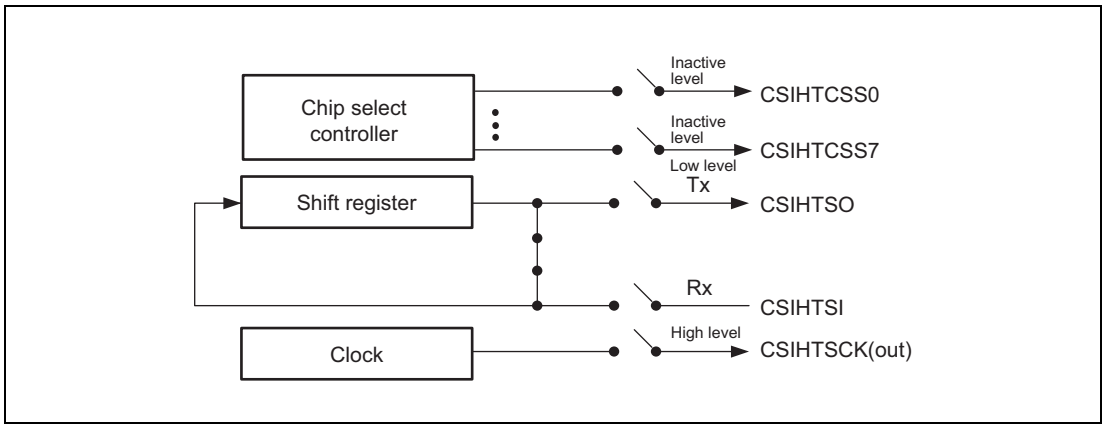


Figure 16.44 Loop-Back Mode Operation

16.5.14 CPU-Controlled High Priority Communication Function

CSIH has a function to abort low priority communication to perform high priority communication if it receives a high-priority communication request from the CPU while low-priority communication is being used. This function supports transmit-only buffer mode as low priority communication and only direct access mode as high-priority communication. To enable this function, CSIHnCTL1.CSIHnPHE and CSIHnCTL1.CSIHnJE must be set to 1.

The following figure illustrates CPU-controlled high-priority communication.

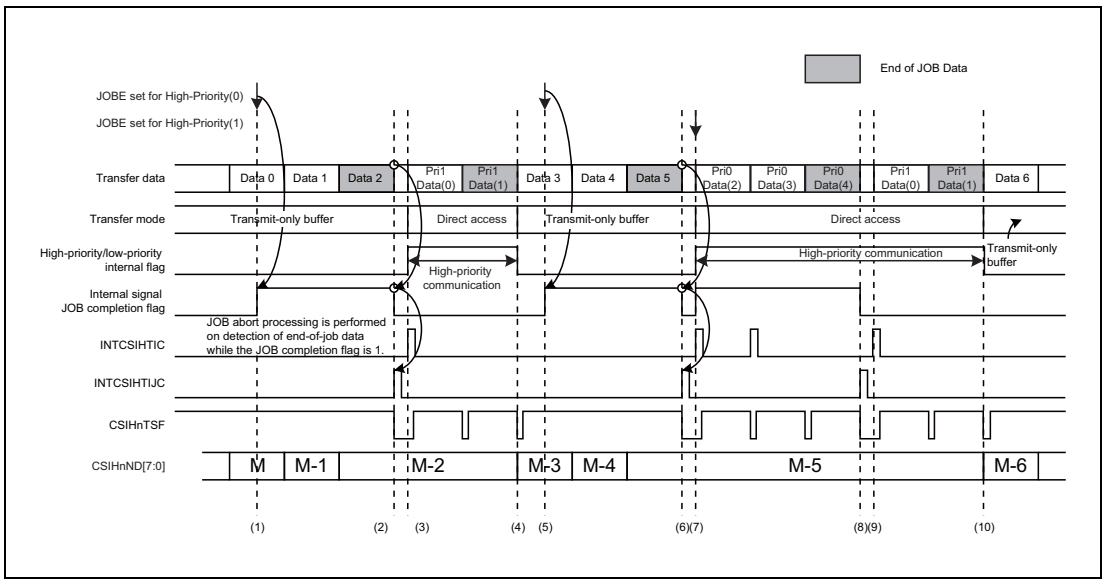


Figure 16.45 Example of CPU-Controlled High-Priority Communication

(1) By setting CSIHnCTL0.CSIHnJOBE = 1 during low-priority communication, start of high-priority communication following end-of-job data is notified, and the internal signal flag is set.

(2) When end-of-job data is detected, the current low-priority communication is aborted and the INTCSIHTIJC interrupt occurs. The internal signal job completion flag is cleared due to the communication abort, and memory mode is automatically switched to direct access mode for the subsequent high-priority communication.

(3) The CPU detects the interrupt and starts communication by writing the first transmission data of high-priority communication to CSIHnTX0W or CSIHnTX0H.

(4) When end-of-job data is detected, communication is aborted. At this time, because the internal signal end-of-job flag is set to 0, the CSIH determines that the next communication is low-priority and , automatically switches memory mode to transmit-only buffer mode, and then resumes the aborted low-priority communication.

(5) Same as (1) above.

(6) Same as (2) above.

(7) The CPU detects an interrupt and starts communication by writing the first transmission data of high-priority communication to CSIHnTX0W or CSIHnTX0H. The CPU sets CSIHnCTL0.CSIHnJOBE = 1 again to notify that the next communication is high-priority.

(8) When end-of-job data is detected, communication is aborted and the INTCSIHTIJC interrupt is generated. At this time, the CPU determines that the subsequent communication is also high-priority because the internal signal JOB completion flag is 1, and waits for communication to start.

(9) Same as (3) above.

(10) Same as (4) above.

CAUTION

Memory mode is switched automatically when communication is changed from low priority to high priority (from transmit-only buffer mode to direct access mode) and from high priority to low priority (from direct access mode to transmit-only buffer mode).

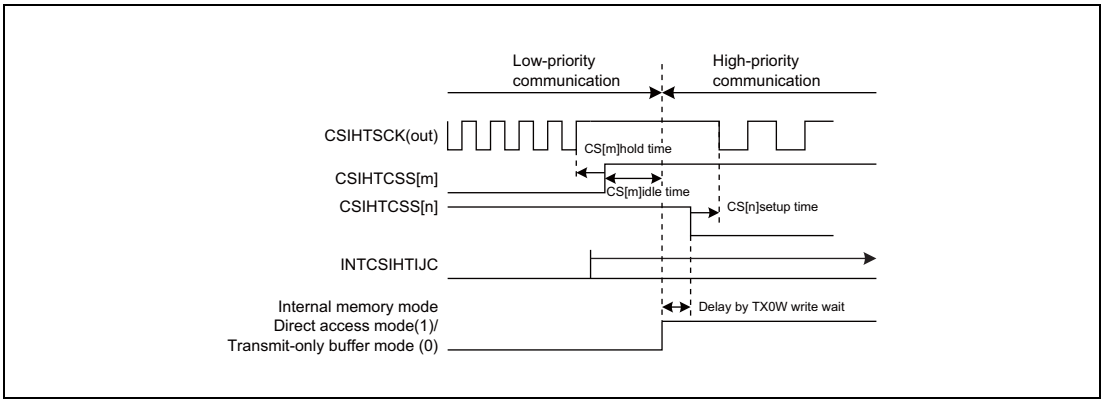


Figure 16.46 Transition from Low-Priority Mode to High-Priority Mode

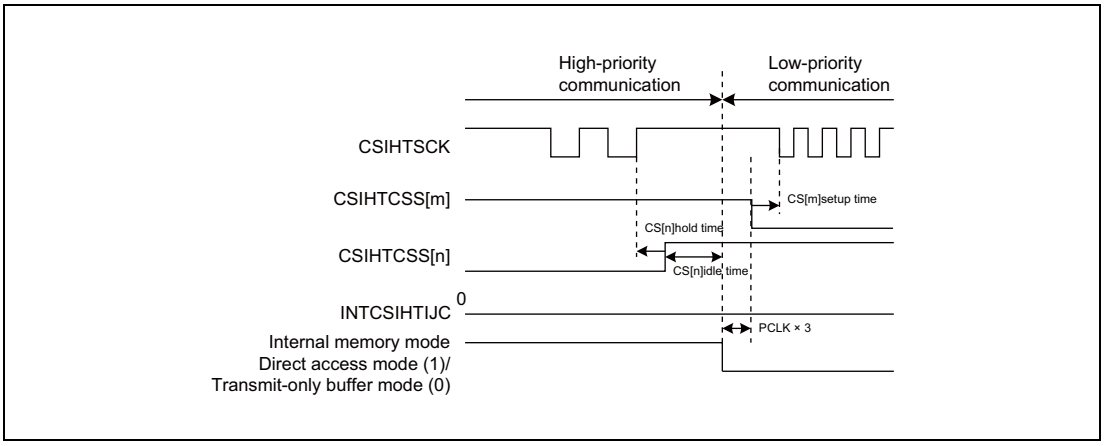


Figure 16.47 Transition from High-Priority Mode to Low-Priority Mode

Do not conduct write operation of communication data or CSIHnCTL0.CSIHnJOBE bit operation during setting inhibited period to switch low and high priority communication mode correctly.

CSIHnTX0W register write inhibited period:

- Period from when CSIHnJOBE bit is set for switching to high priority communication mode to when INTCSIHTIJC interrupt is detected.
- Period from when the last data of high priority communication (End of JOB data) is written to when the CSIHnHPST state = 0 is detected.

CSIHnJOBE register write inhibited period:

- Period from when CSIHnJOBE bit is set for switching to high priority communication mode to when INTCSIHTIJC interrupt is detected.

During high communication mode period, there is no setting inhibited period for CSIHnJOBE bit. It is possible to set CSIHnJOBE bit before writing communication data. For example, to communicate multiple JOB data in high priority mode, it is possible to set CSIHnJOBE bit before writing the first communication data.

CAUTION

When CSIHnJOBE bit is set right before the last communication of high priority communication ends, different operations are required depending on the timing at which setting of CSIHnJOBE bit is internally detected.

If the setting of the CSIHnJOBE bit is detected before the transfer of the last bit is completed, high priority communication mode continues.

When setting of the CSIHnJOBE bit is detected after the transfer of the last bit is completed, the mode temporarily returns to low priority communications. After detection of End of JOB data in low priority communications, the mode changes back to high priority communications.

16.5.15 Enforced Chip Select Idle Setting

It is possible to insert an idle state between two consecutive data transfers by setting CSIHnCFGx.CSIHnIDLx.

1. When CSIHnCFGx.CSIHnIDLx = 0
   If the next CSIHTCSSx is the same as the previous one, an idle state is not inserted and an inter-data time is inserted.
   If the next CSIHTCSSx is different from the previous one, an idle state is inserted.
2. When CSIHnCFGx.CSIHnIDLx = 1
   An idle state is always inserted even if the next CSIHTCSSx is not different from the previous one.

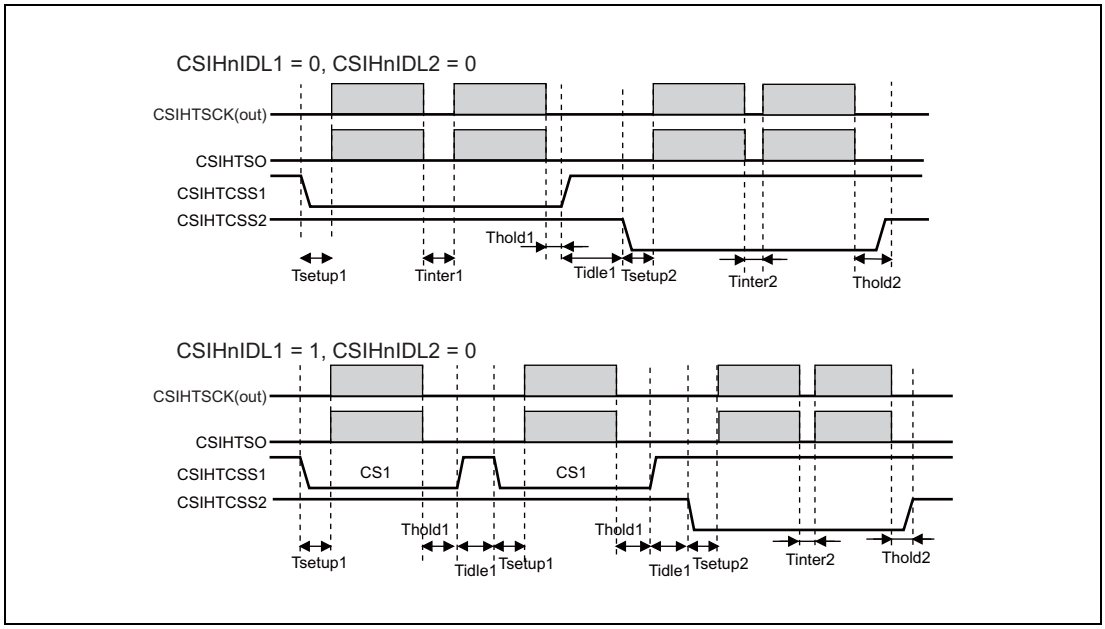


Figure 16.48 Enforced Chip Select Idle Setting Example

CAUTION

If the CPU-controlled high priority communication function is enabled (CSIHnCTL1.CSIHnPHE = 1), when the mode is switched from low priority communication to high priority communication or from high priority communication to low priority communication, an IDLE state is inserted regardless of the setting of CSIHnCFGx.CSIHnIDLx bit.

16.6 Operating Procedures

The examples and procedures below are described according to the memory mode in the following order:

- Direct access mode
- Transmit-only buffer mode
- Dual buffer mode
- FIFO mode

16.6.1 Procedures in Direct Access Mode

Two examples for a master are provided, one with job mode disabled, and the other with job mode enabled.

16.6.1.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedures below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- Transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)

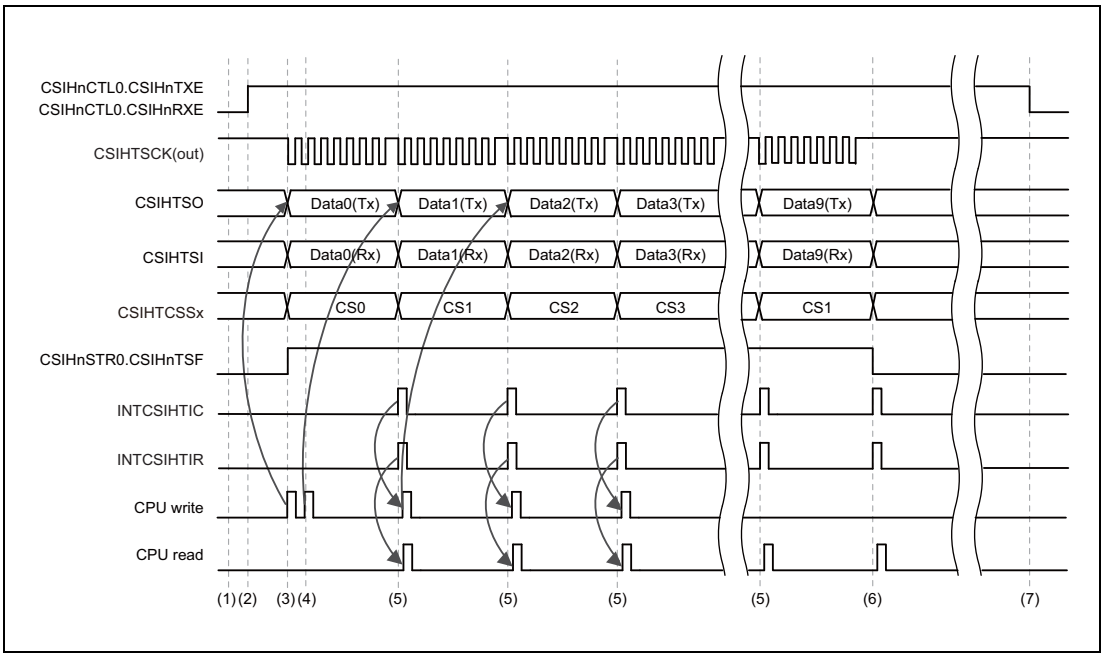


Figure 16.49 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 0

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
   Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits reception), and CSIHnMBS = 1 (selects direct access mode).
3. Write the first data to be sent to the transmission register CSIHnTX0W. Within the same write operation, activate CS0. Transmission starts automatically when the first data becomes available.
4. Write the second data to CSIHnTX0W. If required, you can change the CS to address a different device. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every data transmission/reception, the interrupts INTCSIHTIC and INTCSIHTIR are generated:
   – INTCSIHTIC indicates that the next data can be written to CSIHnTX0W.
   – INTCSIHTIR indicates that the reception register CSIHnRX0W must be read.
6. No more write action is required after completion of data 8. Data 9 (the last data) has been written in advance.
   However, reception register CSIHnRX0W must be read after completion of writing data 8 and 9.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

16.6.1.2 Transmit/Receive in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- Transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)
- Two jobs, each transmitting three data.

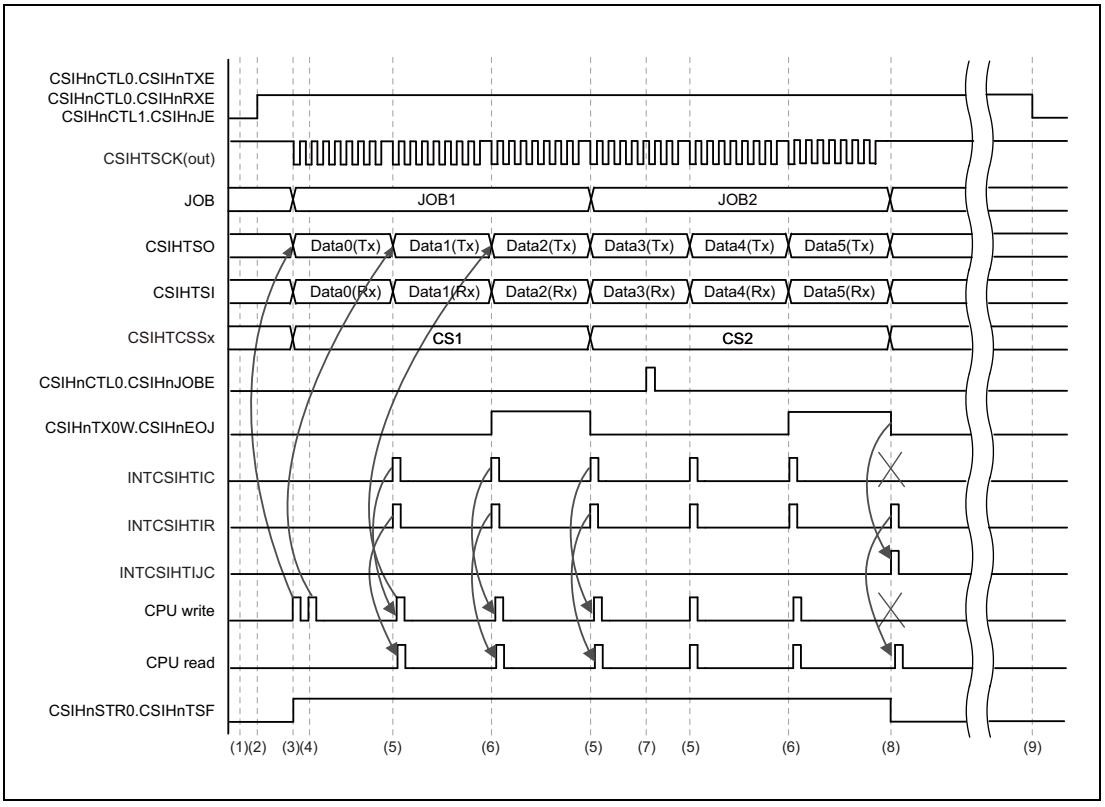


Figure 16.50 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS1 and CS2.
   Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits the reception), and CSIHnMBS = 1 (selects direct access mode).
3. Write the first data to be sent to the transmission register CSIHnTX0W. Transmission starts automatically when the first data becomes available.
   The CSIHnSTR0.CSIHnTSF flag indicates that communication is in progress.
4. Write the second data to CSIHnTX0W. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every data transmission/reception, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated.
   - INTCSIHTIC indicates that the next data can be written to CSIHnTX0W.
   - INTCSIHTIR indicates that the reception register CSIHnRX0W must be read.
6. Setting CSIHnTX0W.CSIHnEOJ = 1 indicates that the last data of the current job is sent. After that, the next job may begin.
7. By setting CSIHnCTL0.CSIHnJOBE = 1, communication is forced to stop at the end of the current job (JOB2).
8. After the forced stop of communication, the interrupt request INTCSIHTIC is replaced by INTCSIHTIJC. INTCSIHTIR is generated as usual.
   The interrupt request INTCSIHTIJC indicates that the communication was forcibly stopped at the end of the current job.
   The interrupt request INTCSIHTIC is not generated. Additionally, the transmission data available in the CSIHnTX0W register is not sent.
9. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.
   To start another transmission without stopping communication, perform steps 3 and later.

16.6.2 Procedures in Transmit-Only Buffer Mode

Two examples for a master is provided, one with job mode disabled, and the other one with job mode enabled.

16.6.2.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0,CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)

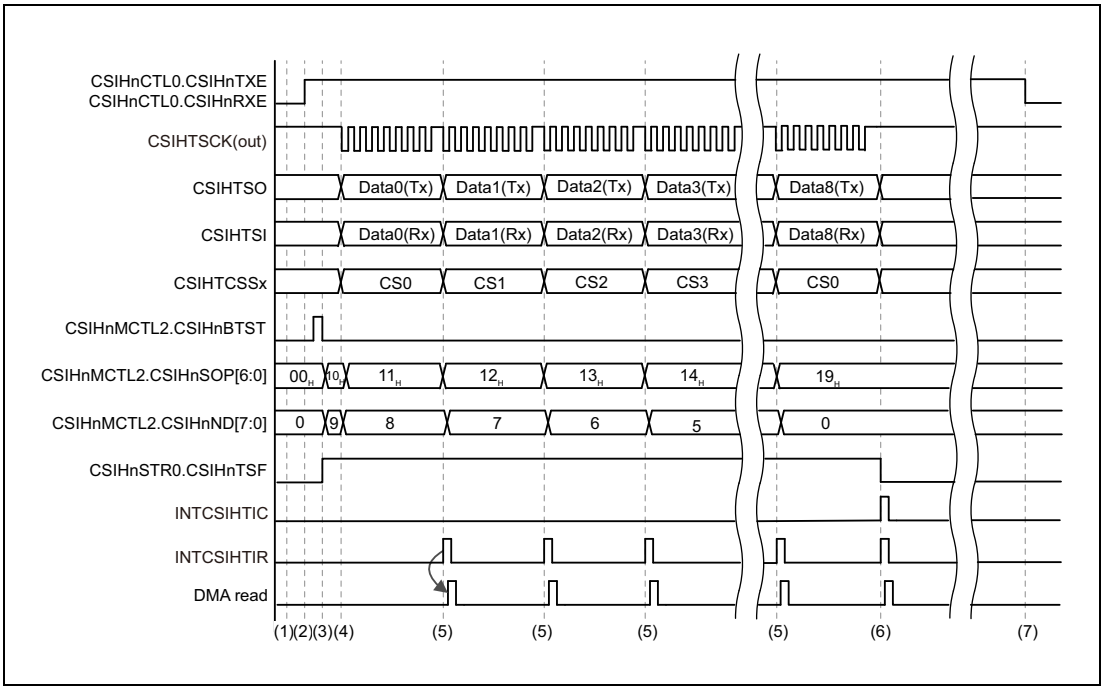


Figure 16.51 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The procedure of writing the data into the buffer is not described here.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (transmit-only buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure the transmission pointer and the number of data by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission/reception is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. After every data reception, the interrupt request INTCSIHTIR is generated. INTCSIHTIR indicates that the receive register CSIHnRX0W must be read.
6. When all transmissions are complete, the interrupt request INTCSIHTIC is generated.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

16.6.2.2 Transmit/Receive in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- The number of data is 8 (CSIHnMCTL2.CSIHnND[7:0] = 08_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0,CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)

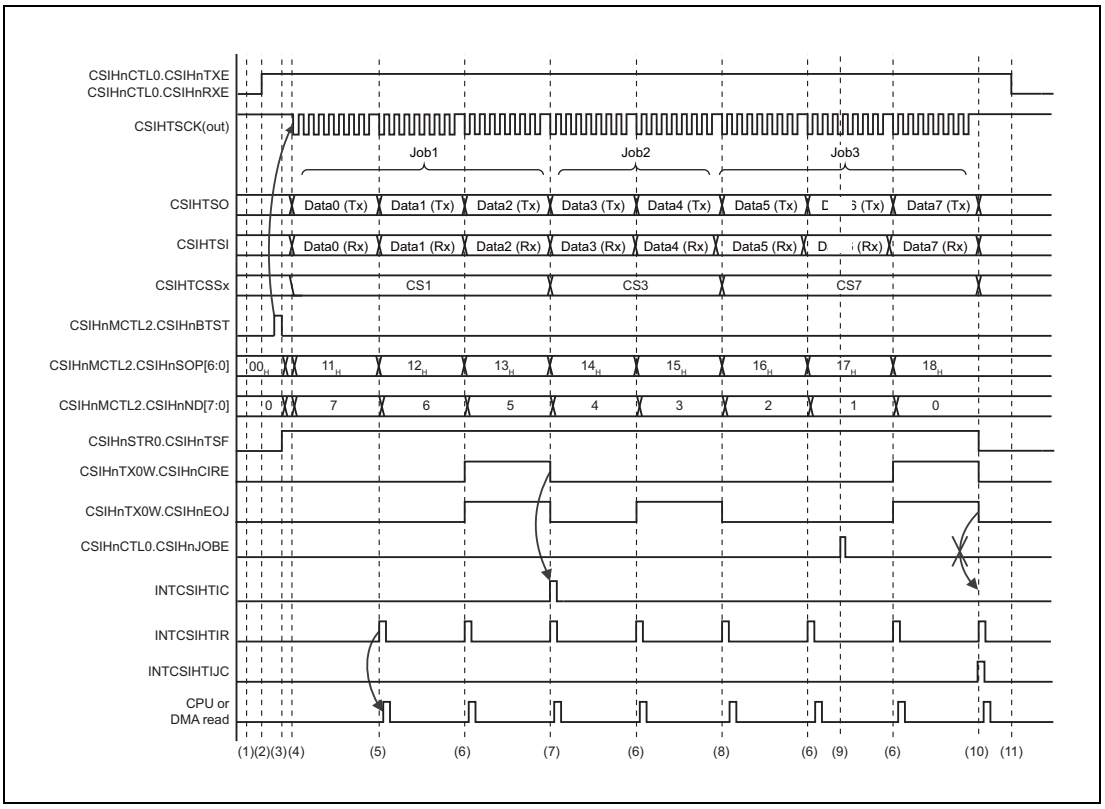


Figure 16.52 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 1

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS1, CS3, and CS7.
   Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (transmit-only buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure the send pointer and the number of data by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. After every data reception, the interrupt request INTCSIHTIR is generated. INTCSIHTIR indicates that the receive register CSIHnRX0W must be read.
6. The CSIHnTX0W.CSIHnEOJ = 1 setting indicates that the last data of the current job is sent.
7. The interrupt request INTCSIHTIC is generated. INTCSIHTIC indicates that the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
8. The INTCSIHTIC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CHABnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
9. By setting CSIHnCTL0.CSIHnJOBE = 1, communication is forced to stop at the end of JOB3.
10. After the forced stop of communication, interrupt requests INTCSIHTIJC and INTCSIHTIR are generated at the end of job3.
    The INTCSIHTIJC interrupt request indicates that the communication was forcibly stopped at the end of the current job.
    The INTCSIHTIC interrupt request is not generated because the INTCSIHTIJC interrupt request is generated instead of the INTCSIHTIC interrupt request. Additionally, the transmission data available in the CSIHnTX0W register is not sent.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

16.6.3 Procedures in Dual Buffer Mode

Examples when job mode is enabled in master mode, disabled in master mode, and disabled in slave mode are provided below.

16.6.3.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CHABnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)

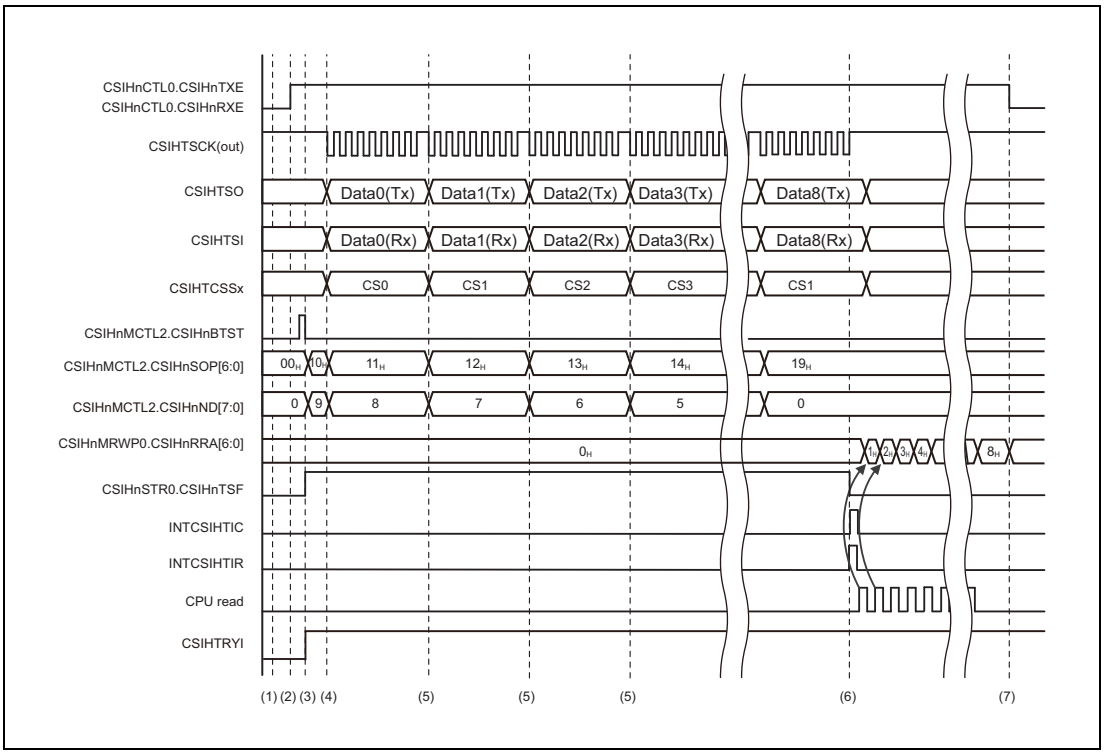


Figure 16.53 Master in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0]. Set CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits the reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure communication by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Permit buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. This is repeated until the last data is transmitted/received.
   The interrupt requests INTCSIHTIC and INTCSIHTIR are not generated.
6. When the last data is transmitted/received, the interrupt requests, INTCSIHTIC and INTCSIHTIR are generated.
   The CPU starts to read the received data from the receive buffer. The start address of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0]. These bits are incremented after each data is read.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

16.6.3.2 Transmit/Receive in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- The number of data is 8 (CSIHnMCTL2.CSIHnND[7:0] = 08_H).
- The transfer start address is 00_H (CSIHnMCTL2.CSIHnSOP[6:0] = 00_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)

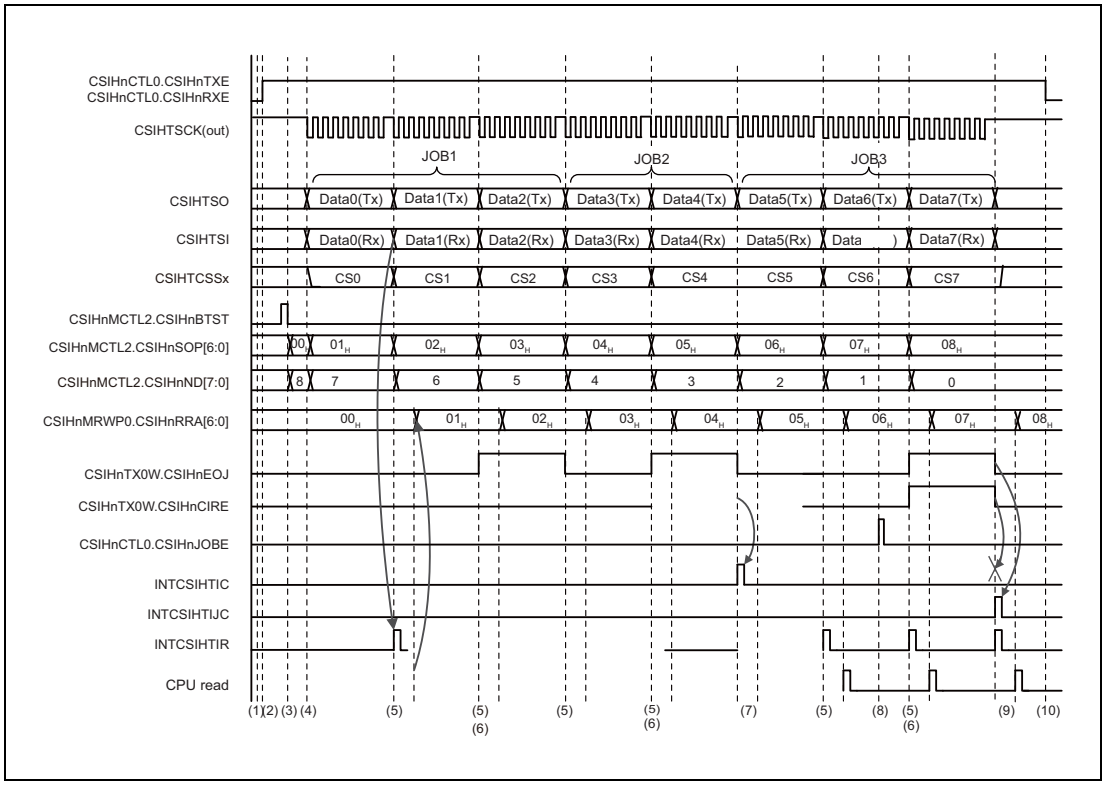


Figure 16.54 Master in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 1

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. The example uses chip select signals CS0 to CS7.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure communication by setting the CSIHnMCTL2.CSIHnSOP[6:0] bits and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission. This is repeated until the last data is transmitted/received.
5. The INTCSIHTIR interrupt request is generated everytime data is received.
   The INTCSIHTIC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
6. CSIHnTX0W.CSIHnEOJ = 1 indicates that the last data of the current job is sent.
7. The INTCSIHTIC interrupt request is generated. INTCSIHTIC indicates that the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
8. By setting CSIHnCTL0.CSIHnJOBE = 1, communication is forced to stop at the end of JOB3.
9. After the forced stop of communication, interrupt requests INTCSIHTIJC and INTCSIHTIR are generated at the end of JOB3.
   The INTCSIHTIJC interrupt request indicates that the communication was forcibly stopped at the end of the current job.
   The INTCSIHTIC interrupt request is not generated because the INTCSIHTIJC interrupt request is generated instead of the INTCSIHTIC interrupt request. Additionally, the transmission data available in register CSIHnTX0W is not sent.
10. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

16.6.3.3 Transmit/Receive in Slave Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCTL1.CSIHnCKR = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)
- Handshake function is enabled (CSIHnCTL1.CSIHnHSE = 1)

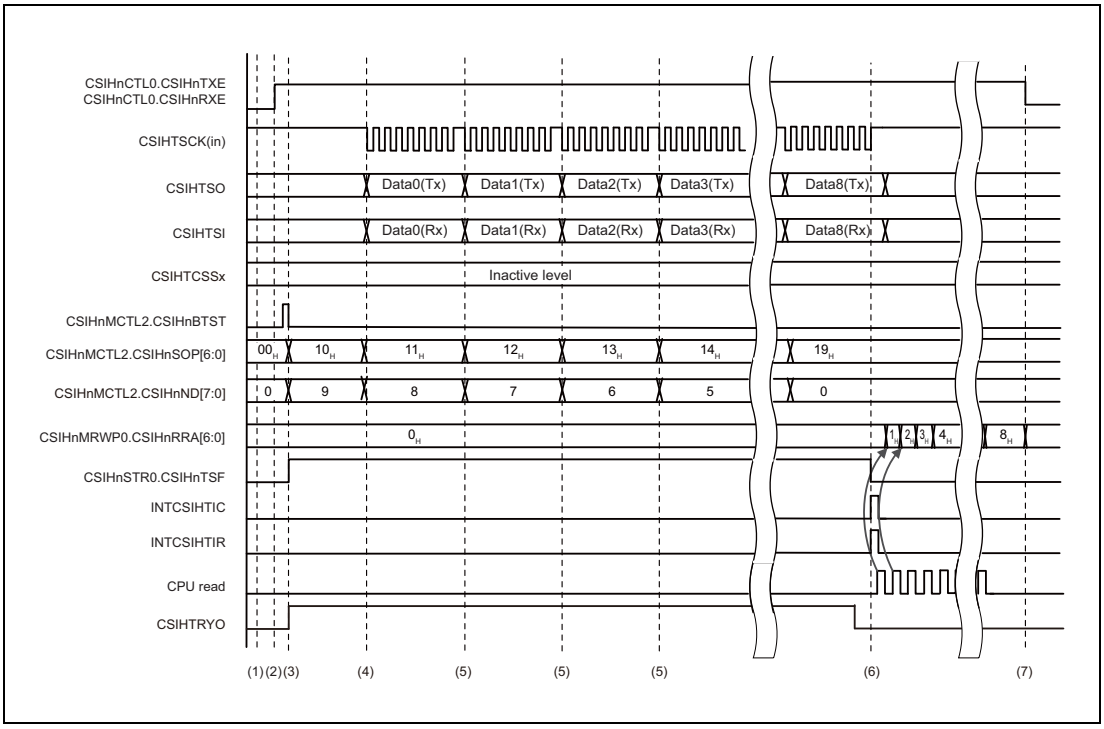


Figure 16.55 Slave in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in register CSIHnCFG0.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Specify the transfer start address by setting the CSIHnMCTL2.CSIHnSOP[6:0] bits and the number of data by setting the CSIHnMCTL2.CSIHnND[7:0] bits. Permit the buffer transfer by setting the CSIHnMCTL2.CSIHnBTST.
4. Transmission is started when the input clock from the master is received. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. This is repeated until the last data is transmitted/received.
   The interrupt requests INTCSIHTIC and INTCSIHTIR are not generated because transmission data is sent from the buffer, and received data is stored in the buffer.
6. When the last data is transmitted/received, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated.
   The CPU starts to read the received data that is stored in the receive buffer. The start address of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0]. These bits are incremented after each data is read.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

16.6.4 Procedures in FIFO Mode

Two examples for a master is provided, one with job mode disabled, the other one with job mode enabled.

16.6.4.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)

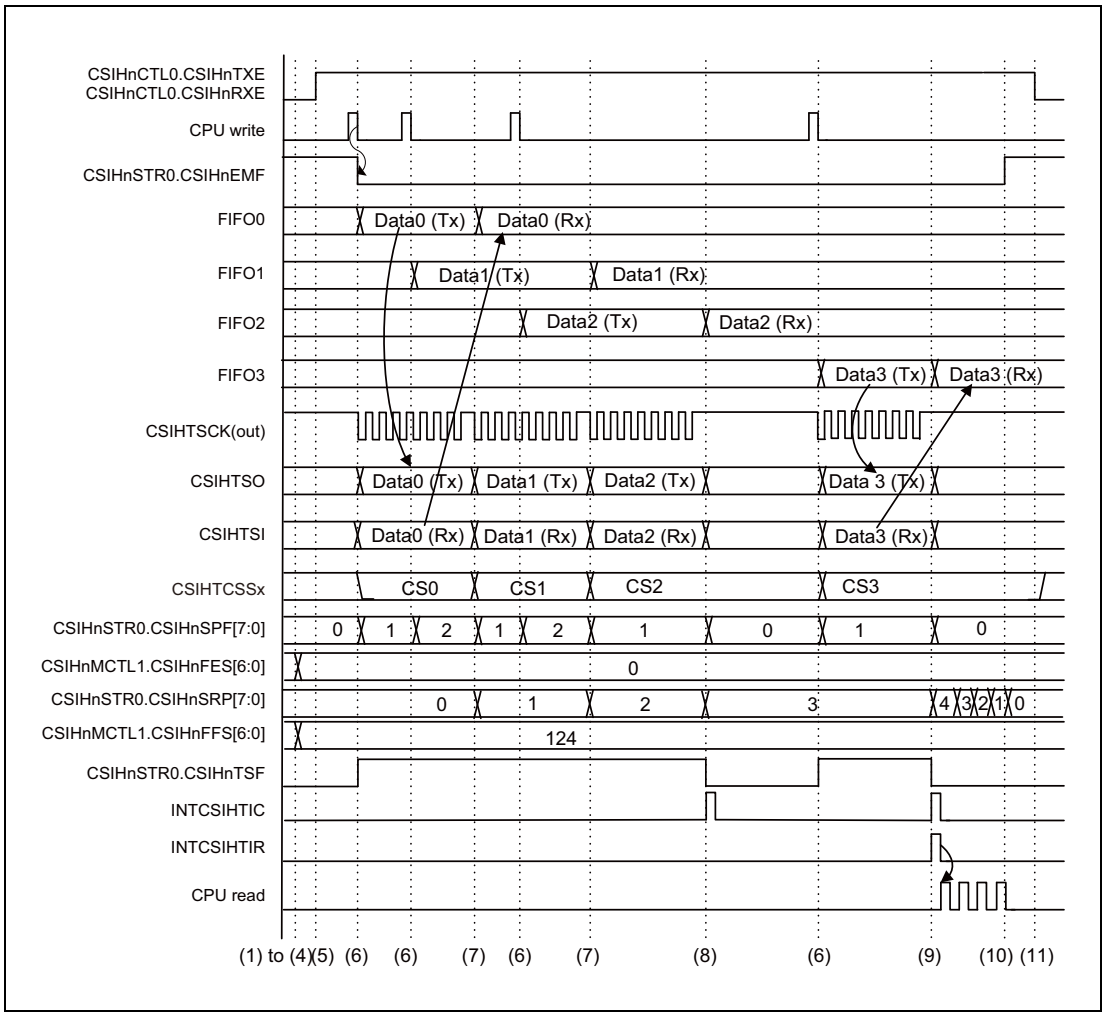


Figure 16.56 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 0

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. Specify the job mode disable and master mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2. Specify the FIFO mode by setting CSIHnMCTL0.CSIHnMMS[1:0] = 00_B. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
2. Set CSIHnSTCR0.CSIHnPCT = 1 to clear all buffer pointers.
3. Check that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00_H.
4. Specify the conditions for generating the INTCSIHTIC interrupt request in the CSIHnMCTL1.CSIHnFES[6:0] bits.
   Specify the conditions for generating the INTCSIHTIR interrupt request in the CSIHnMCTL1.CSIHnFFS[6:0] bits.
5. Set CSIHnCTL0.CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
6. Write the first transmit data to the transmit register CSIHnTX0W. Transmission starts automatically when the first data becomes available.
   Check that CSIHnSTR0.CSIHnEMF = 0.
7. The current transmission is completed. As the CSIHnFES[6:0] bits are not the same as the CSIHnSPF[7:0] bits, the interrupt request INTCSIHTIC is not generated.
8. As the CSIHnFES[6:0] bits are the same as the CSIHnSPF[7:0] bits, the interrupt request INTCSIHTIC is generated.
9. When CSIHnFFS[6:0] = 128 - CSIHnSRP[7:0], the interrupt request INTCSIHTIR is generated. Since CSIHnFES[6:0] = CSIHnSPF[7:0], the interrupt request INTCSIHTIC is generated. After the generation of an interrupt, the CPU starts reading received data that is stored in the receive buffer.
10. When the CPU completes reading the received data that is stored in the receive buffer, CSIHnSTR0.CSIHnEMF is set to 1 and the FIFO buffer becomes empty.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. In addition, if communication is not performed, set CSIHnCTL0.CSIHnPWR = 0 to minimize the power consumption of CSIHn.

16.6.4.2 Transmit/Receive Mode when Job Mode is Enabled in Master Mode

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- JOB1 consists of four data, JOB2 consists of three data, and JOB3 consists of five data.
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)

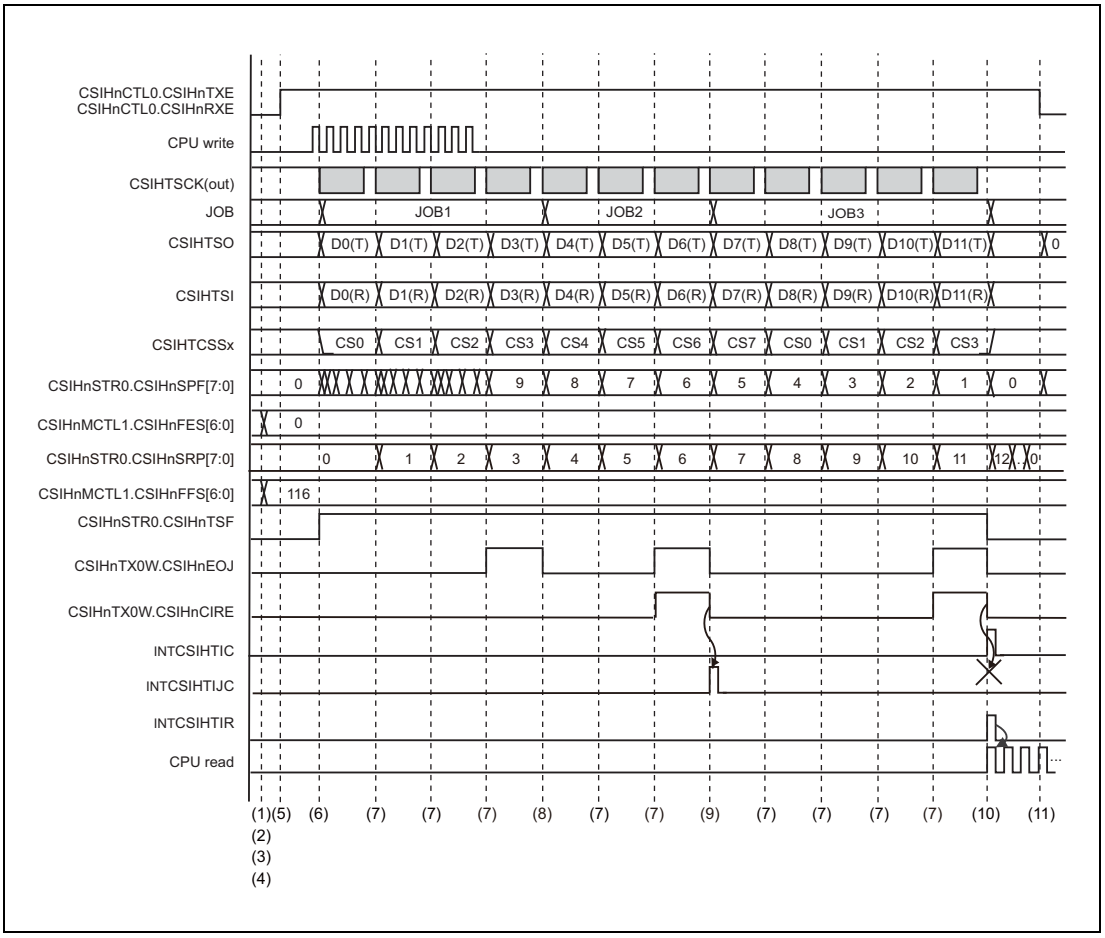


Figure 16.57 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. Set job mode disable and master mode in the corresponding bits of the CSIHnCTL1 and CSIHnCTL2 registers. Set FIFO mode by setting CSIHnMCTL0.CSIHnMMS[1:0] to 00_B. The example uses chip select signals CS0 to CS7.
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all buffer pointers.
3. Make sure that CSIHnSTR0.CSIHnFLF is set to 0, CSIHnSTR0.CSIHnEMF is set to 1, and CSIHnSTR0.CSIHnSPF[7:0] is set to 00_H.
4. With CSIHnMCTL1.CSIHnFES[6:0], specify the conditions for generating the INTCSIHTIC interrupt request. With CSIHnMCTL1.CSIHnFFS[6:0], specify the conditions for generating the INTCSIHTIR interrupt request.
5. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
6. Write the first data to be sent to the CSIHnTX0W transmission register. Transmission starts automatically when the first data becomes available.
   Make sure CSIHnSTR0.CSIHnEMF is set to 0.
7. The current transmission is completed. Since CSIHnFES[6:0] is not the same as CSIHnSPF[7:0], the interrupt request INTCSIHTIC is not generated.
8. The INTCSIHTIJC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
9. The INTCSIHTIJC interrupt request is generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
10. The INTCSIHTIC interrupt request is generated because CSIHnFES[6:0] = CSIHnSPF[7:0]. INTCSIHTIC is generated so that INTCSIHTIJC is not generated.
    When CSIHnFFS[6:0] = 128 - CSIHnSRP[7:0], the interrupt request INTCSIHTIR is generated. After the generation of the INTCSIHTIR interrupt, CPU starts reading the received data stored in received buffer.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

16.7 Detection and Correction of Errors in CSIHn RAM

16.7.1 ECC for the CSIHn RAM

Table 16.46 gives an outline of the ECC functions for the CSIHn RAM.

Table 16.46 List of the ECC Functions for the CSIHn RAM

Item	Outline of Functions
ECC error detection/ correction	The RAM is checked for ECC errors. The following options are selectable. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection The ECC error detection/correction can be disabled by using through mode. Error detection/correction is enabled by default.
Error notification	When an ECC 2-bit error is generated, an error notification is generated. • Error notification can be enabled or disabled when an ECC 2-bit error is detected. In the initial setting, 2-bit error notification is enabled. However, if interrupts are masked by the FEINTFMSK register, interrupt processing is not executed.
Error status	Detection of ECC 2-bit errors and ECC1-bit errors can be monitored. A bit for clearing the error status is provided.

16.7.2 Interrupt Requests

Table 16.47 lists the ECC interrupt requests of CSIHn RAM.

Table 16.47 CSIHn ECC Interrupt Requests (FE-Level Maskable Interrupt)

Unit interrupt signal	Description	Name	DMA Trigger Number
INTECCDCSIHn	CSIHn ECC2 Bit error interrupt	INTECCDCSIH0	—
		INTECCDCSIH1	
		INTECCDCSIH2	
		INTECCDCSIH3	

16.7.3 ECCCSIHnCTL — CSIHn ECC Control Register

The ECCCSIHnCTL register controls the ECC mode and the status of CSIH.

Bits 7, 5, and 4 should be set (written) while the CSIHn operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be 01_B.

Access: This register can be read or written in 16-bit units.

Address: FFC7 00n0$_H$ (n = 0 to 3)

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	EMCA0	—	—	—	ECER2C	ECER1C	—	ECTHM	—	EC1ECP	EC2EDIC	—	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	—
R/W	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R	R/W	R	R/W	R/W	R	R	R	R

Note 1. These bits are always read as 0.

Table 16.48 ECCCSIHnCTL Register Contents (1/2)

Bit Position	Bit Name	Function
15	EMCA1	Access Control to ECC Mode Selection Bit These bits specify whether updating the ECTHM bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 01_B, writing to bit 7 is enabled.
14	EMCA0	
13 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	ECER2C	2-Bit ECC Error Detection Flag Clear This bit clears 2-bit error detection flags of ECER2F (bit 2). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between this bit writing and ECER2F bit setting occurs, writing to this bit has a priority.
9	ECER1C	1-Bit ECC Error Detection Correction Accumulation Flag Clear This bit clears 1-bit error detection/correction flags of ECER1F (bit 1). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between writing to this bit and setting the ECER1F bit occurs, writing to this bit has a priority.
8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTHM	ECC Function through Mode Selection This bit specifies whether to enable or disable the ECC function. Setting this bit to 1 disables the ECC function. When writing to this bit, write 01_B to EMCA1 and EMCA0 at the same time. 0: Through mode is disabled (normal operation mode). 1: Through mode is enabled. (ECC function disable)
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-Bit Error Correction Enable This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.

Table 16.48 ECCCSIHnCTL Register Contents (2/2)

Bit Position	Bit Name	Function
4	EC2EDIC	2-Bit Error Detection Interrupt Control This bit controls whether to generate an interrupt when 2-bit error is detected. 0: When 2-bit error is detected, an INTECCDCSIHn interrupt will not be generated. 1: When 2-bit error is detected, an INTECCDCSIHn interrupt will be generated. (the value after reset)
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ECER2F	2-Bit Error Detection Flag This flag indicates whether 2-bit error is detected during read access to the RAM when error detection is enabled (ECTHM = 0). When 2-bit error interrupt is enabled (EC2EDIC = 1) and this flag is set, an ECC 2-bit error interrupt (INTECCDCSIHn) is output. Write 1 to the ECER2C bit (bit 10) to clear the flag. This bit is also cleared when through mode is enabled (ECTHM = 1). If 2-bit error is detected again while this bit is set, an interrupt will not be generated. 0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred. This bit is read-only. Writing 0 or 1 does not change internal state.
1	ECER1F	1-Bit Error Detection/Correction Flag This flag indicates whether 1-bit error is detected during read access to the RAM when error detection is enabled (ECTHM = 0). Write 1 to the ECER1C bit (bit 9) to clear the flag. This bit is also cleared when through mode is enabled (ECTHM = 1). 0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred. This bit is read-only. Writing 0 or 1 does not change internal state.
0	ECEMF	ECC Error Message Flag This flag indicates whether an error exists in the current read data bus. This bit is updated whenever the RAM outputs data. This bit is also cleared when through mode is enabled (ECTHM = 1) and there is no 1-bit error in the decode circuit input data. 0: The current RAM output data does not have bit errors. 1: The current RAM output data has bit errors.

CAUTION

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.

16.7.4 ECCCSIHnTMC — CSIHn ECC Test Mode Control Register

The ECCCSIHnTMC register is used to switch to and control the test mode.

This register can be used when CSIHn is not accessing the RAM.

When writing to bit 7, ETMA1 and ETMA0 need to be 10_B.

Access: This register can be read or written in 16-bit units.

Address: FFC7 00n4$_H$ (n = 0 to 3)

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA1	ETMA0	—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are always read as 0.

Table 16.49 ECCCSIHnTMC Register Contents (1/2)

Bit Position	Bit Name	Function
15	ETMA1	Access Control to ECC Test Mode Bit These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 10_B, writing to bit 7 is enabled.
14	ETMA0	
13 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTMCE	ECC Test Mode Enable This bit specifies whether to enable access to the test registers and to the test control bits of this register. When writing to this bit, write 10_B to ETMA1 and ETMA0 at the same time. 0: Access to the test registers and test control bits is disabled. 1: Access to the test registers and test control bits is enabled. Test registers: ECCCSIHnTED, ECCCSIHnTRC, ECCCSIHnSYND, ECCCSIHnHORD, ECCCSIHnECRD, ECCCSIHnERDB Test control bits: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	ECTRRS	ECC RAM Read Test Mode Selection This bit selects the targets for reading when the ECCCSIHnTED and ECCCSIHnERDB registers are read. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Read value of the ECCCSIHnTED register will be the write value of the ECCCSIHnTED register. Read value of the ECCCSIHnERDB register will be the write value of the ECCCSIHnERDB register. 1: Read value of the ECCCSIHnTED register can read RAM data. Read value of the ECCCSIHnERDB register will be the ECC Data to be written to RAM.
3	ECREOS	ECC Redundant Bit Output Data Selection This bit specifies whether to store the ECC data generated for write data or the value of the ECCCSIHnERDB register as ECC data in RAM. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: ECC data generated for write data is stored in RAM. 1: The value of ECCCSIHnERDB register is stored in RAM.

Table 16.49 ECCCSIHnTMC Register Contents (2/2)

Bit Position	Bit Name	Function
2	ECENS	ECC Encoder Input Selection This bit specifies whether to use the data written to RAM or the value of the ECCCSIHnTED register as the input to the ECC encoder. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: ECC data is generated from write data to RAM 1: ECC data is generated from register value of the ECCCSIHnTED.
1	ECDCS	ECC Decoder Input Selection This bit specifies whether to use the RAM data or the value of the ECCCSIHnTED register as the target data for generation of syndrome code and error detection. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Syndrome code generation and error detection are performed from RAM data. 1: Syndrome code generation and error detection are performed from ECCCSIHnTED register value.
0	ECREIS	ECC Redundant Bit Input Data Selection This bit specifies whether to use the ECC data stored in RAM or the value of the ECCCSIHnERDB register as the target ECC data for generation of syndrome code and error detection. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Syndrome code generation and error detection are performed from ECC data stored in RAM. 1: Syndrome code generation and error detection are performed from ECCCSIHnERDB register value.

16.7.5 ECCCSIHnTED — CSIHn ECC Encode/Decode Input/Output Replacement Test Register

In ECC test mode, this register handles test data.

The value of this register is used to generate ECC data or syndrome code.

This register can be accessed when ECC test mode in enabled (ECCCSIHnTMC.ECTMCE = 1). When ECCCSIHnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when CSIHn is not accessing the RAM.

Access: This register can be read or written in 32-bit units.

Address: FFC7 $00nC_H$ (n = 0 to 3)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.50 ECCCSIHnTED Register Contents

Bit Position	Bit Name	Function
31 to 0	ECEDB[31:0]	When ECCCSIHnTMC.ECENS = 1, these bits generate ECC data from value of this register and store the register value in RAM. When ECCCSIHnTMC.ECDCS = 1, these bits generate syndrome code from the value of the register and store the register value in ECC decode syndrome data register (ECCCSIHnSYND). In addition, when ECCCSIHnTMC.ECTRRS = 1, RAM data [31:0] instead of written data is read as the value of this register.

16.7.6 ECCCSIHnTRC — CSIHn ECC Redundant Bit Data Control Test Register

This register is a test register for ECC data in ECC test mode and consists of four 8-bit registers, ECCCSIHnSYND, ECCCSIHnHORD, ECCCSIHnECRD, and ECCCSIHnERDB.

This register can be accessed when ECC test mode is enabled (ECCCSIHnTMC.ECTMCE = 1). When ECCCSIHnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when CSIHn is not accessing the RAM.

Access: This register can be read or written in 32-bit units.

Address: FFC7 $00n8_H$ (n = 0 to 3)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECCCSIHnSYND (see **Section 16.7.7**)								ECCCSIHnHORD (see **Section 16.7.8**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECCCSIHnECRD (see **Section 16.7.9**)								ECCCSIHnERDB (see **Section 16.7.10**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

16.7.7 ECCCSIHnSYND — CSIHn ECC Decode Syndrome Data Register

This register is a read-only register for storing generated syndrome data in ECC test mode.

Writing to this register is ignored.

This register can only be read when ECC test mode is enabled (ECCCSIHnTMC.ECTMCE = 1). When ECC test mode is disabled (ECCCSIHnTMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FFC7 $00nB_H$ (n = 0 to 3)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SYND6	SYND5	SYND4	SYND3	SYND2	SYND1	SYND0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.51 ECCCSIHnSYND Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	SYND[6:0]	These bits store generated syndrome code as needed.

16.7.8 ECCCSIHnHORD — CSIHn ECC 7-Bit Redundant Bit Data Hold Test Register

This register is for storing ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can only be read when ECC test mode is enabled (ECCCSIHnTMC.ECTMCE = 1). When ECC test mode is disabled (ECCCSIHnTMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FFC7 00nA_H (n = 0 to 3)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	HORD6	HORD5	HORD4	HORD3	HORD2	HORD1	HORD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.52 ECCCSIHnHORD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	HORD[6:0]	These bits store ECC code for read RAM data as needed. ECC code is also stored when ECCCSIHnTMC.ECTRRS = 1 and if ECCCSIHnTED register is read.

16.7.9 ECCCSIHnECRD — CSIHn ECC Encode Test Register

This register is a read-only register for storing generated ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can only be read when ECC test mode is enabled (ECCCSIHnTMC.ECTMCE = 1). When ECC test mode is disabled (ECCCSIHnTMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FFC7 00n9_H (n = 0 to 3)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ECRD6	ECRD5	ECRD4	ECRD3	ECRD2	ECRD1	ECRD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 16.53 ECCCSIHnECRD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECRD[6:0]	When ECCCSIHnTMC.ECENS = 1,these bits indicate the ECC data generated for the data written to the ECCCSIHnTED register.

16.7.10 ECCCSIHnERDB — CSIHn ECC Redundant Bit Input/Output Replacement Buffer Register

In ECC test mode, this register handles ECC data.

The value of this register can be handled as the ECC data generated when data is written to the RAM or as the ECC data which is read when RAM data is read.

This register can be accessed when ECC test mode in enabled (ECCCSIHnTMC.ECTMCE = 1). When ECCCSIHnTMC.ECTMCE = 0, writing to this register is ignored and 00_H is read.

Access: This register can be read or written in 8-bit units.

Address: FFC7 00n8$_H$ (n = 0 to 3)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ERDB6	ERDB5	ERDB4	ERDB3	ERDB2	ERDB1	ERDB0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.54 ECCCSIHnERDB Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ERDB[6:0]	When ECCCSIHnTMC.ECREOS = 1, the value of this register is stored as the ECC data in the RAM. When ECCCSIHnTEC.ECREIS = 1, the value of this register takes the ECC data read from the RAM. When ECCCSIHnTMC.ECTRRS = 1, ECC data to be stored in RAM will be read for this register value instead of written data.

16.7.11 SELB_READTEST — ECCREAD Test Select Register

SELB_READTEST is used to check read/write access to the CSIHn ECC registers and RS-CANn ECC registers and FLXAn ECC registers.

Setting the bit corresponding to each function to 1 will enable writing to the read-only bit.

Access: This register can be read or written in 16-bit units.

Address: FFC7 8000_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RTCANE7A0 [2:1]		—	—	—	—	—	RTCAN E7A0 [0]	RTFLX AE7A0	RTFLX ATRAM 1	RTFLX ATRAM 0	RTCSIH E7A3	RTCSIH E7A2	RTCSIH E7A1	RTCSIH E7A0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.55 SELB_READTEST Register Contents (1/2)

Bit Position	Bit Name	Function
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14, 13	RTCANE7A0 [2:1]	RS-CAN0 ECC*1 Register Write Access for Testing Enable/Disable 00_B: Write access for testing is disabled. 11_B: Write access for testing is enabled. (The RS-CAN0 ECC read only bit can be written.) Operation is not guaranteed if other than the above settings are configured.
12 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	RTCANE7A0 [0]	RS-CAN0 ECC*1 Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (RS-CAN0 ECC read-only bit can be written).
6	RTFLXAE7A0	FLXA (MRAM) ECC*2 Register Write Access for Testing Enable/Disable 0: Write access for testing is disabled. 1: Write access for testing is enabled (FLXA (MRAM) ECC read-only bit can be written).
5	RTFLXATRAM1	FLXA (TBF) ECC*2 Register Write Access for Testing Enable/Disable 0: Write access for testing is disabled. 1: Write access for testing is enabled (FLXA (TBF) ECC read-only bit can be written).
4	RTFLXATRAM0	FLXA (TBF) ECC*2 Register Write Access for Testing Enable/Disable 0: Write access for testing is disabled. 1: Write access for testing is enabled (FLXA (TBF) ECC read-only bit can be written).
3	RTCSIHE7A3	CSIH3 ECC Register Write Access for Testing Enable/Disable 0: Write access for testing is disabled. 1: Write access for testing is enabled (CSIH3 ECC read-only bit can be written).

Table 16.55 SELB_READTEST Register Contents (2/2)

Bit Position	Bit Name	Function
2	RTCSIHE7A2	CSIH2 ECC Register Write Access for Testing Enable/Disable 0: Write access for testing is disabled. 1: Write access for testing is enabled (CSIH2 ECC read-only bit can be written).
1	RTCSIHE7A1	CSIH1 ECC Register Write Access for Testing Enable/Disable 0: Write access for testing is disabled. 1: Write access for testing is enabled (CSIH1 ECC read-only bit can be written).
0	RTCSIHE7A0	CSIH0 ECC Register Write Access for Testing Enable/Disable 0: Write access for testing is disabled. 1: Write access for testing is enabled (CSIH0 ECC read-only bit can be written).

Note 1. For details, see **Section 20.11, Detection and Correction of Errors in RS-CAN RAM**.

Note 2. For details, see **Section 21.4, Detection and Correction of Errors in FlexRay RAM**.

CAUTION

To enable the RS-CAN0 ECC register write access for testing, set SELB_READTEST.RTCANE7A0[2:0] to 111_B

Section 17 LIN Master Interface (RLIN2)

This section contains a generic description of the LIN master interface (RLIN2).

The first part in this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RLIN2.

17.1 Features of RH850/F1M RLIN2

17.1.1 Number of Units and Channels

This microcontroller has the following number of RLIN2 units and channels.

Table 17.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	2	3	
Name	RLIN24n (n = 0, 1)	RLIN24n (n = 0 to 2)	

Table 17.2 Configurations of RLIN2 Units and Correspondence between the Number of Channels per Unit and the Channel Numbers of the Units

Unit Name	Channels per Unit	Unit Channel Number	Channel Name RLIN2m	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
RLIN240	4	0	RLIN20	√	√	√
		1	RLIN21	√	√	√
		2	RLIN22	√	√	√
		3	RLIN23	√	√	√
RLIN241	4	0	RLIN24	√	√	√
		1	RLIN25	√	√	√
		2	RLIN26	—	√	√
		3	RLIN27	—	√	√
RLIN242	2	0	RLIN28	—	√	√
		1	RLIN29	—	√	√

Table 17.3 Indices

Index	Description
n	Throughout this section, the individual RLIN2 units are identified by the index “n” (n = 0 to 2).
m	Throughout this section, the individual channels are identified by the index “m” (m = 0 to 9).
i	Throughout this section, the individual channels of units that compose RLIN2 are identified by the index “i” (i = 0 to 3).
b	Throughout this section, the individual data buffers implemented in RLIN2 are identified by the index “b” (b = 1 to 8).

For example, RLN24nGLWBR is the LIN wake-up baud rate select register, which is the global register of RLIN2. RLN24nmLiMD is the LIN mode register, which is the channel register.

The following lists the indices corresponding to each product.

Table 17.4 Indices Correspondence of Each Product

Indices Correspondence of Each Product		
144 pins	**176 pins**	**233 pins**
i = 0 to 3 (RLIN240) i = 0, 1 (RLIN241)	i = 0 to 3 (RLIN240) i = 0 to 3 (RLIN241) i = 0, 1 (RLIN242)	i = 0 to 3 (RLIN240) i = 0 to 3 (RLIN241) i = 0, 1 (RLIN242)
b = 1 to 8	b = 1 to 8	b = 1 to 8

17.1.2 Register Base Address

RLIN2 base addresses are listed in the following table.

RLIN2 register addresses are given as offsets from the base addresses.

Table 17.5 Register Base Addresses

Base Address Name	Base Address
<RLIN240_base>	FFCE 0000_H
<RLIN241_base>	FFCE 0080_H
<RLIN242_base>	FFCE 0100_H

17.1.3 Clock Supply

The RLIN2 clock supply is shown in the following table.

Table 17.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RLIN24n	LIN communication clock sources	CKSCLK_ILIN
	Register access clock	CKSCLK_ILIN

17.1.4 Interrupt Request

RLIN2 interrupt requests are listed in the following table.

Table 17.7 Interrupt Requests (1/2)

Unit Interrupt Signal	Description	Interrupt Number
RLIN240		
INTRLIN20	RLIN20 interrupt	58
INTRLIN21	RLIN21 interrupt	59
INTRLIN22	RLIN22 interrupt	162
INTRLIN23	RLIN23 interrupt	163

Table 17.7 Interrupt Requests (2/2)

Unit Interrupt Signal	Description	Interrupt Number
RLIN241		
INTRLIN24	RLIN24 interrupt	226
INTRLIN25	RLIN25 interrupt	227
INTRLIN26	RLIN26 interrupt	275
INTRLIN27	RLIN27 interrupt	276
RLIN242		
INTRLIN28	RLIN28 interrupt	285
INTRLIN29	RLIN29 interrupt	286

17.1.5 Reset Sources

RLIN2 reset sources are listed in the following table. RLIN2 is initialized by these reset sources.

Table 17.8 Reset Sources

Unit Name	Reset Source
RLIN24n	All reset sources (ISORES)

17.1.6 External Input/Output Signals

External input/output signals of RLIN2 are listed in the following table.

Table 17.9 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
RLIN240		
RLIN2mRX (m = 0 to 3)	RLIN240 receive data input	RLIN2mRX (m = 0 to 3)
RLIN2mTX (m = 0 to 3)	RLIN240 transmit data output	RLIN2mTX (m = 0 to 3)
RLIN241		
RLIN2mRX (m = 4 to 7)	RLIN241 receive data input	RLIN2mRX (m = 4 to 7)
RLIN2mTX (m = 4 to 7)	RLIN241 transmit data output	RLIN2mTX (m = 4 to 7)
RLIN242		
RLIN2mRX (m = 8, 9)	RLIN242 receive data input	RLIN2mRX (m = 8, 9)
RLIN2mTX (m = 8, 9)	RLIN242 transmit data output	RLIN2mTX (m = 8, 9)

CAUTION

When the P0_0 pin is used as RLIN20RX, the output on the P0_0 pin ($\overline{\text{RESETOUT}}$ signal) is at the low level during a reset and after release from the reset state.

For details, see Section 2.11.1.1, P0_0: $\overline{\text{RESETOUT}}$.

17.2 Overview

17.2.1 Functional Overview

The LIN Master Interface is a hardware LIN communication controller that complies with LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAEJ2602 (SEP 2005), and automatically performs frame communication and error determination.

Table 17.10 shows the LIN Master Interface specifications.

Table 17.10 LIN Master Interface Specifications

Item		Specifications
Channel count		10 channels (In this product, 2-channel version and 4-channel version of RLIN2 is included.)
LIN communication function	Protocol	LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAEJ2602 (SEP 2005)
	Variable frame structure	• Transmission break width: 13 to 28 Tbits • Transmission break delimiter width: 1 to 4 Tbits • Inter-byte space (header): 0 to 7 Tbits (space between Sync field and ID field)*1 • Response space: 0 to 7 Tbits*1 • Inter-byte space: 0 to 3 Tbits (space between data bytes in response area) • Transmit wake-up: 1 to 16 Tbits
	Checksum	• Automatic operation for both transmission and reception • Classic or enhanced selectable (definable by frame)
	Response field data byte count	Variable from 0 to 8 bytes
	Frame communication modes	• Mode in which header transmission and response transmission/reception are started with a single transmission start request • Mode in which header transmission and response transmission are started with separate transmission start requests (frame separate mode)
	Wake-up transmission and reception	Available in LIN wake-up mode • Wake-up transmission function (1 to 16 Tbits) • Wake-up reception Low-level width of input signals measured
	Status	• Successful frame/wake-up transmission • Successful header transmission • Successful frame/wake-up reception*2 • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)
	Error status	• Bit error • Checksum error • Frame timeout error • Physical bus error • Framing error
	Baud rate selection	Baud rate conforming to the LIN specifications generated using baud rate generator
	Test mode	Self-test mode for user evaluation
	Interrupt function	• Successful frame/wake-up transmission • Successful frame/wake-up reception*2 • Error detection The logical OR of these three events is the interrupt source (INTRLIN2m) for each channel.

Note 1. Since the same register is used for configuration, the inter-byte space (header) = response space.
Note 2. For wake-up reception, the input signal low-level width count is indicated.

17.2.2 Block Diagram

Figure 17.1 shows a block diagram of the LIN master interface.

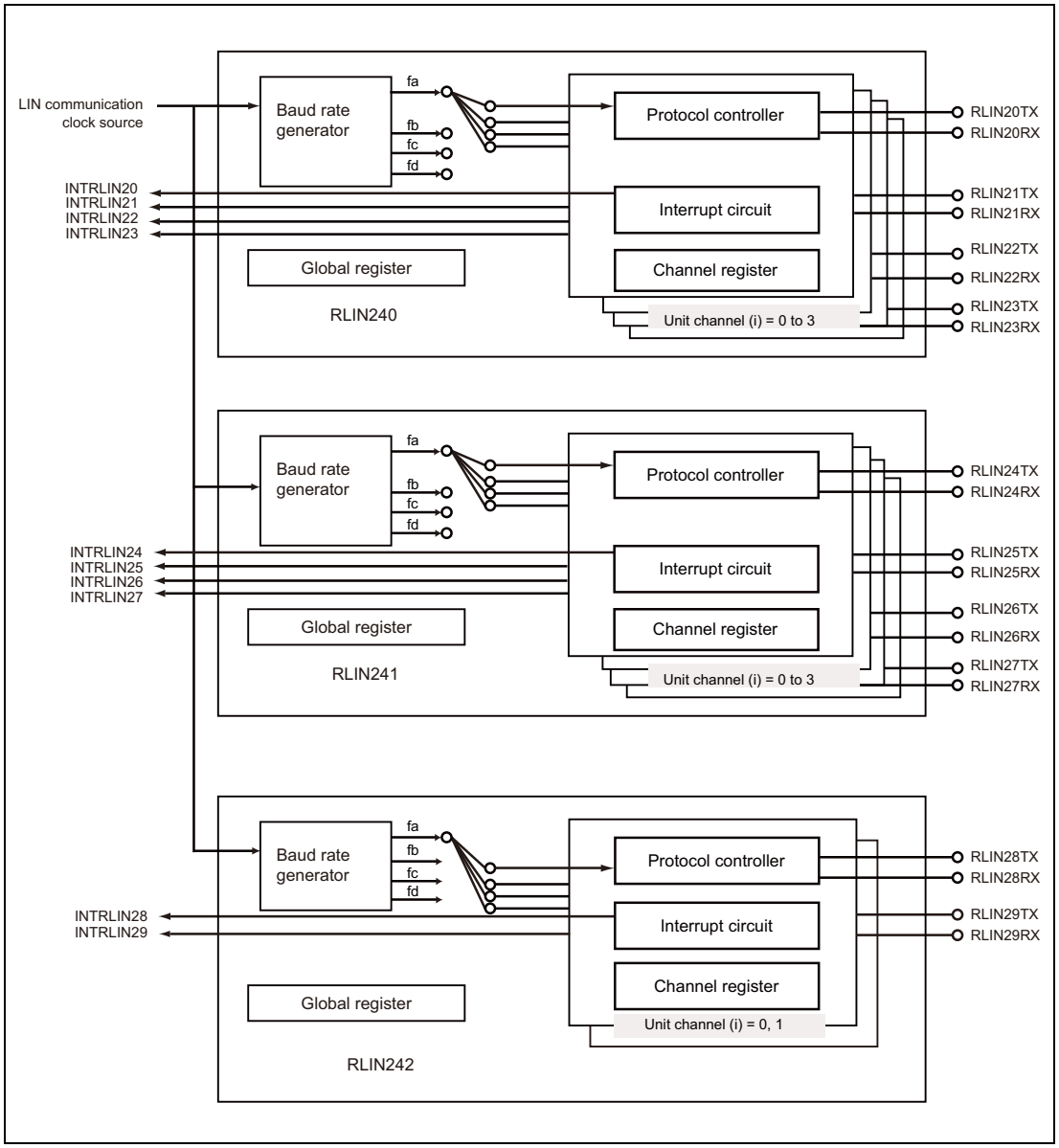


Figure 17.1 LIN Master Interface Block Diagram (232 Pins, RLIN2 10 Channels Included)

17.3 Registers

The registers of the LIN master interface include global registers and channel registers. As the global registers are allocated for each unit, they can be individually set for each unit. As the channel registers are allocated for each channel, individually control each channel.

17.3.1 List of Registers

RLIN2 registers are listed in the following table.

For details about <RLIN24n_base>, see **Section 17.1.2, Register Base Address**.

Table 17.11 List of Registers

Module	Register	Symbol	Address
Global registers			
RLN24n	LIN wake-up baud rate selection register	RLN24nGLWBR	<RLIN24n_base> + 01_H
RLN24n	LIN baud rate prescaler 0 register	RLN24nGLBRP0	<RLIN24n_base> + 02_H
RLN24n	LIN baud rate prescaler 1 register	RLN24nGLBRP1	<RLIN24n_base> + 03_H
RLN24n	LIN self-test control register	RLN24nGLSTC	<RLIN24n_base> + 04_H
Channel registers			
RLN24nm	LIN mode register	RLN24nmLiMD	<RLIN24n_base> + 08_H + i × 20_H
RLN24nm	LIN break field configuration register	RLN24nmLiBFC	<RLIN24n_base> + 09_H + i × 20_H
RLN24nm	LIN space configuration register	RLN24nmLiSC	<RLIN24n_base> + $0A_H$ + i × 20_H
RLN24nm	LIN wake-up configuration register	RLN24nmLiWUP	<RLIN24n_base> + $0B_H$ + i × 20_H
RLN24nm	LIN interrupt enable register	RLN24nmLiIE	<RLIN24n_base> + $0C_H$ + i × 20_H
RLN24nm	LIN error detection enable register	RLN24nmLiEDE	<RLIN24n_base> + $0D_H$ + i × 20_H
RLN24nm	LIN control register	RLN24nmLiCUC	<RLIN24n_base> + $0E_H$ + i × 20_H
RLN24nm	LIN transmission control register	RLN24nmLiTRC	<RLIN24n_base> + 10_H + i × 20_H
RLN24nm	LIN mode status register	RLN24nmLiMST	<RLIN24n_base> + 11_H + i × 20_H
RLN24nm	LIN status register	RLN24nmLiST	<RLIN24n_base> + 12_H + i × 20_H
RLN24nm	LIN error status register	RLN24nmLiEST	<RLIN24n_base> + 13_H + i × 20_H
RLN24nm	LIN data field configuration register	RLN24nmLiDFC	<RLIN24n_base> + 14_H + i × 20_H
RLN24nm	LIN ID buffer register	RLN24nmLiIDB	<RLIN24n_base> + 15_H + i × 20_H
RLN24nm	LIN checksum buffer register	RLN24nmLiCBR	<RLIN24n_base> + 16_H + i × 20_H
RLN24nm	LIN data buffer 1 register	RLN24nmLiDBR1	<RLIN24n_base> + 18_H + i × 20_H
RLN24nm	LIN data buffer 2 register	RLN24nmLiDBR2	<RLIN24n_base> + 19_H + i × 20_H
RLN24nm	LIN data buffer 3 register	RLN24nmLiDBR3	<RLIN24n_base> + $1A_H$ + i × 20_H
RLN24nm	LIN data buffer 4 register	RLN24nmLiDBR4	<RLIN24n_base> + $1B_H$ + i × 20_H
RLN24nm	LIN data buffer 5 register	RLN24nmLiDBR5	<RLIN24n_base> + $1C_H$ + i × 20_H
RLN24nm	LIN data buffer 6 register	RLN24nmLiDBR6	<RLIN24n_base> + $1D_H$ + i × 20_H
RLN24nm	LIN data buffer 7 register	RLN24nmLiDBR7	<RLIN24n_base> + $1E_H$ + i × 20_H
RLN24nm	LIN data buffer 8 register	RLN24nmLiDBR8	<RLIN24n_base> + $1F_H$ + i × 20_H

Note: When writing to a register not used, write the value after reset.

17.3.2 Global Registers

17.3.2.1 RLN24nGLWBR — LIN Wake-Up Baud Rate Selection Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nGLWBR: <RLIN24n_base> + 01_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LWBR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 17.12 RLN24nGLWBR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LWBR0	Wake-up Baud Rate Selection Bit 0: In LIN wake-up mode, the clock specified by the LCKS bit setting in the RLN24nmLiMD register is used (when LIN 1.3 is used). 1: In LIN wake-up mode, the clock fa is used regardless of the setting of the LCKS bit in the RLN24nmLiMD registers (when LIN 2.X is used).

Set the RLN24nGLWBR register when the OMM0 bit in the RLN24nmLiMST register of all channels in the same unit is 0 (LIN reset mode).

LWBR0 Bit (Wake-Up Baud Rate Selection Bit)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN24nGLWBR register to 0. This allows the 2.5-Tbit or longer low-level width of the input signal to be measured.
When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. With this setting, fa is selected as the LIN system clock (fLIN) during LIN wake-up mode regardless of the setting of the LCKS bit in the RLN24nmLiMD register (the LCKS bit is not changed) and the 2.5-Tbit or longer low-level width of the input signal can be measured.
Setting the baud rate to 19200 bps while fa is selected allows the 130 µs or longer low-level width of the input signal to be detected during LIN wake-up mode regardless of the setting of the LCKS bit in the RLN24nmLiMD register.

17.3.2.2 RLN24nGLBRP0 — LIN Baud Rate Prescaler 0 Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nGLBRP0: <RLIN24n_base> + 02_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP0[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.13 RLN24nGLBRP0 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP0[7:0]	Assuming that the value set in this register is N (0 to 255), the baud rate prescaler divides the LIN communication clock source by N+1. Setting range: 00_H to FF_H

Set the RLN24nGLBRP0 register when all channels in the same unit are in LIN reset mode (the OMM0 bit in the RLN24nmLiMST register is 0).

The value set in this register is used to control the frequency of baud rate clock source fa, fb, and fc.

Assuming that the value set in this register is N, baud rate prescaler 0 divides the LIN communication clock source by N+1.

17.3.2.3 RLN24nGLBRP1 — LIN Baud Rate Prescaler 1 Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nGLBRP1: <RLIN24n_base> + 03_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP1[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.14 RLN24nGLBRP1 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP1[7:0]	Assuming that the value set in this register is M (0 to 255), the baud rate prescaler divides the LIN communication clock source by M+1. Setting range: 00_H to FF_H

Set the RLN24nGLBRP1 register when all channels in the same unit are in LIN reset mode (the OMM0 bit in the RLN24nmLiMST register is 0).

The value set in this register is used to control the frequency of baud rate clock source fd.

Assuming that the value set in this register is M, baud rate prescaler 1 divides the LIN communication clock source by M+1.

17.3.2.4 RLN24nGLSTC — LIN Self-Test Control Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nGLSTC: <RLIN24n_base> + 04_H

Value after reset: 00_H

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

Table 17.15 RLN24nGLSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	—	Writing $A7_H$, 58_H, and 01_H successively to the RLN24nGLSTC register places the module in LIN self-test mode.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN self-test mode. 1: The module is in LIN self-test mode.

The RLN24nGLSTC register cancels protection of LIN self-test mode.

Set the RLN24nGLSTC register when all channels in the same unit are in LIN reset mode (the OMM0 bit in the RLN24nmLiMST register is 0).

Writing $A7_H$, 58_H, and 01_H successively to the RLN24nGLSTC register places the module in LIN self-test mode.

When successive writing is completed and the module is placed in LIN self-test mode, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For details about transitioning to LIN self-test mode, see **Section 17.15, LIN Self-Test Mode**.

When read, bits 6 to 1 return "000000_B", and bit 7 returns an undefined value.

LSTM Bit (LIN Self-Test Mode Bit)

When transitioning to LIN self-test mode, this bit is set to 1.
For details about exiting LIN self-test mode, see **Section 17.15, LIN Self-Test Mode**.
Writing "1" to this bit with other method than successive writing of $A7_H$, 58_H, and 01_H to the RLN24nGLSTC register does not alter the value.

17.3.3 Channel Registers

17.3.3.1 RLN24nmLiMD — LIN Mode Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiMD: <RLIN24n_base> + 08_H + i × 20_H

Value after reset: 00_H

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

Table 17.16 RLN24nmLiMD Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	LCKS[1:0]	LIN System Clock Selection Bit b3 b2 0 0: fa (Clock generated by baud rate prescaler 0) 0 1: fb (1/2 clock generated by baud rate prescaler 0) 1 0: fc (1/8 clock generated by baud rate prescaler 0) 1 1: fd (1/2 clock generated by baud rate prescaler 1)
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN24nmLiMD register while the OMM0 bit in the RLN24nmLiMST register is 0 (LIN reset mode).

LCKS[1:0] Bits (LIN System Clock Selection Bit)

The LCKS bits select the clock to be input to the protocol controller.
With 00_B set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).
With 01_B set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).
With 10_B set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).
With 11_B set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).

When the LWBR0 bit in the RLN24nGLWBR register is 1 (when LIN 2.x is used) and the RLN24nmLiMST register is 01_H (LIN wake-up mode), regardless of the setting of this LCKS bit, fa is input to the protocol controller (LCKS bit is not changed).

17.3.3.2 RLN24nmLiBFC — LIN Break Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiBFC: <RLIN24n_base> + 09_H + i × 20_H

Value after reset: 00_H

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

Table 17.17 RLN24nmLiBFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	BDT[1:0]	Transmission Break Delimiter (High Level) Width Setting Bit b5 b4 0 0: 1 Tbit 0 1: 2 Tbits 1 0: 3 Tbits 1 1: 4 Tbits
3 to 0	BLT[3:0]	Transmission Break (Low Level) Width Setting Bit b3 b0 0 0 0 0: 13 Tbits 0 0 0 1: 14 Tbits 0 0 1 0: 15 Tbits : 1 1 1 0: 27 Tbits 1 1 1 1: 28 Tbits

Set the RLN24nmLiBFC register while the OMM0 bit in the RLN24nmLiMST register is 0 (LIN reset mode).

Some combinations of the set values result in the length of a frame exceeding the timeout time. Set an appropriate value.

BDT[1:0] Bits (Transmission Break Delimiter (High Level) Width Setting Bit)

Set the break delimiter (high level) width of transmission frame header part.
1 Tbit to 4 Tbits can be set.

BLT[3:0] Bits (Transmission Break (Low Level) Width Setting Bit)

Set the break (low level) width of transmission frame header part.
13 Tbits to 28 Tbits can be specified.

17.3.3.3 RLN24nmLiSC — LIN Space Configuration Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiSC: <RLIN24n_base> + $0A_H$ + i × 20_H

Value after reset: 00_H

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

Table 17.18 RLN24nmLiSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Setting Bit b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Inter-Byte Space (Header)/Response Space Setting Bit b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Set the RLN24nmLiSC register while the OMM0 bit in the RLN24nmLiMST register is 0 (LIN reset mode).

Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Set an appropriate value.

IBS[1:0] Bits (Inter-Byte Space Setting Bit)

These bits set the width of the inter-byte space of the transmission frame response part.
0 Tbit to 3 Tbits can be set.
These bits are enabled only during response transmission; they are disabled during response reception.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Setting Bit)

These bits set the widths of the inter-byte space (header) of the transmission frame header field and the response space.
0 Tbit to 7 Tbits can be set.
The response space setting is enabled only during response transmission; the setting is disabled during response reception.
The value of the inter-byte space (header) value is equal to the response space value.

17.3.3.4 RLN24nmLiWUP — LIN Wake-Up Configuration Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiWUP: <RLIN24n_base> + $0B_H$ + i × 20_H

Value after reset: 00_H

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

Table 17.19 RLN24nmLiWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low Level Width Setting Bit b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN24nmLiWUP register while the OMM0 bit in the RLN24nmLiMST register is 0 (LIN reset mode).

WUTL[3:0] Bits (Wake-Up Transmission Low Level Width Setting Bit)

These bits set the low level width when the wake-up signal is transmitted.
1 Tbit to 16 Tbits can be set.
When the LWBR0 bit in the RLN24nGLWBR register is 1 (when LIN 2.x is used), regardless of the LCKS bit in the RLN24nmLiMD register, fa is selected as the LIN system clock (fLIN) (LCKS bit is not changed).

17.3.3.5 RLN24nmLiIE — LIN Interrupt Enable Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiIE: <RLIN24n_base> + $0C_H$ + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 17.20 RLN24nmLiIE Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ERRIE	Error Detection Interrupt Request Enable Bit 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Frame/Wake-up Reception Completion Interrupt Request Enable Bit 0: Disables frame/wake-up reception completion Interrupt request. 1: Enables frame/wake-up reception completion Interrupt request.
0	FTCIE	Frame/Wake-up Transmission Completion Interrupt Request Enable Bit 0: Disables frame/wake-up transmission completion interrupt request. 1: Enables frame/wake-up transmission completion interrupt request.

Set the RLN24nmLiIE register while the OMM0 bit in the RLN24nmLiMST register is 0 (LIN reset mode).

ERRIE Bit (Error Detection Interrupt Request Enable Bit)

The ERRIE bit enables or disables interrupt request upon detection of an error.
When set to 0, the interrupt request is not generated when the ERR flag in the RLN24nmLiST register is set to 1.
When set to 1, the interrupt request is generated when the ERR flag in the RLN24nmLiST register is set to 1.
Errors that generate the interrupt request are bit errors, physical bus errors, frame timeout errors, framing errors, and checksum errors.
Detection of a bit error, physical bus error, frame timeout error, and framing error can be enabled or disabled using the RLN24nmLiEDE register.

FRCIE Bit (Frame/Wake-up Reception Completion Interrupt Request Enable Bit)

The FRCIE bit enables or disables interrupt request upon completion of frame reception or wake-up signal reception (input signal low level width count).
When set to 0, the interrupt request is not generated when the FRC flag in the RLN24nmLiST register is set to 1.
When set to 1, the interrupt request is generated when the FRC flag in the RLN24nmLiST register is set to 1.

FTCIE Bit (Frame/Wake-up Transmission Completion Interrupt Request Enable Bit)

The FTCIE bit enables or disables interrupt request upon completion of frame transmission or wake-up signal transmission.
When set to 0, the interrupt request is not generated when the FTC flag in the RLN24nmLiST register is set to 1.
When set to 1, the interrupt request is generated when the FTC flag in the RLN24nmLiST register is set to 1.

17.3.3.6 RLN24nmLiEDE — LIN Error Detection Enable Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiEDE: <RLIN24n_base> + $0D_H$ + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FERE	FTERE	PBERE	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 17.21 RLN24nmLiEDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable Bit 0: Disables framing error detection 1: Enables framing error detection
2	FTERE	Frame Timeout Error Detection Enable Bit 0: Disables frame timeout error detection 1: Enables frame timeout error detection
1	PBERE	Physical Bus Error Detection Enable Bit 0: Disables physical bus error detection 1: Enables physical bus error detection
0	BERE	Bit Error Detection Enable Bit 0: Disables bit error detection 1: Enables bit error detection

Set the RLN24nmLiEDE register while the OMM0 bit in the RLN24nmLiMST register is 0_B (LIN reset mode).

FERE Bit (Framing Error Detection Enable Bit)

The FERE bit enables or disables detection of the framing error.
When set to 0, the framing error is not detected.
When set to 1, the framing error is detected.
When this bit is set to 1, the detection result is reflected to the FER flag in the RLN24nmLiEST register.
For details on the framing error, see **Section 17.14, Error Status**.

FTERE Bit (Frame Timeout Error Detection Enable Bit)

The FTERE bit enables or disables detection of the frame timeout error.
When set to 0, the frame timeout error is not detected.

When set to 1, the frame timeout error is detected.
When this bit is set to 1, the detection result is reflected to the FTER flag in the RLN24nmLiEST register.
For details on the frame timeout error, see **Section 17.14, Error Status**.

PBERE Bit (Physical Bus Error Detection Enable Bit)

The PBERE bit enables or disables detection of the physical bus error.
When set to 0, the physical bus error is not detected.
When set to 1, the physical bus error is detected.
When this bit is set to 1, the detection result is reflected to the PBER flag in the RLN24nmLiEST register.
For details on the physical bus error, see **Section 17.14, Error Status**.

BERE Bit (Bit Error Detection Enable Bit)

The BERE bit enables or disables detection of the bit error.
When set to 0, the bit error is not detected.
When set to 1, the bit error is detected.
When this bit is set to 1, the detection result is reflected to the BER flag in the RLN24nmLiEST register.
For details on the bit error, see **Section 17.14, Error Status**.

17.3.3.7 RLN24nmLiCUC — LIN Control Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiCUC: <RLIN24n_base> + $0E_H$ + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 17.22 RLN24nmLiCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode. 1: LIN operation mode.
0	OM0	LIN Reset 0: Transition to LIN reset mode. 1: Exit LIN reset mode.

When de-asserting the LIN reset mode, set 01_H to the RLN24nmLiCUC register in order to transition to LIN wake-up mode, 03_H in order to transition to LIN operation mode.

In LIN self-test mode, set 03_H to the RLN24nmLiCUC register after transitioning to LIN self-test mode.

After writing a value to this register, confirm that the value written is actually reflected to the RLN24nmLiMST register before writing another value.

OM1 Bit (LIN Mode Selection Bit)

The OM1 bit selects the operating mode (LIN wake-up mode or LIN operation mode) that is entered after exiting LIN reset mode.
Setting to 0 makes a transition to LIN wake-up mode.
Setting to 1 makes a transition to LIN operation mode.
This bit is enabled only when the OMM0 bit in the RLN24nmLiMST register is 1.
Writing a value to this bit is disabled while the FTS bit in the RLN24nmLiTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects whether to transition to or exit LIN reset mode.
Setting to 0 makes a transition to causes RLIN2 to enter LIN reset mode.
Setting to 1 de-asserts causes RLIN2 to exit LIN reset mode.

17.3.3.8 RLN24nmLiTRC — LIN Transmission Control Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiTRC: <RLIN24n_base> + 10_H + i × 20_H

Value after reset: 00_H

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

Table 17.23 RLN24nmLiTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	Response Transmission Start Bit 0: Response transmission is stopped in frame separate mode. 1: Response transmission is started in frame separate mode.
0	FTS	Frame Transmission/Wake-up Transmission/Reception Start Bit 0: Frame Transmission/wake-up transmission/reception is stopped. 1: Frame Transmission/wake-up transmission/reception is started.

RTS Bit (Response Transmission Start Bit)

Set the RTS bit to 1 in frame separate mode after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame transmission and transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 02_H using the store instruction.

Writing a value to this bit is disabled when the OMM0 bit is 0 (LIN reset mode). When the OMM1 bit is 0 (LIN wake-up mode), do not write 1.

Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission/wake-up transmission/ reception is is stopped).

FTS Bit (Frame Transmission/Wake-Up Transmission/Reception Start Bit)

Set the FTS bit to 1 to start frame/wake-up transmission.

Also set this bit to 1 to allow wake-up reception (counting of the input signal low level width).

Only 1 can be written to this bit; 0 cannot be written.

Writing a value to this bit is disabled when the OMM0 bit is 0 (LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication and transition to LIN reset mode.

17.3.3.9 RLN24nmLiMST — LIN Mode Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: RLN24nmLiMST: <RLIN24n_base> + 11_H + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 17.24 RLN24nmLiMST Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.
When the OMM0 bit is 0_B (LIN reset mode), the value of this bit is invalid.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

17.3.3.10 RLN24nmLiST — LIN Status Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiST: <RLIN24n_base> + 12_H + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 17.25 RLN24nmLiST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Header Transmission Completion Flag 0: Header transmission has not been completed. 1: Header transmission has been completed.
6	D1RC	Data 1 Reception Completion Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Frame/Wake-up Reception Completion Flag 0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed.
0	FTC	Frame/Wake-up Transmission Completion Flag 0: Transmission has not been completed. 1: Frame or wake-up transmission has been completed.

The RLN24nmLiST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (the value of the FTS bit of the RLN24nmLiTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

Do not write to this register while the FTS bit in the RLN24nmLiTRC register is 1 (frame transmission/ wake-up transmission/reception is started)

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Header Transmission Completion Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the previous value from before 1 was written.
“1” is set upon completion of header transmission, but no interrupt request is generated.
To clear the bit to 0 before the next communication is started (when the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit while in LIN operation mode.

D1RC Flag (Data 1 Reception Completion Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the previous value from before 1 was written.
“1” is set upon completion of Data 1 reception, but no interrupt request is generated.
To clear the bit to 0 before the next communication (the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit in LIN operation mode.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (the value of any of the flags of the RLN24nmLiEST registers is 1). Then, an interrupt request is generated if the ERRIE bit in the RLN24nmLiIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN24nmLiEST register in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

FRC Flag (Frame/Wake-up Reception Completion Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the previous value from before 1 was written.
The FRC flag is set to 1 upon completion of frame or wake-up reception. Then, an interrupt request is generated if the FRCIE bit in the RLN24nmLiIE register is 1 (interrupt is enabled).
To clear the bit to 0 before the next communication (the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

FTC Flag (Frame/Wake-up Transmission Completion Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the previous value from before 1 was written.
The FTC flag is set to 1 upon completion of frame or wake-up transmission. Then, an interrupt request is generated if the FTCIE bit in the RLN24nmLiIE register is 1 (interrupt is enabled).
To clear the bit to 0 before the next communication (the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

17.3.3.11 RLN24nmLiEST — LIN Error Status Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiEST: <RLIN24n_base> + 13_H + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	CSER	—	FER	FTER	PBER	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 17.26 RLN24nmLiEST Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	FTER	Frame Timeout Error Flag 0: Frame timeout error has not been detected. 1: Frame timeout error has been detected.
1	PBER	Physical Bus Error Flag 0: Physical bus error has not been detected. 1: Physical bus error has been detected.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN24nmLiEST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (the value of the FTS bit of the RLN24nmLiTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

When the FTS bit in the RLN24nmLiTRC register is 1 (frame transmission/wake-up transmission/ reception is started), do not write a value to this register.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the previous value from before 1 was written.
The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0 before the next communication (the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit in LIN operation mode.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the previous value from before 1 was written.
The FER flag is set to 1 upon framing error detection when the FERE bit of the RLN24nmLiEST register is 1 (framing error detection enabled). To clear the bit to 0 before the next communication (the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit in LIN operation mode.

FTER Flag (Frame Timeout Error Flag)

Only 0 can be written to the FTER flag; when 1 is written, the bit retains the previous value from before 1 was written.
The FTER flag is set to 1 upon frame timeout error detection when the FTERE bit of the RLN24nmLiEDE register is 1 (frame timeout error detection enabled). To clear the bit to 0 before the next communication (the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit in LIN operation mode.

PBER Flag (Physical Bus Error Flag)

Only 0 can be written to the PBER flag; when 1 is written, the bit retains the previous value from before 1 was written.
The PBER flag is set to 1 upon physical bus error detection when the PBERE bit of the RLN24nmLiEDE register is 1 (physical bus error detection enabled). To clear the bit to 0 before the next communication (the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the previous value from before 1 was written.
The BER flag is set to 1 upon bit error detection when the BERE bit of the RLN24nmLiEDE register is 1 (bit error detection enabled). To clear the bit to 0 before the next communication (the value of the FTS bit of the RLN24nmLiTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

17.3.3.12 RLN24nmLiDFC — LIN Data Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiDFC: <RLIN24n_base> + 14_H + i × 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	FSM	CSM	RFT	RFDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.27 RLN24nmLiDFC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	FSM	Frame Separate Mode Selection Bit 0: Frame separate mode is not set. 1: Frame separate mode is set.
5	CSM	Checksum Selection Bit 0: Classic 1: Enhanced
4	RFT	Response Field Communication Direction Selection Bit 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Setting Bit b3 b0 0 0 0 0: 0 bytes + checksum 0 0 0 1: 1 byte + checksum 0 0 1 0: 2 bytes + checksum : 0 1 1 1: 7 bytes + checksum 1 0 0 0: 8 bytes + checksum Settings other than the above are prohibited.

Set the RLN24nmLiDFC register when the FTS bit in the RLN24nmLiTRC register is 0 (frame transmission/wake-up transmission/reception is halted).

FSM Bit (Frame Separate Mode Selection Bit)

The FSM bit sets the response transmission mode.

When set to 0, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the RLN24nmLiTRC register is 1), response is transmitted/received without setting the RTS bit in the RLN24nmLiTRC register.
When set to 1, frame separate mode is selected. When the RTS bit of the RLN24nmLiTRC register is set to 1 during header transmission, response transmission is executed after header transmission has completed.
For response reception (the RFT bit is 0), set this bit to 0.
When transitioning to LIN self-test mode, set this bit to 0 before transition.
For details on frame separate mode, see **Section 17.11.1, Transmission of LIN Frames**.

CSM Bit (Checksum Selection Bit)

The CSM bit sets checksum mode.
When set to 0, classic checksum mode is selected.

When set to 1, enhanced checksum mode is selected.
When frame timeout error detection is enabled (the FTERE bit in the RLN24nmLiEDE register is 1), the timeout time differs depending on the setting of this bit. For details, see **Section 17.14, Error Status**.

RFT Bit (Response Field Communication Direction Selection Bit)

The RFT bit sets the direction of the response field/wake-up signal communication.
When set to 0, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low level width count).
When set to 1, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.

RFDL[3:0] Bits (Response Field Length Setting Bit)

The RFDL bits set the length of data in the response field.
The data length can be 0 to 8 bytes excluding the checksum size.

17.3.3.13 RLN24nmLiIDB — LIN ID Buffer Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiIDB: <RLIN24n_base> + 15_H + i × 20_H

Value after reset: Undefined

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0	ID[5:0]					
Value after reset	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17.28 RLN24nmLiIDB Register Contents

Bit Position	Bit Name	Function
7	IDP1	Parity Setting Bit (P1) Sets the parity bit (P1) to be transmitted in the ID field.
6	IDP0	Parity Setting Bit (P0) Sets the parity bit (P0) to be transmitted in the ID field.
5 to 0	ID[5:0]	ID Setting Bit Sets the 6-bit ID value to be transmitted in the ID field.

Set this register when the FTS bit in the RLN24nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

In LIN self-test mode, the operation is as follows:

Write the value to be transmitted prior to communication. After completion of frame transmission/ reception (after loopback), the reversed value of the received value can be read.

For details about LIN self-test mode, see **Section 17.15, LIN Self-Test Mode**.

IDP Bits (Parity Setting)

The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame. IDP0 sets P0 and IDP1 sets P1.
Since parity is not automatically calculated, set the calculated value. Note that even if the specified calculation result is incorrect, it is transmitted as is.

ID Bits (ID Setting)

Set the 6-bit ID to be transmitted in the ID field of the LIN frame.

17.3.3.14 RLN24nmLiCBR — LIN Checksum Buffer Register

Access: This register is a read-only register that can be read in 8-bit units. In LIN self-test mode, this register can be read or written in 8-bit units.

Address: RLN24nmLiCBR: <RLIN24n_base> + 16_H + i × 20_H

Value after reset: Undefined

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

Table 17.29 RLN24nmLiCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the checksum value transmitted or received.

In LIN operation mode, this register operates as follows:

- When the RFT bit in the RLN24nmLiDFC register is 1 (transmission):
  The value transmitted can be read. Read the value after transmission is completed.
  Writing to this register is invalid.
- When the RFT bit in the RLN24nmLiDFC register is 0 (reception):
  The value received can be read. Read the value after reception is completed.
  Writing to this register is invalid.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the RLN24nmLiDFC register is 1 (transmission):
  After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.
  Writing to this register is invalid.
- When the RFT bit in the RLN24nmLiDFC register is 0 (reception):
  Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about LIN self-test mode, see **Section 17.15, LIN Self-Test Mode**.

Set this register when the FTS bit in the RLN24nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

17.3.3.15 RLN24nmLiDBRb — LIN Data Buffer b Register

Access: This register can be read or written in 8-bit units.

Address: RLN24nmLiDBR1: <RLIN24n_base> + 18_H + i × 20_H
RLN24nmLiDBR2: <RLIN24n_base> + 19_H + i × 20_H
RLN24nmLiDBR3: <RLIN24n_base> + $1A_H$ + i × 20_H
RLN24nmLiDBR4: <RLIN24n_base> + $1B_H$ + i × 20_H
RLN24nmLiDBR5: <RLIN24n_base> + $1C_H$ + i × 20_H
RLN24nmLiDBR6: <RLIN24n_base> + $1D_H$ + i × 20_H
RLN24nmLiDBR7: <RLIN24n_base> + $1E_H$ + i × 20_H
RLN24nmLiDBR8: <RLIN24n_base> + $1F_H$ + i × 20_H

Value after reset: Undefined

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

Table 17.30 RLN24nmLiDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or reads the data to be received. Setting range: 00_H to FF_H

- For response transmission:

  These registers set the data to be transmitted in the response field.
  Use these registers with the following settings:

  – The RFT bit in RLN24nmLiDFC register is 1 (transmission).
  – The FSM bit in RLN24nmLiDFC register is 0 (not frame separate mode).
  – The FTS bit in RLN24nmLiTRC register is 0 (frame transmission/wake-up transmission/ reception is stopped).

  or

  – The RFT bit in RLN24nmLiDFC register is 1 (transmission).
  – The FSM bit in RLN24nmLiDFC register is 1 (frame separate mode).
  – The RTS bit in RLN24nmLiTRC register is 0 (response transmission is halted).

- For response reception:

  These registers hold the data received in the response field.
  The received data is overwritten. If an error is detected, the data prior to reception interruption is stored.
  Do not read these registers when the FTS bit is 1 (frame transmission/wake-up transmission/ reception is started)

In LIN self-test mode, the operation is as follows:

Write the value to be transmitted prior to communication. After completion of frame transmission/ reception (after loopback), the reversed value of the received value can be read.

For details about LIN self-test mode, see **Section 17.15, LIN Self-Test Mode**.

17.4 Interrupt Sources

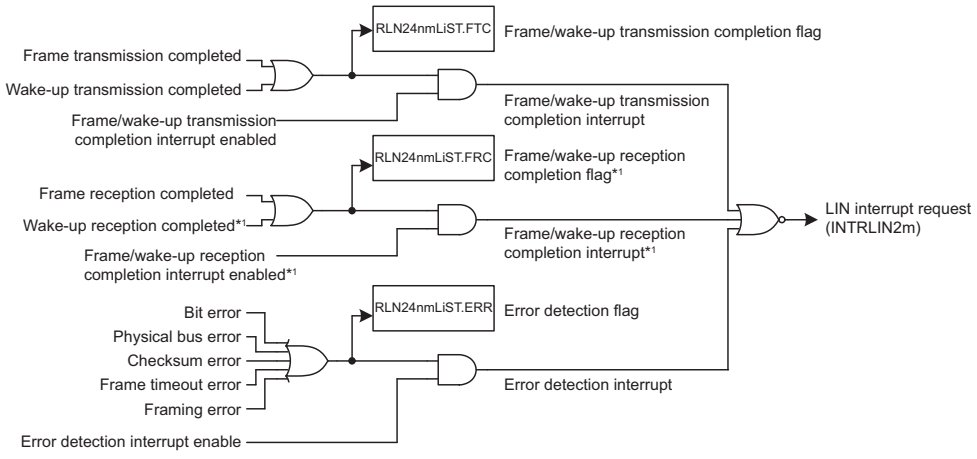
The LIN master interface generates LIN interrupt requests.

There are three interrupt sources for each channel; successful frame/wake-up transmission, successful frame/wakeup reception, and error detection.

Interrupt requests from these three sources are ORed to generate one interrupt request “LIN interrupt”.

The respective interrupt request is output when the corresponding flag in the RLN24nmLiST register is set to 1 while the corresponding bit in the RLN24nmLiIE register is 1 (interrupt enabled). However, if an interrupt is requested when the corresponding flag in the RLN24nmLiST register has been set to 1, it is ignored. Therefore, clear the corresponding flag to 0 to enable the interrupt again.

Figure 17.2 shows a block diagram of the LIN interrupt.



Note 1. The wake-up receive signal indicates the input signal low level width count.

Figure 17.2 LIN Interrupt Block Diagram

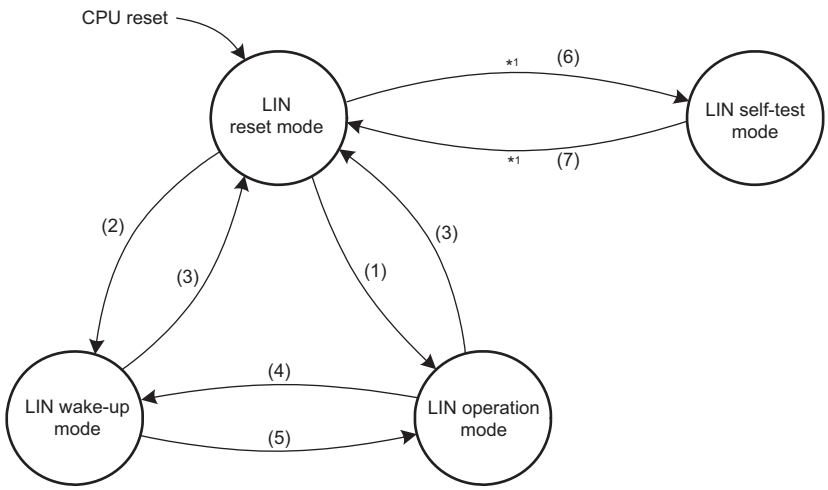
17.5 Modes

The LIN master interface provides the following four modes:

- LIN reset mode
- LIN operation mode
- LIN wake-up mode
- LIN self-test mode

The mode transitions except LIN self-test mode is controlled independently for respective channels.

Figure 17.3 shows mode transitions. **Table 17.31** describes mode transition conditions. **Table 17.32** lists operations available in each mode.



Note 1. See **Section 17.15, LIN Self-Test Mode** for the details.

Figure 17.3 Mode Transitions

Table 17.31 Transition Condition of Each Mode

	Transition Mode			Transition Condition
(1)	LIN reset mode	→	LIN operation mode	RLN24nmLiCUC.OM1,OM0 = 11_B
(2)	LIN reset mode	→	LIN wake-up mode	RLN24nmLiCUC.OM1,OM0 = 01_B
(3)	LIN wake-up mode LIN operation mode	→	LIN reset mode	RLN24nmLiCUC.OM0 = 0_B
(4)	LIN operation mode	→	LIN wake-up mode	RLN24nmLiCUC.OM1,OM0 = 01_B
(5)	LIN wake-up mode	→	LIN operation mode	RLN24nmLiCUC.OM1,OM0 = 11_B
(6)	LIN reset mode	→	LIN self-test mode	See **Section 17.15, LIN Self-Test Mode**.
(7)	LIN self-test mode	→	LIN reset mode	See **Section 17.15, LIN Self-Test Mode**.

Table 17.32 Operations Available in Each Mode

LIN Operation Mode	LIN Wake-Up Mode	LIN Self-Test Mode
Header transmission Response transmission Response reception Error detection	Wake-up transmission Wake-up reception Error detection	Self-test

Whether a transition has been made to LIN reset mode, LIN operation mode, or LIN wake-up mode can be verified by reading the OMM1 and OMM0 bits in the RLN24nmLiMST register.

For a description of LIN self-test mode, see **Section 17.15, LIN Self-Test Mode**.

17.6 LIN Reset Mode

Setting the OM0 bit in the RLN24nmLiCUC register to 0 (LIN reset mode) makes a transition to LIN reset mode. The transition to LIN reset mode can be verified by confirming that the OMM0 bit in the RLN24nmLiMST register has been set to 0 (LIN reset mode). In this mode, the LIN communication stops.

From LIN reset mode, transitions to LIN operation mode, LIN wake-up mode, and LIN self-test mode can be made.

When the mode transitions to LIN reset mode, the following registers are initialized to their reset values and retain their initial values while in LIN reset mode:

- RLN24nmLiTRC register
- RLN24nmLiST register
- RLN24nmLiEST register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN24nGLWBR register
- RLN24nGLBRP0 register
- RLN24nGLBRP1 register
- RLN24nmLiMD register
- RLN24nmLiBFC register
- RLN24nmLiSC register
- RLN24nmLiWUP register
- RLN24nmLiIE register
- RLN24nmLiEDE register
- RLN24nmLiDFC register
- RLN24nmLiIDB register
- RLN24nmLiCBR register
- RLN24nmLiDBRb register

17.7 LIN Operation Mode

In LIN operation mode, frame processing (header transmission, response transmission, response reception, and error detection) is performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN24nmLiCUC register to 11_B changes the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN24nmLiMST register to 11_B. Communication settings should be performed after the RLN24nmLiMST register has become 11_B.

17.8 LIN Wake-Up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) is performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN24nmLiCUC register to 01_B changes the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN24nmLiMST register to 01_B. Communication settings should be performed after the RLN24nmLiMST register has become 01_B.

17.9 Header Transmission/Response Transmission/Response Reception

17.9.1 Header Transmission

Figure 17.4 shows the operation in header transmission. **Table 17.33** shows the processing in header transmission.

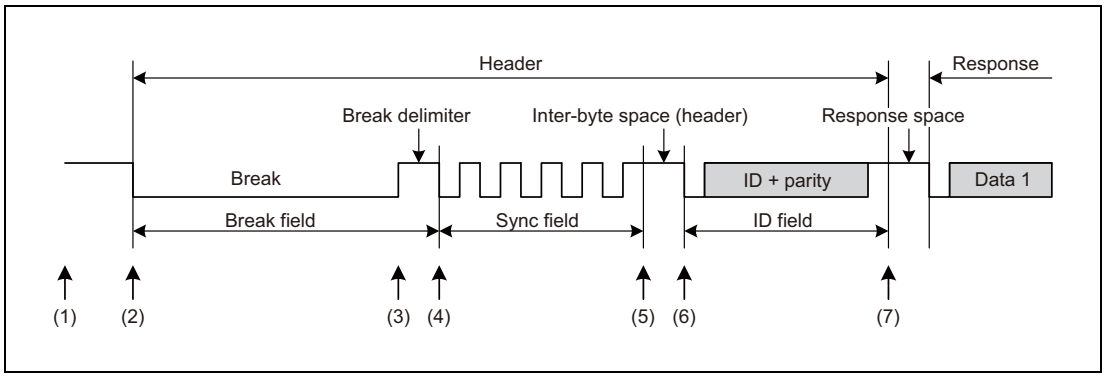


Figure 17.4 Operation in Header Transmission

Table 17.33 Processing in Header Transmission

	Software Processing	LIN Master Interface Processing
(1)	• Sets a baud rate • Enables interrupts • Enables error detection • Sets frame configuration parameters • Transitions to LIN operation mode • Sets information to be transmitted (ID, parity, data length, response direction, checksum method, and transmission data)	Waits for the setting of the FTS bit in the RLN24nmLiTRC register by software (idle).
(2)	Sets the FTS bit in the RLN24nmLiTRC register to 1 (frame transmission/wake-up transmission/reception started)	Transmits a break.
(3)	Waits for an interrupt request	Transmits a break delimiter.
(4)		Transmits a sync field (55H).
(5)		Transmits an inter-byte space (header).
(6)		Transmits an ID field.
(7)		Sets the successful header transmission flag.

NOTE

For information about error detection, see **Section 17.14, Error Status**.

17.9.2 Response Transmission

Figure 17.5 shows the operation of the LIN master interface in response transmission. **Table 17.34** shows the processing in response transmission.

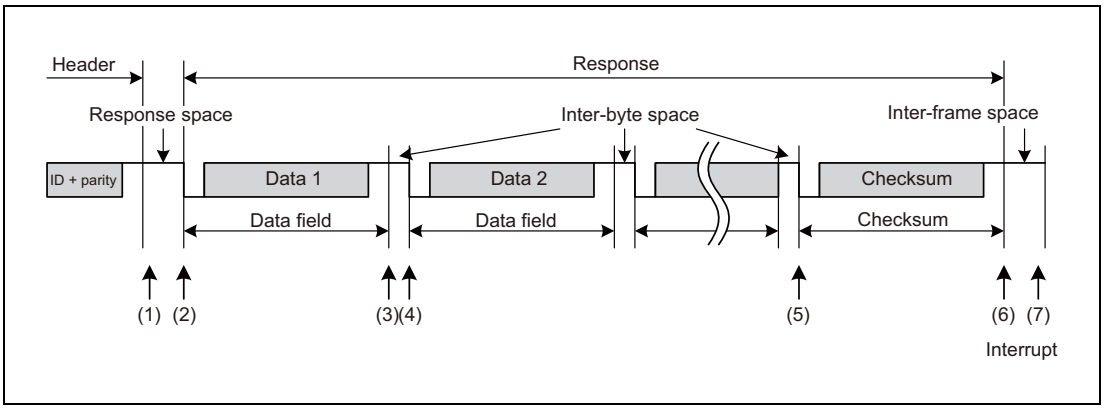


Figure 17.5 Operation in Response Transmission

Table 17.34 Processing in Response Transmission

	Software Processing	LIN Master Interface Processing
(1)	(When in frame separate mode) • Sets the RTS bit in the RLN24nmLiTRC register to 1 (response transmission started) (When not in frame separate mode) • Waits for an interrupt request	(When in frame separate mode) • Waits for the RTS bit in the RLN24nmLiTRC register to be set to 1 by software. (During this time, “1” is output.) • When the bit is set to 1, sends a response space. (When not in frame separate mode) • Sends a response space.
(2)	Waits for an interrupt request	Transmits data 1.
(3)		Transmits an inter-byte space.
(4)		• Transmits the data 2. • Transmits an inter-byte space • Transmits the data 3. • Transmits an inter-byte space (Repeats data and inter-byte spaces as many times as many times as the data length specified in bits RFDL[3:0] in the RLN24nmLiDFC register.) : :
(5)		Transmits the checksum.
(6)		• Sets the successful frame/wake-up transmission flag. • Sets the FTS bit in the RLN24nmLiTRC register to 0 (frame transmission or wake-up transmission/reception stopped) (When in frame separate mode) • Sets the RTS bit in the RLN24nmLiTRC register to 0 (response transmission stopped).
(7)	• Processing after communication Checks the RLN24nmLiST register, and clears flags.	Idle

NOTE

For information about error detection, see **Section 17.14, Error Status**.

17.9.3 Response Reception

Figure 17.6 shows the operation of the LIN master interface in response reception. **Table 17.35** shows the processing in response reception.

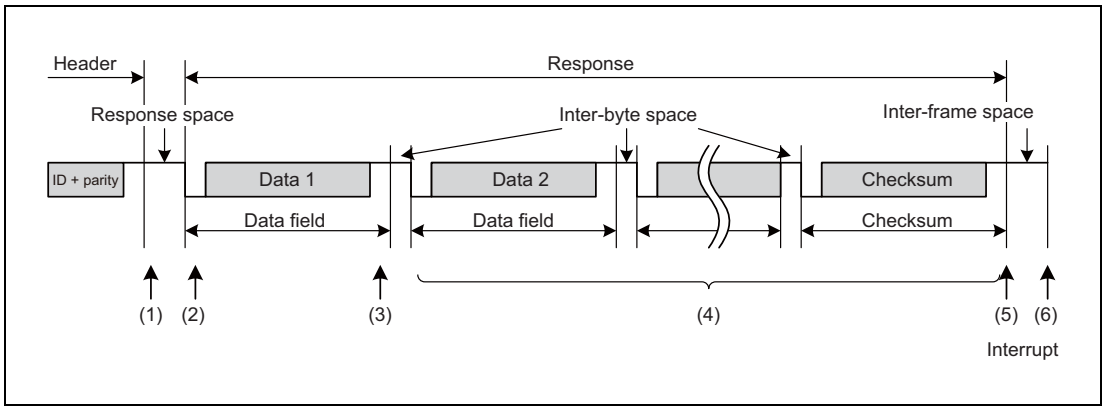


Figure 17.6 Operation in Response Reception

Table 17.35 Processing in Response Reception

	Software Processing	LIN Master Interface Processing
(1)	Waits for an interrupt request (no processing).	Waits for detection of a start bit.
(2)	Waits for an interrupt request.	Receives the data 1 when the start bit is detected.
(3)		Sets the Data 1 reception completion flag.
(4)		• Receives the data 2 when the start bit is detected. • Receives the data 3 when the start bit is detected. (Repeats the reception of data as many times as the data length specified in bits RFDL[3:0] in the RLN24nmLiDFC register.) : : • Receives the checksum when the start bit is detected.
(5)		• Evaluates the checksum. • Sets the successful frame/wake-up reception flag. • Sets the FTS bit in the RLN24nmLiTRC register to 0 (frame transmission or wake-up transmission/reception stopped).
(6)	• Processing after communication Reads the received data. Checks the RLN24nmLiST register, and clears flags.	Idle

NOTE

For information about error detection, see **Section 17.14, Error Status**.

17.10 Data Transmission/Reception

17.10.1 Data Transmission

One bit of data is transmitted per 1 Tbit.

The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data are compared bit by bit, and the results are stored in the BER flag in the RLN24nmLiEST register (see **Section 17.14, Error Status**).

In the LIN master interface, because data is generated every 1 Tbit = 16fLIN, the sampling point for received data is the 13th clock cycle (81.25% position).

Figure 17.7 shows an example of data transmission timing.

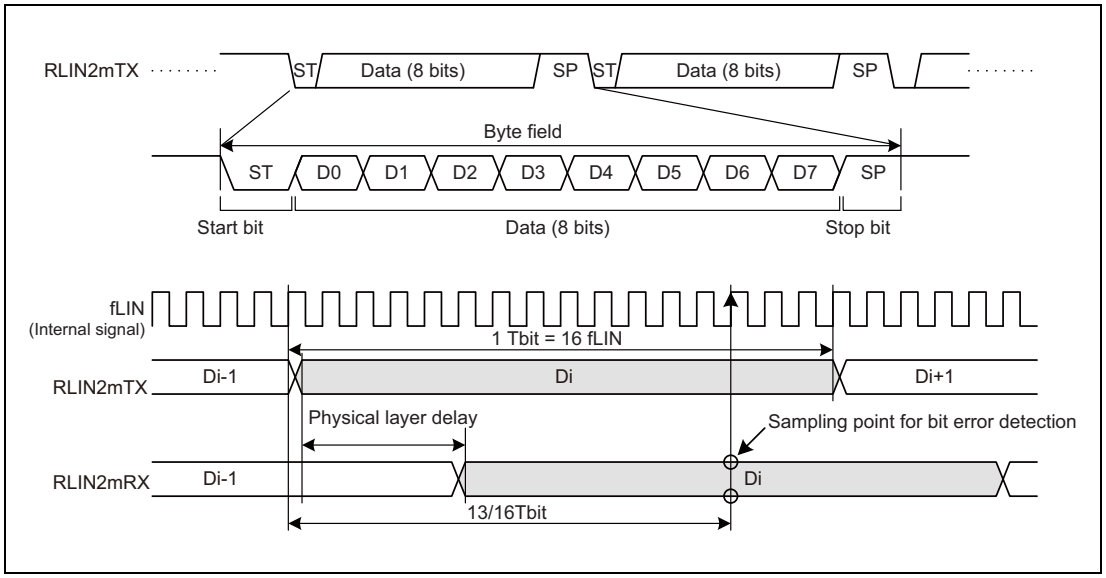


Figure 17.7 Example of Data Transmission Timing

17.10.2 Data Reception

Data reception is performed by using the synchronized RLIN2mRX signal (an internal signal) that is the input from the RLIN2mRX pin synchronized with the LIN system clock (fLIN).

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN2mRX signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized RLIN2mRX signal is low level. The falling edge is not recognized as a start bit if the RLIN2mRX signal reset is de-asserted is fixed to low level or if a high level is detected on re-sampling.

The bit sampling period after detection of the start bit is one Tbit.

Figure 17.8 shows an example of data reception timing.

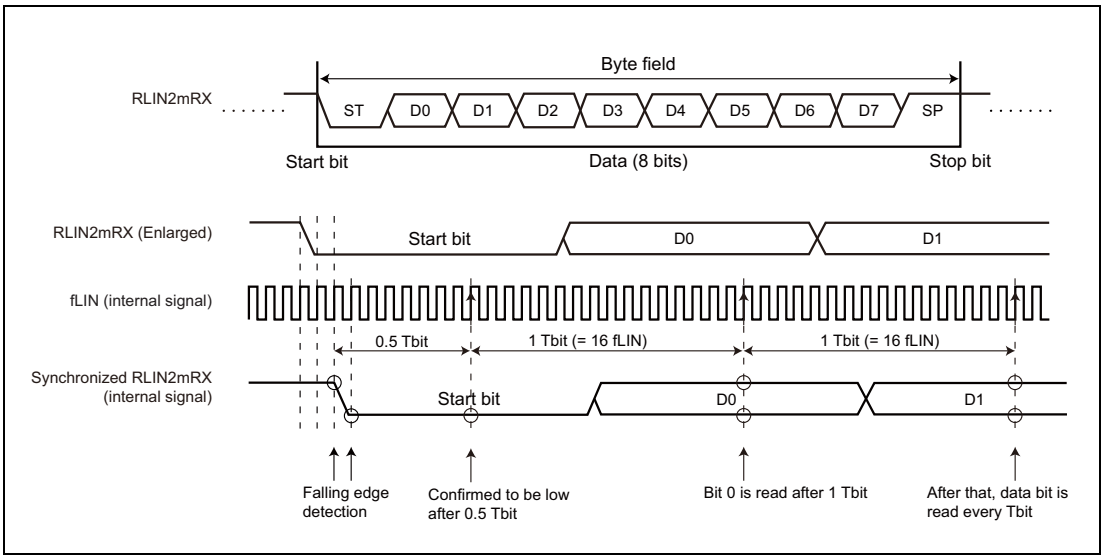


Figure 17.8 Example of Data Reception Timing

17.11 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN master interface sends or receives data continuously.

17.11.1 Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN24nmLiDBR1 to RLN24nmLiDBR8 are sequentially transmitted to data areas 1 to 8 of the LIN frame. In the case of a 4-byte transmission, the contents stored in registers RLN24nmLiDBR1 to RLN24nmLiDBR4 are transmitted to data areas 1 to 4 of the LIN frame, but the contents of registers RLN24nmLiDBR5 to RLN24nmLiDBR8 are not transmitted. The transmitted checksum data is stored in the RLN24nmLiCBR register.

Figure 17.9 shows the LIN transmission processing and the corresponding buffers.

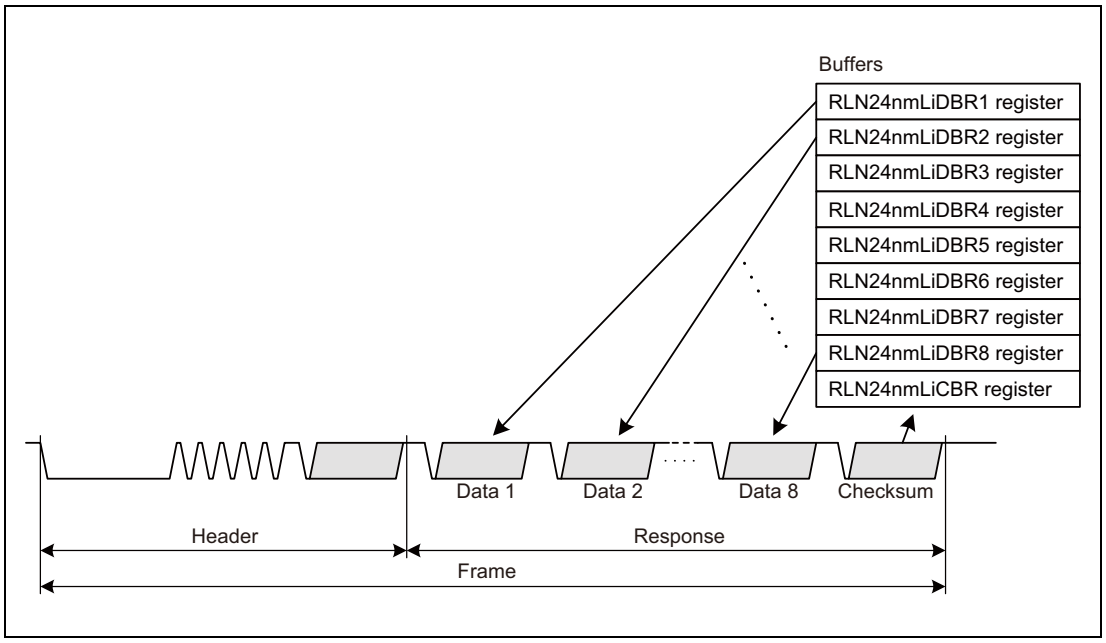


Figure 17.9 LIN Transmission Processing and Corresponding Buffers

(1) Frame Separate Mode

Setting the FSM bit in the RLN24nmLiDFC register to 1 sets the frame separate mode.

In frame separate mode, a header and a response are transmitted when prompted by separate transmission start requests.

When the transmission of a header is finished, the HTRC flag in the RLN24nmLiST register is set to 1 (successful header transmission).

17.11.2 Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame are stored in registers RLN24nmLiDBR1 to RLN24nmLiDBR8, respectively, upon receipt of a stop bit. In the case of a 4-byte reception, the contents of data areas 1 to 4 of the LIN frame are stored in registers RLN24nmLiDBR1 to RLN24nmLiDBR4, respectively; however, no data is stored in registers RLN24nmLiDBR5 to RLN24nmLiDBR8. The received checksum data is stored in the RLN24nmLiCBR register.

Figure 17.10 shows the LIN reception processing and the corresponding buffers.

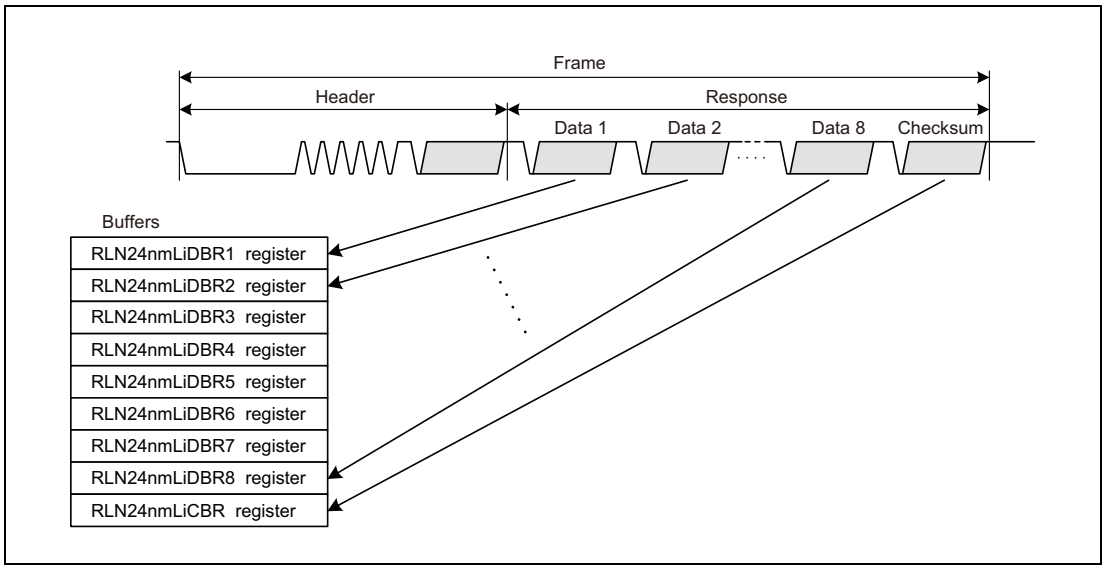


Figure 17.10 LIN Reception Processing and Corresponding Buffers

(1) Reception of Data 1

When the reception of the first byte of data is finished, the D1RC flag in the RLN24nmLiST register is set to 1 (successful data 1 reception).

17.12 Wake-Up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

17.12.1 Wake-Up Transmission

In LIN wake-up mode, setting the RFT bit in the RLN24nmLiDFC register to 1 (transmission) and the FTS bit in the RLN24nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low level width of the wake-up signal is set using the WUTL[3:0] bits in the RLN24nmLiWUP register. However, if the value of the LWBR0 bit of the RLN24nGLWBR register is 1 (when LIN 2.x is used), the LIN system clock (fLIN) has the low-level width of fa (JW) regardless of the setting of the LCKS bit of the RLN24nmLiMD register. By setting the baud rate to 19200 bps when fa is selected and setting the WUTL[3:0] bits of the RLN24nmLiWUP register to 0100_B (5Tbits), 260 µs low width can be output in LIN wake-up mode regardless of the setting of the LCKS bit of the RLN24nmLiMD register.

If a wake-up low level is output without any error, the FTC flag in the RLN24nmLiST register is set to 1 (successful frame or wake-up transmission); when the FTCIE bit in the RLN24nmLiIE register is 1 (successful frame/wake-up transmission interrupt enabled), an interrupt request is generated.

If an error is detected, wake-up transmission is aborted and the error flag for the error detected (the PBER flag or BER flag in the RLN24nmLiEST register) is set to 1 (physical bus error detection / bit error detection).

Figure 17.11 shows the wake-up transmission timing.

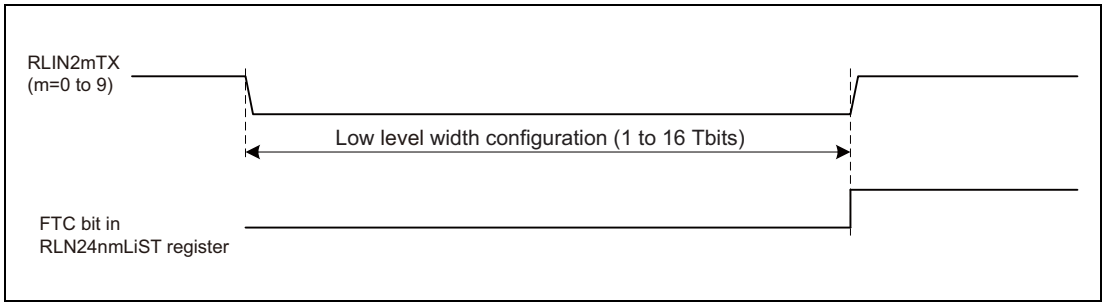


Figure 17.11 Wake-Up Transmission Timing

17.12.2 Wake-Up Reception

The detection of a wake-up signal involves the use of an input signal low level width count function.

The input signal low level width count function measures the low level width of the input signal to the RLIN2mRX pin, using the same sampling point as data reception. This allows an input signal of fLIN with a low-level width of 2.5-Tbit or longer to be measured.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN24nGLWBR register to 0. When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1.

When LWBR0 bit is set to 1, regardless of the setting of the LCKS bit in the RLN24nmLiMD register, fa is selected as the LIN system clock (fLIN) (the LCKS bit is not changed).

Setting the baud rate to 19200 bps while fa is selected allows an input signal with a low-level width of 130 µs or longer to be detected during LIN wake-up mode regardless of the setting of the LCKS bit in the RLN24nmLiMD register.

When using this function, in LIN wake-up mode set the RFT bit in the RLN24nmLiDFC register to 0 (reception), and the FTS bit in the RLN24nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started).

When the low level width to be measured is reached, the FRC flag in the RLN24nmLiST register is set to 1 (successful frame or wake-up reception). If the FRCIE bit in the RLN24nmLiIE register is 1 (successful frame or wake-up reception interrupt enabled), an interrupt request is generated.

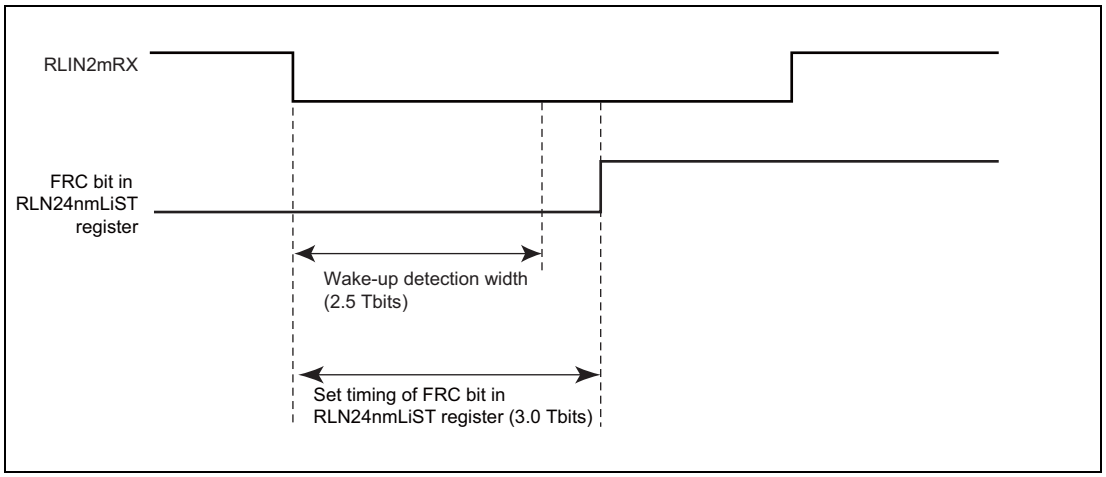


Figure 17.12 Input Signal Low Level Count Function

17.12.3 Wake-Up Collision

If the master node and the slave node transmit wake-up signals simultaneously, a collision will occur on the LIN bus; however the a collision of wake-up signals is not detected in the LIN master interface.

17.13 Status

The LIN master interface detects seven types of statuses.

Three of these statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, can generate interrupt requests.

Table 17.36 shows the types of statuses.

Table 17.36 Types of Statuses

Status	Status Set Condition	Status Clear Condition	Operation Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN24nmLiCUC register is set to not-LIN–reset-mode, if the LIN master interface actually exits LIN reset mode.	After the OM0 bit in the RLN24nmLiCUC register is set to LIN reset mode, if the LIN master interface enters LIN reset mode.	All modes	OMM0 bit in the RLN24nmLiMST register	—
Operation mode	After the OM1 bit in the RLN24nmLiCUC register is set to LIN operation mode, if the LIN master interface actually enters.	After the OM1 bit in the RLN24nmLiCUC register is set to LIN wake-up mode, if the LIN master interface enters LIN wake-up mode.	• LIN operation mode • LIN wake-up mode	OMM1 bit in the RLN24nmLiMST register	—
Successful frame/wake-up transmission	When a frame (header transmission + response transmission) or a wake-up signal is transmitted successfully.	• When the next communication is started • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FTC flag in the RLN24nmLiST register	√
Successful frame/wake-up reception	When a frame (header transmission + response reception) or a wake-up signal is received successfully.	• When the next communication is started • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FRC flag in the RLN24nmLiST register	√
Error detection	If any of the CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN24nmLiEST register is set to 1 (error detected).	• When the next communication is started • When cleared by software*1 • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	ERR flag in the RLN24nmLiST register	√
Successful data 1 reception	The RFT bit in the RLN24nmLiDFC register is 0 (reception) and the first byte of the response field is received successfully.*2	• When the next communication is started • When cleared by software • After transition to LIN reset mode	LIN operation mode	D1RC flag in the RLN24nmLiST register	—
Successful header transmission	When a header field is transmitted successfully.	• When the next communication is started • When cleared by software • After transition to LIN reset mode	LIN operation mode	HTRC flag in the RLN24nmLiST register	—

Note 1. In LIN operation mode, the ERR flag in the RLN24nmLiST register is cleared to 0 by writing 0 to the CSER flag, FER flag, FTER flag, PBER flag, BER flag in the RLN24nmLiEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN24nmLiDFC register are 0000_B (0 bytes + checksum).

17.14 Error Status

17.14.1 Types of Error Statuses

The LIN master interface can detect five types of error statuses in LIN master mode. The condition of these error statuses can be checked by means of the corresponding bits in the RLN24nmLiEST register.

All error statuses represent interrupt sources.

Table 17.37 shows the types of error statuses.

Table 17.37 Types of Error Statuses

Status	Error Detection Condition	Operating Mode Capable of Error Detection	Communication	Selection of Detection Enable/ Disable	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match *1	• LIN operation mode • LIN wake-up mode	Aborted	√	BER flag in the RLN24nmLiEST register
Physical bus error	• LIN bus detects high level when sending a break • LIN bus detects low level when sending a break delimiter • LIN bus detects high level when sending a wake-up	• LIN operation mode • LIN wake-up mode	Aborted	√	PBER flag in the RLN24nmLiEST register
Frame timeout error	A frame transmission/reception does not complete within a given time*2	LIN operation mode	Aborted	√	FTER flag in the RLN24nmLiEST register
Framing error	In response field reception, a stop bit of each data byte is low level	LIN operation mode	Aborted	√	FER flag in the RLN24nmLiEST register
Checksum error	In response field reception, the result of checksum test gives an error	LIN operation mode	—	×	CSER flag in the RLN24nmLiEST register

Note 1. If a bit error is detected, the process is aborted after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is aborted immediately. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is aborted immediately after the bit that caused the error is transmitted.

Note 2. The timeout time depends on the response field data length (the RFDL[3:0] bits in the RLN24nmLiDFC register) and the checksum selection (the CSM bit in the RLN24nmLiDFC register), and can be calculated according to the following formula:
When classic checksum is selected (when the CSM bit in the RLN24nmLiDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
When enhanced checksum is selected (when the CSM bit in the RLN24nmLiDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is longer than the TFRAME_MAX of LIN Specification Package Revision 1.3 when classic checksum is selected, or the TFRAME_MAX of LIN Specification Package Revision 2.x when enhanced checksum is selected.

The error status is cleared when the next communication is started, by software, or at a transition to LIN reset mode.

17.14.2 Target Time Domain for Error Detection

Figure 17.13 shows the time domain in which the LIN master interface monitors for error detection.

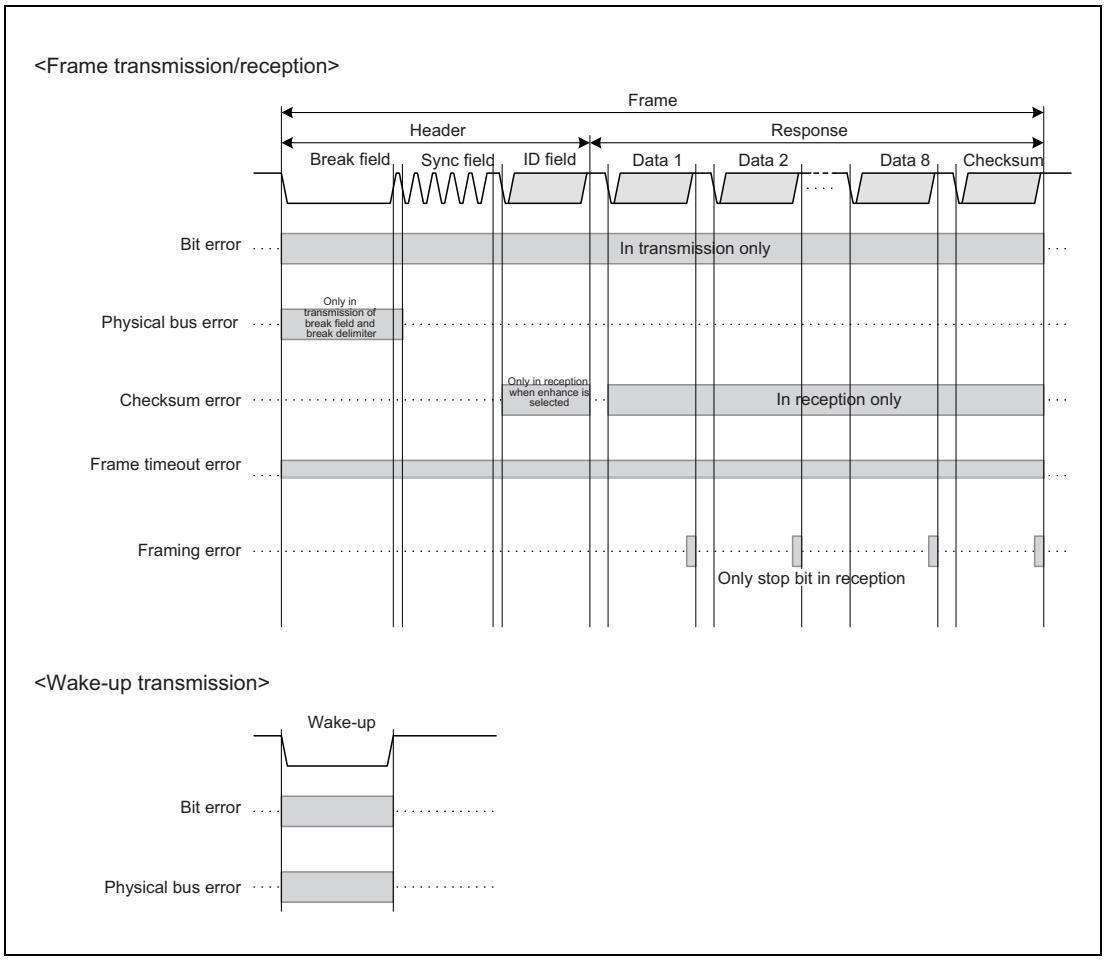


Figure 17.13 Target Time Domain for Error Detection

17.15 LIN Self-Test Mode

The LIN master interface provides LIN self-test mode. When the LIN master interface enters LIN self-test mode, RLIN2mTX and RLIN2mRX are disconnected from the external pins, and are internally connected in the LIN master interface. Thus, the frame transmitted from RLIN2mTX is looped back to RLIN2mRX.

Two types of self-test can be performed:

- LIN self-test mode (transmission): header transmission and response transmission
- LIN self-test mode (reception): header transmission and response reception

In LIN self-test mode, the operation is performed at the fastest baud rate, regardless of the setting of the baud rate generator.

Regardless of the setting of the baud rate related registers, the baud rate operates at the LIN communication clock source/16 [bps].

In LIN self-test mode, the following functions are not supported:

- LIN wake-up mode
- Frame separate mode

Do not use these functions.

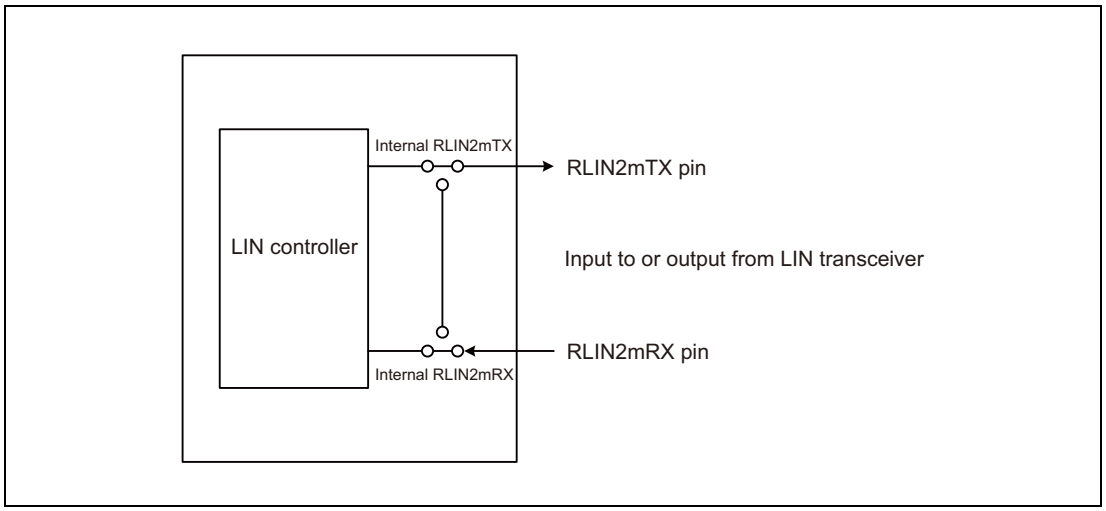


Figure 17.14 Connection in LIN Reset Mode, LIN Wake-Up Mode, and LIN Operation Mode

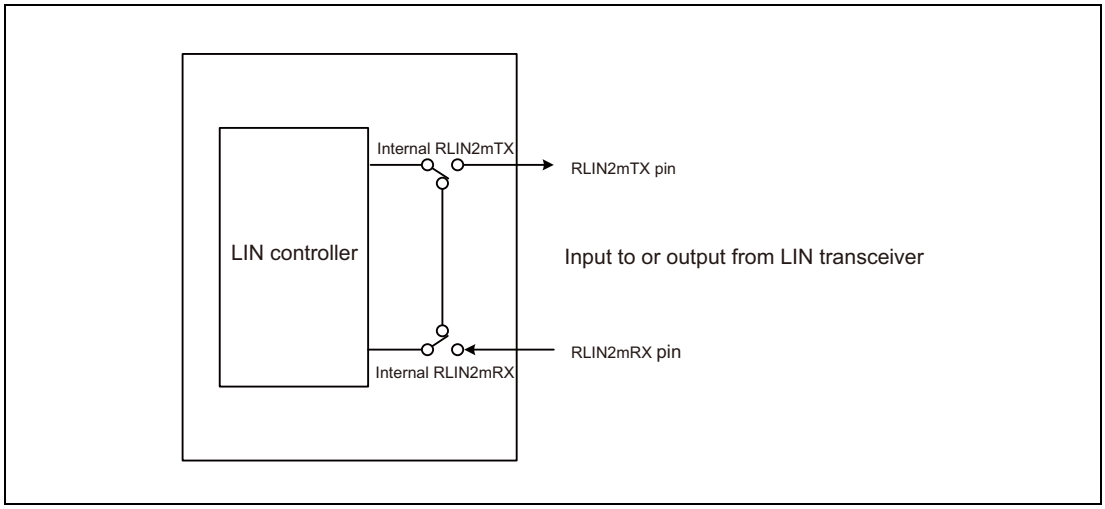


Figure 17.15 Connection in LIN Self-Test Mode

17.15.1 Transition to LIN Self-Test Mode

Writing to the RLN24nGLSTC register enables LIN self-test mode.

The LSTM bit in the RLN24nGLSTC register being set to 1 indicates that the mode has transitioned.

A specific sequence is required to transition to LIN self-test mode. In this sequence, information must be written three times consecutively to the LIN self-test control register, as follows:

- Switch all channels of the unit to LIN reset mode.
  Set the OM0 bit in the RLN24nmLiCUC register to 0 (LIN reset mode).
  Read the OMM0 bit in the RLN24nmLiMST register and confirm that it is 0 (LIN reset mode).
- 1st write: RLN24nGLSTC register = $1010\ 0111_B$ ($A7_H$)
- 2nd write: RLN24nGLSTC register = $0101\ 1000_B$ (58_H)
- 3rd write: RLN24nGLSTC register = $0000\ 0001_B$ (01_H)
- Confirm that all channels have transitioned to LIN self-test mode
  Read the LSTM bit in the RLN24nGLSTC register and confirm that it is 1 (LIN self-test mode).

If the key of the first write ($A7_H$) is written twice by mistake, the transition to LIN self-test mode is suspended. The above sequence should be retried from the 1st write step. In addition, if a write to another LIN-related register in the same unit is performed during transition to LIN self-test mode (three consecutive write operations to the RLN24nGLSTC register), the transition is also suspended.

17.15.2 Transmission in LIN Self-Test Mode

To execute a self-test on transmission, perform the procedure below:

- Set the baud rate related registers.
  RLN24nGLBRP0 register = xxxx $xxxx_B$ *1
  RLN24nGLBRP1 register = xxxx $xxxx_B$ *1
  RLN24nmLiMD register = 0000 $xx00_B$ *1
- Set interrupt enable register and error enable related registers.
  RLN24nmLiIE register = 0000 $0xxx_B$ *2
  RLN24nmLiEDE register = 0000 $xxxx_B$
- Set the break field and space related registers.
  RLN24nmLiBFC register = 00xx $xxxx_B$
  RLN24nmLiSC register = 00xx $0xxx_B$
- Exit the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN24nmLiCUC register, and check that the OMM1 and OMM0 bits in the RLN24nmLiMST register are 11_B.
- Set the transmit frame related registers.
  RLN24nmLiDFC register = 00x1 $xxxx_B$
  RLN24nmLiIDB register = xxxx $xxxx_B$
  RLN24nmLiDBR1 to RLN24nmLiDBR8 registers = xxxx $xxxx_B$
- Start header transmission to response transmission
  Set the FTS bit in the RLN24nmLiTRC register to 1 (frame transmission/wake-up transmission/ reception started).
  The LIN self-test mode (transmission) executed, interrupts are generated, and status and error status are also updated. The checksum is automatically calculated by the LIN master interface.
  To suspend the LIN self-test mode (transmission) while it is running, set the OM0 bit in the RLN24nmLiCUC register to 0 (LIN reset mode), which causes a transition to the LIN reset mode.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN24nmLiIDB, RLN24nmLiDBRb, and RLN24nmLiCBR registers (the data is reversed before being stored because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN24nmLiTRC register is cleared.
- If the transmission fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN24nmLiTRC register is cleared.

NOTE

x: Don't care

Note 1. The settings of the following registers are not reflected in the operation of the LIN self-test mode:
the RLN24nGLBRP0 register, the RLN24nGLBRP1 register, and the LCKS bit in the RLN24nmLiMD register. Therefore, setting these registers is not necessary.

Note 2. If necessary, set the related registers described in **Section 7, Exceptions and Interrupts**.

17.15.3 Reception in LIN Self-Test Mode

To execute a self-test on reception, perform the procedure below:

- Set the baud rate related registers.
  RLN24nGLBRP0 register = xxxx $xxxx_B$*1
  RLN24nGLBRP1 register = xxxx $xxxx_B$*1
  RLN24nmLiMD register = 0000 $xx00_B$*1
- Set the interrupt enable and error enable related registers.
  RLN24nmLiIE register = 0000 $0xxx_B$*2
  RLN24nmLiEDE register = 0000 $x0xx_B$
- Set the break field and space related registers.
  RLN24nmLiBFC register = 00xx $xxxx_B$
  RLN24nmLiSC register = 00xx $0xxx_B$*1
- Exit the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN24nmLiCUC register, and check that the OMM1 and OMM0 bits in the RLN24nmLiMST register are 11_B.
- Set the receive frame related registers.
  RLN24nmLiDFC register = 00x0 $xxxx_B$
  RLN24nmLiIDB register = xxxx $xxxx_B$
  RLN24nmLiDBR1 to RLN24nmLiDBR8 registers = xxxx $xxxx_B$
  RLN24nmLiCBR register = xxxx $xxxx_B$
  Since the checksum value to be transmitted is not automatically calculated, perform the calculation and specify the calculated value in the RLN24nmLiCBR register. By specifying an incorrect checksum, the checksum error can be tested.
- Start header transmission to response reception
  Set the FTS bit in the RLN24nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started).
  The LIN self-test mode (reception) executed, interrupts are generated, and status and error status are also updated.
  To suspend the LIN self-test mode (reception) while it is running, set the OM0 bit in the RLN24nmLiCUC register to 0 (LIN reset mode), which causes a transition to the LIN reset mode.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN24nmLiIDB, RLN24nmLiDBRb, and RLN24nmLiCBR registers (the data is reversed before being stored because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN24nmLiTRC register is cleared.
- If the reception fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN24nmLiTRC register is cleared.

NOTE

x: Don't care

Note 1. The settings of the following registers are not reflected in the operation of the LIN self-test mode:
the RLN24nGLBRP0 register, the RLN24nGLBRP1 register, the LCKS bit in the RLN24nmLiMD register, and the IBS bit and IBHS bit (response space only) in the RLN24nmLiSC. Therefore, setting these registers is not necessary.

Note 2. If necessary, set the related registers described in **Section 7, Exceptions and Interrupts**.

17.15.4 Exiting LIN Self-Test Mode

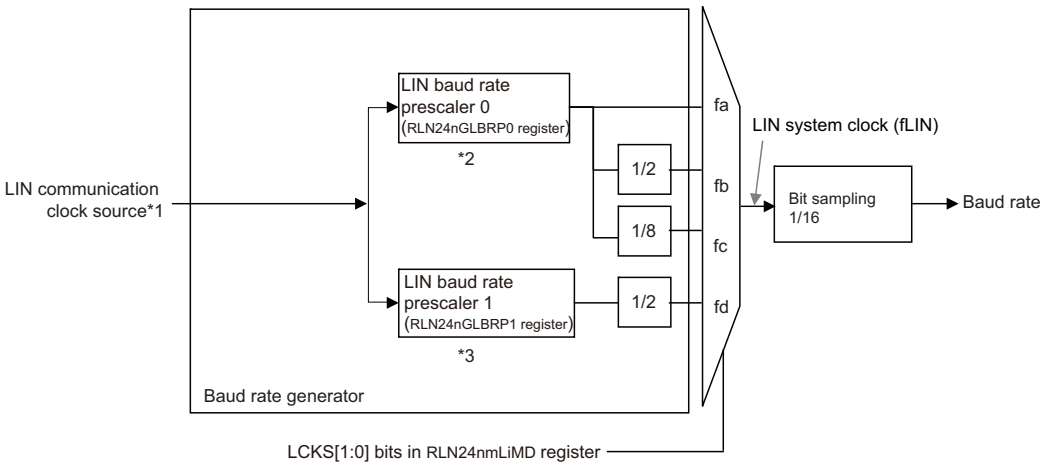
To exit LIN self-test mode, perform the procedure below:

- Switch all channels of the unit to LIN reset mode.
  Write 0 to the OM0 bit in the RLN24nmLiCUC register to make a transition to LIN reset mode. However, if the OMM1 and OMM0 bits in the RLN24nmLiMST register are not 11_B in any channels of the unit after the transition to LIN self-test mode, write 11_B to the OM1 and OM0 bits in the RLN24nmLiCUC register in any one channel. Check that the OMM1 and OMM0 bits in the RLN24nmLiMST register are set to 11_B, and then make a transition to LIN reset mode.
- Verify that the LIN master interface has exited LIN self-test mode.
  Read the LSTM bit in the RLN24nGLSTC register and confirm that it is not 0 (not LIN self-test mode)
- Verify the transition to LIN reset mode.
  Read the OMM0 bit in the RLN24nmLiMST register and confirm that it is 0 (LIN reset mode).

17.16 Baud Rate Generator

The LIN system clock (fLIN) is obtained by dividing the LIN communication clock source by the baud rate generator, and the baud rate is obtained by dividing that clock by 16. The inverse of this baud rate is called the bit time (Tbit).

Figure 17.16 shows a block diagram for baud rate generation.



Note 1. For the LIN communication clock source, see **11.5.3, Clock Domain Settings**.
Note 2. According to the RLN24nGLBRP0 register, set N + 1 division (N = 0 to 255).
Note 3. According to the RLN24nGLBRP1 register, set M + 1 division (M = 0 to 255).

Figure 17.16 Block Diagram of Baud Rate Generation

Set the LIN communication clock source in a range from 4 MHz to 40 MHz.

By setting the RLN24nGLBRP0 register so that fa is 307200 Hz (= 19200 × 16), the resulting system clock frequencies are fa = 19200 × 16, fb = 9600 × 16, and fc = 2400 × 16. These system clock frequencies are divided by 16 in the bit timing generator, enabling baud rates of 19200 bps, 9600 bps, and 2400 bps to be generated. Also, by setting the RLN24nGLBRP1 register so that fd is 166672 Hz (= 10417 × 16), the resulting system clock frequency is fd = 10417 × 16. This system clock frequency is divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

The formula for calculating baud rate is shown below.

Baud rate:

= {Frequency of LIN communication clock source} / (RLN24nGLBRP0 + 1) / 16 [bps]
(When fa is selected)

= {Frequency of LIN communication clock source} / (RLN24nGLBRP0 + 1) / 2 / 16 [bps]
(When fb is selected)

= {Frequency of LIN communication clock source} / (RLN24nGLBRP0 + 1) / 8 / 16 [bps]
(When fc is selected)

= {Frequency of LIN communication clock source} / (RLN24nGLBRP1 + 1) / 2 / 16 [bps]
(When fd is selected)

Section 18 LIN/UART Interface (RLIN3)

This section contains a generic description of the LIN/UART interface (RLIN3).

The first part of this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RLIN3.

18.1 Features of RH850/F1M RLIN3

18.1.1 Number of Units and Channels

This microcontroller has the following number of RLIN3 units.

Each RLIN3 unit has a single channel interface. "Number of channels" therefore has the same meaning as "number of units" in this section.

Table 18.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	6		
Name	RLIN3n (n = 0 to 5)		

Table 18.2 Unit Configurations and Channels

Unit Name (Channel Name) RLIN3n	Channels per Unit	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
RLIN30	1	√	√	√
RLIN31	1	√	√	√
RLIN32	1	√	√	√
RLIN33	1	√	√	√
RLIN34	1	√	√	√
RLIN35	1	√	√	√

Note: The channel names are same as those of the corresponding units.

Table 18.3 Indices

Index	Description
n	Throughout this section, the individual RLIN3 units are identified by the index “n” (n = 0 to 5); for example, RLN3nLCUC is the LIN control register.
b	Throughout this section, the individual transmit/receive data buffers of RLIN3n are identified by the index “b” (b = 1 to 8); for example, RLN3nLDBRb is the data buffer register.

The following lists the index value corresponding to each product.

Table 18.4 Index Correspondence of Each Product

Index Corresponding to Product
All products
b = 1 to 8

18.1.2 Register Base Address

RLIN3 base addresses are listed in the following table.

RLIN3 register addresses are given as offsets from the base addresses.

Table 18.5 Register Base Addresses

Base Address Name	Base Address
<RLIN30_base>	FFCE 2000$_H$
<RLIN31_base>	FFCE 2040$_H$
<RLIN32_base>	FFCE 2080$_H$
<RLIN33_base>	FFCE 20C0$_H$
<RLIN34_base>	FFCE 2100$_H$
<RLIN35_base>	FFCE 2140$_H$

18.1.3 Clock Supply

The RLIN3 clock supply is shown in the following table.

Table 18.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RLIN3n	LIN communication clock sources	CKSCLK_ILIN[*1,*2]
	Register access clock	CKSCLK_ILIN[*1,*2]

Note 1. The clock domain CKSCLK_ILIN divided clock can be supplied only to RLIN30 channel.

Note 2. Set the LIN communication clock source in the range of 4 MHz to 60 MHz.

18.1.4 Interrupt Request

RLIN3 interrupt requests are listed in the following table.

Table 18.7 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
RLIN30			
INTRLIN3n (n = 0)	RLIN30 interrupt	33	—
INTRLIN3nUR0 (n = 0)	RLIN30 transmit interrupt	34	10
INTRLIN3nUR1 (n = 0)	RLIN30 receive completion interrupt	35	11
INTRLIN3nUR2 (n = 0)	RLIN30 status interrupt	36	—
RLIN31			
INTRLIN3n (n = 1)	RLIN31 interrupt	120	—
INTRLIN3nUR0 (n = 1)	RLIN31 transmit interrupt	121	86
INTRLIN3nUR1 (n = 1)	RLIN31 receive completion interrupt	122	87
INTRLIN3nUR2 (n = 1)	RLIN31 status interrupt	123	—
RLIN32			
INTRLIN3n (n = 2)	RLIN32 interrupt	164	—
INTRLIN3nUR0 (n = 2)	RLIN32 transmit interrupt	165	44
INTRLIN3nUR1 (n = 2)	RLIN32 receive completion interrupt	166	45
INTRLIN3nUR2 (n = 2)	RLIN32 status interrupt	167	—
RLIN33			
INTRLIN3n (n = 3)	RLIN33 interrupt	228	—
INTRLIN3nUR0 (n = 3)	RLIN33 transmit interrupt	229	111
INTRLIN3nUR1 (n = 3)	RLIN33 receive completion interrupt	230	112
INTRLIN3nUR2 (n = 3)	RLIN33 status interrupt	231	—
RLIN34			
INTRLIN3n (n = 4)	RLIN34 interrupt	232	—
INTRLIN3nUR0 (n = 4)	RLIN34 transmit interrupt	233	50
INTRLIN3nUR1 (n = 4)	RLIN34 receive completion interrupt	234	51
INTRLIN3nUR2 (n = 4)	RLIN34 status interrupt	235	—
RLIN35			
INTRLIN3n (n = 5)	RLIN35 interrupt	236	—
INTRLIN3nUR0 (n = 5)	RLIN35 transmit interrupt	237	113
INTRLIN3nUR1 (n = 5)	RLIN35 receive completion interrupt	238	114
INTRLIN3nUR2 (n = 5)	RLIN35 status interrupt	239	—

18.1.5 Reset Sources

RLIN3 reset sources are listed in the following table. RLIN3 is initialized by these reset sources.

Table 18.8 Reset Sources

Unit Name	Reset Source
RLIN3n	All reset sources (ISORES)

18.1.6 External Input/output Signals

External input/output signals of RLIN3 are listed below.

Table 18.9 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
RLIN30		
RLIN3nRX (n = 0)	RLIN30 receive data input	RLIN30RX
RLIN3nTX (n = 0)	RLIN30 transmit data output	RLIN30TX
RLIN31		
RLIN3nRX (n = 1)	RLIN31 receive data input	RLIN31RX
RLIN3nTX (n = 1)	RLIN31 transmit data output	RLIN31TX
RLIN32		
RLIN3nRX (n = 2)	RLIN32 receive data input	RLIN32RX
RLIN3nTX (n = 2)	RLIN32 transmit data output	RLIN32TX
RLIN33		
RLIN3nRX (n = 3)	RLIN33 receive data input	RLIN33RX
RLIN3nTX (n = 3)	RLIN33 transmit data output	RLIN33TX
RLIN34		
RLIN3nRX (n = 4)	RLIN34 receive data input	RLIN34RX
RLIN3nTX (n = 4)	RLIN34 transmit data output	RLIN34TX
RLIN35		
RLIN3nRX (n = 5)	RLIN35 receive data input	RLIN35RX
RLIN3nTX (n = 5)	RLIN35 transmit data output	RLIN35TX

18.2 Overview

18.2.1 Functional Overview

The LIN/UART interface is a hardware LIN communication controller that supports LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602, and automatically performs frame communication and error determination.

The LIN/UART interface is provided with UART mode and can also be used as a UART.

The appropriate mode should be used for the LIN/UART interface according to the application: LIN master, LIN slave, or UART.

LIN master

- LIN reset mode
- LIN mode (LIN master mode)
  - LIN wake-up mode
  - LIN operation mode
- LIN self-test mode

LIN slave

- LIN reset mode
- LIN mode (LIN slave mode [auto baud rate] or LIN slave mode [fixed baud rate])
  - LIN wake-up mode
  - LIN operation mode
- LIN self-test mode

UART

- LIN reset mode
- UART mode

Table 18.10 shows the LIN/UART interface specifications.

Table 18.10 LIN/UART Interface Specifications (1/3)

Item		Specifications	
	Channel count	Up to 6 channels	
LIN communication function	Protocol	LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602	
	Variable frame structure	Master	• Break transmission width: 13 to 28 Tbits • Break delimiter transmission width: 1 to 4 Tbits • Transmission inter-byte space width (header): 0 to 7 Tbits (space between Sync field and ID field)*1 • Transmission response space width: 0 to 7 Tbits*1 • Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area) • Transmission wake-up width: 1 to 16 Tbits
		Slave	• Break reception width : 9.5 or 10.5 Tbits [for fixed baud rate] : 10 or 11 Tbits [for auto baud rate] • Transmission response space width: 0 to 7 Tbits • Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area) • Transmission wake-up width: 1 to 16 Tbits
	Checksum	• Automatic operation for both transmission and reception • Classic or enhanced selectable (for each frame)	
	Response field data byte count	Variable from 0 to 8 bytes Multi-byte (9 or more bytes) response transmission and reception also possible	
	Frame communication modes	Master	• Mode in which header transmission and response transmission/reception are started with a single transmission start request • Mode in which header transmission and response transmission are started with separate transmission start requests (frame separate mode)
		Slave	• Mode in which header is automatically received with fixed baud rate • Mode in which header is automatically received with the baud rate set according to the sync field measurement result of the sync field and break field detected
	Wake-up transmission and reception	LIN wake-up mode provided • Wake-up transmission (1 to 16 Tbits) • Wake-up reception Low-level width of input signals measured	
	Status	Master	• Successful frame/wake-up transmission • Successful header transmission • Successful frame/wake-up reception*2 • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)
		Slave	• Successful response/wake-up transmission • Successful response/wake-up reception*2 • Successful header reception • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)

Table 18.10 LIN/UART Interface Specifications (2/3)

Item		Specifications	
LIN communication function	Error status	Master	• Bit error • Checksum error • Frame timeout error/response timeout error • Physical bus error • Framing error • Response preparation error
		Slave	• Bit error • Checksum error • Frame timeout error/response timeout error • Sync field error • ID parity error • Framing error • Response preparation error
	Baud rate selection	Baud rates conforming to the LIN specifications generated using baud rate generator	
	Test mode	Self-test mode for user evaluation	
	Interrupt function	Master	• Successful header/frame/wake-up transmission • Successful frame/wake-up reception*2 • Error detection
		Slave	• Successful response/wake-up transmission • Successful Header/response/wake-up reception*2 • Error detection
UART communication function	Data buffer	• Transmission data buffer/transmission data buffer for wait (exclusively for transmission; data length of 1. Character length of 7, 8, and 9 bits supported) • UART buffer (exclusively for transmission; variable data length from 1 to 9. Character length of 7 and 8 bits supported) • Reception data buffer (exclusively for reception; data length of 1. Character length of 7, 8, and 9 bits supported)	
	Data format	Character length: 7 or 8 bits Length of 9 bits supported by using the expansion bit.	
		Transmission stop bit: 1 or 2 bits	
		Parity function: odd, even, 0, or none	
		LSB- or MSB-first transfer selectable	
		Reverse input/output of transmission/reception data possible	
	Status	• Transmission status • Reception status • Successful UART buffer transmission • Error detection • Expansion bit detection • ID match • Reset mode status	
	Error status	• Bit error • Framing error • Parity error • Overrun error	
	Baud rate selection	With the built-in baud rate generator, any baud rate can be set.	
	When a certain expansion bit is at the expected level, the data received can be compared to the 8-bit data preset in the register.		

Table 18.10 LIN/UART Interface Specifications (3/3)

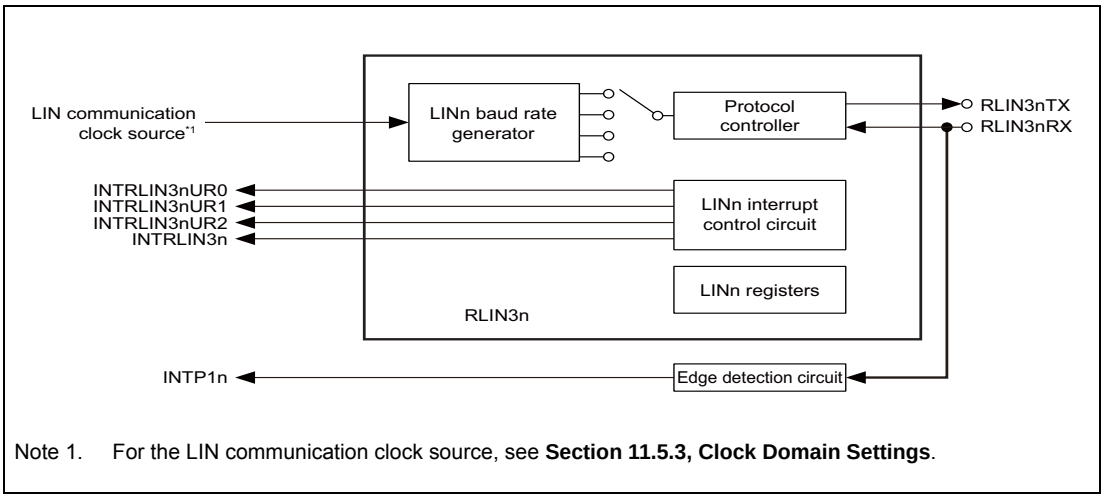
Item		Specifications
UART communication function	Reception of the stop bit is guaranteed (start of transmission can be delayed when start of transmission is attempted during reception of the stop bit).	
	Interrupt function	• Transmission start/complete • Reception complete • Status/error detection

Note 1. Since the same register is used for setting, the inter-byte space (header) = response space.

Note 2. For wake-up reception, the input signal low-level width count is indicated.

18.2.2 Block Diagram

Figure 18.1 shows a block diagram of the LIN/UART interface.



Note 1. For the LIN communication clock source, see **Section 11.5.3, Clock Domain Settings**.

Figure 18.1 LIN/UART Interface Block Diagram

18.2.3 Terms used in block diagram

- RLIN3nTX, RLIN3nRX: LIN/UART interface I/O pins
- LINn baud rate generator: Generates the LIN/UART interface communication clock.
- LINn registers: LIN/UART interface registers
- LINn interrupt control circuit: Controls interrupt requests generated by the LIN/UART interface

18.3 Registers

18.3.1 List of Registers

RLIN3 registers are listed in the following table.

For details about <RLIN3n_base>, see **Section 18.1.2, Register Base Address**.

Table 18.11 List of Registers (1/2)

Module Name	Register Name	Symbol	Address	LIN Master	LIN Slave	UART
RLN3n	LIN wake-up baud rate select register	RLN3nLWBR	<RLIN3n_base> + 01_H	√	√	√
RLN3n	LIN / UART baud rate prescaler 01 register	RLN3nLBRP01	<RLIN3n_base> + 02_H	—	√	√
RLN3n	LIN / UART baud rate prescaler 0 register	RLN3nLBRP0	<RLIN3n_base> + 02_H	√	√	√
RLN3n	LIN / UART baud rate prescaler 1 register	RLN3nLBRP1	<RLIN3n_base> + 03_H	√	√	√
RLN3n	LIN self-test control register	RLN3nLSTC	<RLIN3n_base> + 04_H	√	√	—
RLN3n	LIN / UART mode register	RLN3nLMD	<RLIN3n_base> + 08_H	√	√	√
RLN3n	LIN break field configuration register/ UART configuration register	RLN3nLBFC	<RLIN3n_base> + 09_H	√	√	√
RLN3n	LIN / UART space configuration register	RLN3nLSC	<RLIN3n_base> + $0A_H$	√	√	√
RLN3n	LIN wake-up configuration register	RLN3nLWUP	<RLIN3n_base> + $0B_H$	√	√	—
RLN3n	LIN interrupt enable register	RLN3nLIE	<RLIN3n_base> + $0C_H$	√	√	—
RLN3n	LIN / UART error detection enable register	RLN3nLEDE	<RLIN3n_base> + $0D_H$	√	√	√
RLN3n	LIN / UART control register	RLN3nLCUC	<RLIN3n_base> + $0E_H$	√	√	√
RLN3n	LIN / UART transmission control register	RLN3nLTRC	<RLIN3n_base> + 10_H	√	√	√
RLN3n	LIN / UART mode status register	RLN3nLMST	<RLIN3n_base> + 11_H	√	√	√
RLN3n	LIN / UART status register	RLN3nLST	<RLIN3n_base> + 12_H	√	√	√
RLN3n	LIN / UART error status register	RLN3nLEST	<RLIN3n_base> + 13_H	√	√	√
RLN3n	LIN data field configuration register	RLN3nLDFC	<RLIN3n_base> + 14_H	√	√	√
RLN3n	LIN / UART ID buffer register	RLN3nLIDB	<RLIN3n_base> + 15_H	√	√	√
RLN3n	LIN checksum buffer register	RLN3nLCBR	<RLIN3n_base> + 16_H	√	√	—
RLN3n	UART data buffer 0 register	RLN3nLUDB0	<RLIN3n_base> + 17_H	—	—	√
RLN3n	LIN / UART data buffer 1 register	RLN3nLDBR1	<RLIN3n_base> + 18_H	√	√	√
RLN3n	LIN / UART data buffer 2 register	RLN3nLDBR2	<RLIN3n_base> + 19_H	√	√	√
RLN3n	LIN / UART data buffer 3 register	RLN3nLDBR3	<RLIN3n_base> + $1A_H$	√	√	√
RLN3n	LIN / UART data buffer 4 register	RLN3nLDBR4	<RLIN3n_base> + $1B_H$	√	√	√
RLN3n	LIN / UART data buffer 5 register	RLN3nLDBR5	<RLIN3n_base> + $1C_H$	√	√	√
RLN3n	LIN / UART data buffer 6 register	RLN3nLDBR6	<RLIN3n_base> + $1D_H$	√	√	√
RLN3n	LIN / UART data buffer 7 register	RLN3nLDBR7	<RLIN3n_base> + $1E_H$	√	√	√
RLN3n	LIN / UART data buffer 8 register	RLN3nLDBR8	<RLIN3n_base> + $1F_H$	√	√	√
RLN3n	UART operation enable register	RLN3nLUOER	<RLIN3n_base> + 20_H	—	—	√
RLN3n	UART option register 1	RLN3nLUOR1	<RLIN3n_base> + 21_H	—	—	√
RLN3n	UART transmission data register	RLN3nLUTDR	<RLIN3n_base> + 24_H	—	—	√
RLN3n	UART transmission data register L	RLN3nLUTDRL	<RLIN3n_base> + 24_H	—	—	√
RLN3n	UART transmission data register H	RLN3nLUTDRH	<RLIN3n_base> + 25_H	—	—	√
RLN3n	UART reception data register	RLN3nLURDR	<RLIN3n_base> + 26_H	—	—	√
RLN3n	UART reception data register L	RLN3nLURDRL	<RLIN3n_base> + 26_H	—	—	√

Table 18.11 List of Registers (2/2)

Module Name	Register Name	Symbol	Address	LIN Master	LIN Slave	UART
RLN3n	UART reception data register H	RLN3nLURDRH	<RLIN3n_base> + 27_H	—	—	√
RLN3n	UART wait transmission data register	RLN3nLUWTDR	<RLIN3n_base> + 28_H	—	—	√
RLN3n	UART wait transmission data register L	RLN3nLUWTDRL	<RLIN3n_base> + 28_H	—	—	√
RLN3n	UART wait transmission data register H	RLN3nLUWTDRH	<RLIN3n_base> + 29_H	—	—	√

Note: √: Used, —: Not used
When writing to an unused register, write the value after reset.

18.3.2 LIN Master Related Registers

18.3.2.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			LWBR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.12 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings 1 1 1 1: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 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
0	LWBR0	Wake-up Baud Rate Select 0: In LIN wake-up mode, the clock specified in the LCKS bit of the RLN3nLMD register is used. (LIN1.3) 1: In LIN wake-up mode, the clock fa is used regardless of the setting in the LCKS bit of the RLN3nLMD register. (LIN2.x)

Configure the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

These bits select the number of sampling in one Tbit (reciprocal of the baud rate).
In LIN master mode (LIN/UART mode select bits in LIN mode register = 00_B), set these bits to 0000_B or 1111_B (16 samplings).

LPRS[2:0] Bits (Prescaler Clock Select)

These bits select the frequency division ratio for the prescaler.
The LIN communication clock source is divided by this prescaler.

LWBR0 Bit (Wake-up Baud Rate Select)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.
When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects fa as the LIN system clock (fLIN) during LIN wake-up mode regardless of the setting of the

RLN3nLMD.LCKS bit (the LCKS bit is not changed). This allows an input signal with a low-level width of 2.5 Tbits or more to be measured.

Setting the baud rate to 19200 bps while fa is selected allows an input signal with a low-level width of 130 us or longer to be detected in the LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS bit.

18.3.2.2 RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 02_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP0[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.13 RLN3nLBRP0 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP0[7:0]	Assuming that the value set in this register is N (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by N + 1. Setting range: 00_H to FF_H

Configure the RLN3nLBRP0 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock sources fa, fb, and fc.

Assuming that the value set in this register is N, baud rate prescaler 0 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by N + 1.

18.3.2.3 RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 03_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP1[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.14 RLN3nLBRP1 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP1[7:0]	Assuming that the value set in this register is M (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by M + 1. Setting range: 00_H to FF_H

Set the RLN3nLBRP1 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock source fd.

Assuming that the value set in this register is M, baud rate prescaler 1 divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) by M+1.

18.3.2.4 RLN3nLSTC — LIN Self-Test Control Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 04_H

Value after reset: 00_H

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

Table 18.15 RLN3nLSTC Register Contents

Bit Position	Bit Name	Function
7 to 0	—	Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN self-test mode. 1: The module is in LIN self-test mode.

The RLN3nLSTC register cancels protection of LIN self-test mode.

Set the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.

When successive writing is completed and the mode is changed to LIN self-test mode, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN self-test mode, see **Section 18.9, LIN Self-Test Mode**.

When read, bits 6 to 1 return "000000_B", and bit 7 returns an undefined value.

LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.
For exiting LIN self-test mode, see **Section 18.9, LIN Self-Test Mode**.
Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of $A7_H$, 58_H, and 01_H.

18.3.2.5 RLN3nLMD — LIN Mode Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base>+ 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	LCKS[1:0]		LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.16 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4	LIOS	LIN Interrupt Output Select 0: RLIN3n interrupt is used. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are used.
3, 2	LCKS[1:0]	LIN System Clock Select b3 b2 0 0: fa (Clock generated by baud rate prescaler 0) 0 1: fb (1/2 clock generated by baud rate prescaler 0) 1 0: fc (1/8 clock generated by baud rate prescaler 0) 1 1: fd (1/2 clock generated by baud rate prescaler 1)
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 0 0: LIN master mode

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LRDNFS Bit (LIN Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.
With 0 set, the RLIN3 interrupt is generated from the LIN/UART interface.
With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are generated from the LIN/UART interface.
For each interrupt source, see **Section 18.4, Interrupt Sources**.

LCKS[1:0] Bits (LIN System Clock Select)

The LCKS bits select the clock to be input to the protocol controller.
With 00_B set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).
With 01_B set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).
With 10_B set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).
With 11_B set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).

When 1_B is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x), and the RLN3nLMST register is 01_H (LIN wake-up mode), the protocol controller is supplied with fa regardless of the setting of this bit (the LCKS bit is not changed).

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.
To use the LIN/UART interface as an LIN master, set these bits to 00_B.

18.3.2.6 RLN3nLBFC — LIN Break Field Configuration Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

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

Table 18.17 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	BDT[1:0]	Transmission Break Delimiter (High Level) Width Select b5 b4 0 0: 1 Tbit 0 1: 2 Tbits 1 0: 3 Tbits 1 1: 4 Tbits
3 to 0	BLT[3:0]	Transmission Break (Low Level) Width Select b3 b0 0 0 0 0: 13 Tbits 0 0 0 1: 14 Tbits 0 0 1 0: 15 Tbits : 1 1 1 0: 27 Tbits 1 1 1 1: 28 Tbits

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Some combinations of the set values result in the length of a frame exceeding the timeout time. Set the appropriate values in this register.

BDT[1:0] Bits (Transmission Break Delimiter (High Level) Width Select)

The BDT bits set the break delimiter (high level) width of transmission frame header.
1 Tbit to 4 Tbits can be set.

BLT[3:0] Bits (Transmission Break (Low Level) Width Select)

The BLT bits set the break (low level) width of transmission frame header.
13 Tbits to 28 Tbits can be set.

18.3.2.7 RLN3nLSC — LIN Space Configuration Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base>+ $0A_H$

Value after reset: 00_H

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

Table 18.18 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Inter-Byte Space (Header)/Response Space Select b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space of the transmission frame response field.
0 Tbit to 3 Tbits can be set.
These bits are enabled only during response transmission; these are disabled during response reception.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the width of the inter-byte space (header) of the transmission frame header field and the response space.
0 Tbit to 7 Tbits can be set.
The response space setting is enabled only during response transmission; setting is disabled during response reception.
The inter-byte space (header) is equal to the response space.

18.3.2.8 RLN3nLWUP — LIN Wake-Up Configuration Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base>+ $0B_H$

Value after reset: 00_H

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

Table 18.19 RLN3nLWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low Level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL bits set the low level width of the wake-up signal transmission.
1 Tbit to 16 Tbits can be set.

With 1 is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x), fa is selected as the LIN system clock (fLIN) regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed).

18.3.2.9 RLN3nLIE — LIN Interrupt Enable Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 18.20 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SHIE	Successful Header Transmission Interrupt Request Enable 0: Disables successful header transmission interrupt request. 1: Enables successful header transmission interrupt request.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Request Enable 0: Disables successful frame/wake-up reception interrupt request. 1: Enables successful frame/wake-up reception interrupt request.
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Request Enable 0: Disables successful frame/wake-up transmission interrupt request. 1: Enables successful frame/wake-up transmission interrupt request.

Set the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

SHIE Bit (Successful Header Transmission Interrupt Request Enable)

The SHIE bit enables or disables interrupt requests upon successful transmission of a header.
With 0 set, the interrupt request for RLIN3n transmission is not generated when the HTRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n transmission is generated when the HTRC flag in the RLN3nLST register is set to 1.

ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.
With 0 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n status is generated when the ERR flag in the RLN3nLST register is set to 1.
Occurrence factors are bit errors, physical bus errors, frame/response timeout errors, framing errors, checksum errors, and response preparation errors.
Detection of the bit error, physical bus error, frame/response timeout error, and framing error can be enabled or disabled using the RLN3nLEDE register.

FRCIE Bit (Successful Frame/Wake-up Reception Interrupt Request Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a frame or a wake-up signal (input signal low-level width count).

With 0 set, the interrupt request for successful RLIN3n reception is not generated when the FRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for successful RLIN3n reception is generated when the FRC flag in the RLN3nLST register is set to 1.

FTCIE Bit (Successful Frame/Wake-up Transmission Interrupt Request Enable)

The FTCIE bit enables or disables an interrupt request upon successful transmission of a frame or a wake-up signal.
With 0 set, the interrupt request for RLIN3n transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

18.3.2.10 RLN3nLEDE —LIN Error Detection Enable Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + $0D_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LTES	—	—	—	FERE	FTERE	PBERE	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Table 18.21 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable*1 0: Disables framing error detection. 1: Enables framing error detection.
2	FTERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	PBERE	Physical Bus Error Detection Enable 0: Disables physical bus error detection. 1: Enables physical bus error detection.
0	BERE	Bit Error Detection Enable*1 0: Disables bit error detection. 1: Enables bit error detection.

Note 1. Set FERE bit and BERE bit to 1.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.
With 0 set, the timeout function is used for frame timeout.
With 1 set, the timeout function is used for response timeout.
For details on the timeout error, see **Section 18.7.7, Error Status**.

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
Set this bit to 1. The detection result of the framing error is indicated in the FER flag in the RLN3nLEST register.
For details on the framing error, see **Section 18.7.7, Error Status**.

FTERE Bit (Timeout Error Detection Enable)

The FTERE bit enables or disables detection of the frame timeout error or the response timeout error.
With 0 set, the frame timeout error or response timeout error is not detected.

With 1 set, the frame timeout error or response timeout error is detected.
When this bit is set to 1, the detection result is indicated in the FTER flag in the RLN3nLEST register.
With the LTES bit, either the frame timeout error or response timeout error can be selected.
Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.
For details on the timeout error, see **Section 18.7.7, Error Status**.

PBERE Bit (Physical Bus Error Detection Enable)

The PBERE bit enables or disables detection of the physical bus error.
With 0 set, the physical bus error is not detected.
With 1 set, the physical bus error is detected.
When this bit is set to 1, the detection result is indicated in the PBER flag in the RLN3nLEST register.
For details on the physical bus error, see **Section 18.7.7, Error Status**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
Set this bit to 1. The detection result of the bit error is indicated in the BER flag in the RLN3nLEST register.
For details on the bit error, see **Section 18.7.7, Error Status**.

18.3.2.11 RLN3nLCUC — LIN Control Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + $0E_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 18.22 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode. 1: LIN operation mode.
0	OM0	LIN Reset 0: LIN reset mode. 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to transition to LIN wake-up mode or to 03_H to transition to LIN operation mode after exiting LIN reset mode.

In LIN self-test mode, set the RLN3nLCUC register to 03_H after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.
With 0 set, LIN wake-up mode.
With 1 set, LIN operation mode.
This bit is valid only when the OMM0 bit in the RLN3nLMST register is 1.
Writing a value to this bit is disabled while the FTS bit in the RLN3nLTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.
With 0 set, LIN reset mode.
With 1 set, LIN reset mode is canceled.

18.3.2.12 RLN3nLTRC — LIN Transmission Control Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

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

Table 18.23 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped in frame separate mode. 1: Response transmission/reception is started in frame separate mode.
0	FTS	Frame Transmission/Wake-up Transmission /Reception Start 0: Frame Transmission/wake-up transmission/reception is stopped. 1: Frame Transmission/wake-up transmission reception is started.

RTS Bit (Response Transmission/Reception Start)

Set the RTS bit to 1 in frame separate mode after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame communication (including error detection) or transition to LIN reset mode.
Only 1 can be written to this bit; 0 cannot be written.
To write 1 to this bit, write 02_H to the RLN3nLTRC register using the store instruction.
Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).
Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 at the end of data group communication or transition to LIN reset mode.

FTS Bit (Frame Transmission/Wake-up Transmission/Reception Start)

Set the FTS bit to 1 to start frame transmission and reception.
Also set this bit to 1 to allow wake-up transmission and reception (input signal low-level width count).
Only 1 can be written to this bit; 0 cannot be written.
Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication (including error detection) and transition to LIN reset mode.

18.3.2.13 RLN3nLMST — LIN Mode Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RLIN3n_base> + 11_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 18.24 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

18.3.2.14 RLN3nLST — LIN Status Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 18.25 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Transmission Flag 0: Header transmission has not been completed. 1: Header transmission has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Frame/Wake-up Reception Flag 0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed.
0	FTC	Successful Frame/Wake-up Transmission Flag 0: Frame or wake-up transmission has not been completed. 1: Frame or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Successful Header Transmission Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The HTRC flag is set to 1 upon completion of header transmission. Here, an interrupt request for RLN3n transmission is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not generated. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC

register is 1), write 0 to the bit in LIN operation mode.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when at least one of the flags of the RLN3nLEST register is set to 1). Here, an interrupt request for RLN3n status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the RPER, CSER, FER, FTER, PBER, and BER flags in the RLN3nLEST register in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

FRC Flag (Successful Frame/Wake-up Reception Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FRC flag is set to 1 upon completion of frame or wake-up reception. Here, an interrupt request for RLN3n reception complete is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

FTC Flag (Successful Frame/Wake-up Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FTC flag is set to 1 upon completion of frame or wake-up transmission. Here, an interrupt request for RLN3n transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.
When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

18.3.2.15 RLN3nLEST — LIN Error Status Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RPER	—	CSER	—	FER	FTER	PBER	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 18.26 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	FTER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	PBER	Physical Bus Error Flag 0: Physical bus error has not been detected. 1: Physical bus error has been detected.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

When the FTS bit in the RLN3nLTRC register is 1 (frame transmission or wake-up transmission/ reception is started), do not write a value to this register.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The RPER flag is set to 1 upon response preparation error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the value of the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled), the FER flag is set to 1 upon framing error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

FTER Flag (Timeout Error Flag)

Only 0 can be written to the FTER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the FTERE bit of the RLN3nLEDE register is 1 (frame/response timeout error detection enabled), the FTER flag is set to 1 upon frame timeout error or response timeout error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

PBER Flag (Physical Bus Error Flag)

Only 0 can be written to the PBER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the PBERE bit of the RLN3nLEDE register is 1 (physical bus error detection enabled), the PBER flag is set to 1 upon physical bus error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled), the BER flag is set to 1 upon bit error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

18.3.2.16 RLN3nLDFC — LIN Data Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LSS	FSM	CSM	RFT	RFDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.27 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Checksum is not included.)
6	FSM	Frame Separate Mode Select 0: Frame separate mode is not set. 1: Frame separate mode is set.
5	CSM	Checksum Select 0: Classic checksum 1: Enhanced checksum
4	RFT	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 byte (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited.

LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received.
With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.
With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.
Set the LSS bit only when the FSM bit is 1 (frame separate mode) and response data of 9 bytes or more is to be transmitted or received.

Set the LSS bit only when the RTS bit in the RLN3nLTRC is 0 (response transmit/receive is stopped).

FSM Bit (Frame Separate Mode Select)

The FSM bit sets the response communication mode.
With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the RLN3nLTRC register is 1), response is transmitted/received without setting the RTS bit in the RLN3nLTRC register.

With 1 set, frame separate mode is selected. If the RTS bit of the RLN3nLTRC register is set to 1 during header transmission, response transmission is executed after header transmission is completed.

For response reception which is 8 bytes or less (the RFT bit is 0), set the FSM bit to 0.
When causing a transition to LIN self-test mode, set this bit to 0 before transition.
For details on frame separate mode, see **Section 18.7.4.1, Transmission of LIN Frames**.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, set the FSM bit to 1.

CSM Bit (Checksum Select)

The CSM bit sets the checksum mode.
With 0 set, classic checksum mode is selected.
With 1 set, enhanced checksum mode is selected.
When the timeout error is used (the FTERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details on the bit error, see **Section 18.7.7, Error Status**.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, do not change the CSM bit setting after the first data group through the last data group.
During communication of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

RFT Bit (Response Field Communication Direction Select)

The RFT bit sets the direction of the response field/wake-up signal communication.
With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low-level width count).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, do not change the RFT bit setting after the first data group through the last data group.

RFDL[3:0] Bits (Response Field Length Select)

The RFDL bits set the length of the response field data.
The data length can be 0 to 8 bytes excluding the checksum size.
To transmit response data with the FSM bit set to 0 (not frame separate mode), set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).
To transmit response data with the FSM bit set to 1 (frame separate mode), set the RFDL bits before response transmission (the RTS bit in the RLN3nLTRC register is 0).
To receive response data, set the RFDL bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).
When response data of 9 bytes or more is to be transmitted or received, set the RFDL bits before data group transmission/reception (RTS bit in the RLN3nLTRC register is 0).
Only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

18.3.2.17 RLN3nLIDB — LIN ID Buffer Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.28 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7	IDP1	Parity Setting (P1) Sets the parity bits (P1) to be transmitted in the ID field.
6	IDP0	Parity Setting (P0) Sets the parity bits (P0) to be transmitted in the ID field.
5 to 0	ID[5:0]	ID Setting Sets the 6-bit ID value to be transmitted in the ID field.

Set the RLN3nLIDB register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

In LIN self-test mode, this register operates as described below.

Write the value to be transmitted before communication. After completion of frame transmission/ reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 18.9, LIN Self-Test Mode**.

IDP[1:0] Bits (Parity Setting)

The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame. IDP0 is for P0 and IDP1 is for P1.

Since parity is not automatically calculated, set the calculation result. Note that if the erroneous result is set, it is transmitted as is.

ID[5:0] Bits (ID Setting)

The ID bits set the 6-bit ID value to be transmitted in the ID field of the LIN frame.

18.3.2.18 RLN3nLCBR — LIN Checksum Buffer Register

Access: This register is a read-only register that can be read in 8-bit units. In LIN self-test mode, this register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 16_H

Value after reset: 00_H

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

Table 18.29 RLN3nLCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the checksum data transmitted or received.

In LIN mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
  The value transmitted can be read from the register. Read the value after transmission is completed.
  Writing to this register is invalid.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
  The value received can be read from the register. Read the value after reception is completed.
  Writing to this register is invalid.

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for other data groups.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
  After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
  Write the value to be received before communication. After completion of frame transmission/ reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 18.9, LIN Self-Test Mode**.

Set the this register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

18.3.2.19 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

Access: This register can be read or written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H
RLN3nLDBR2: <RLIN3n_base> + 19_H
RLN3nLDBR3: <RLIN3n_base> + $1A_H$
RLN3nLDBR4: <RLIN3n_base> + $1B_H$
RLN3nLDBR5: <RLIN3n_base> + $1C_H$
RLN3nLDBR6: <RLIN3n_base> + $1D_H$
RLN3nLDBR7: <RLIN3n_base> + $1E_H$
RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Value after reset: 00_H

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

Table 18.30 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or reads the received data. Setting range: 00_H to FF_H

- For response transmission:
  These registers set the data to be transmitted in the response field.
  Use these registers with the following settings.
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 0 (not frame separate mode)
  - FTS in RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted)

  or
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 1 (frame separate mode)
  - RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)
- For response reception:
  These registers hold the data received in the response field.
  The received data is overwritten. If an error is detected, the data prior to reception interruption is stored in the register.
  Do not read these registers when the FTS bit is 1 (frame transmission or wake-up transmission/reception is started)
- For transmission of response data of 9 bytes or more:
  Use these registers with the following settings.
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 1 (frame separate mode)
  - RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)
- For reception of response data of 9 bytes or more:
  Do not read these registers when the RTS bit is 1 (response transmission/reception is started).

In LIN self-test mode, these registers operate as described below.

Write the value to be transmitted before communication. After completion of frame transmission/ reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 18.9, LIN Self-Test Mode**.

18.3.3 LIN Slave Related Registers

18.3.3.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

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

Table 18.31 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings 0 0 1 1: 4 samplings 0 1 1 1: 8 samplings 1 1 1 1: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 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
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).
When the communication is performed in LIN slave mode (fixed baud rate) (LMD[1:0] bits in the RLN3nLMD register = 11_B), set these bits to "0000_B" or "1111_B" (16 samplings).
When the LIN frame communication is performed in LIN slave mode (auto baud rate) (LMD[1:0] bits in the RLN3nLMD register = 10_B), set these bits to "0011_B" (4 samplings) or "0111_B" (8 samplings).

LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS bits select the frequency division ratio for the prescaler. The LIN communication clock source is divided by this prescaler.

In LIN slave mode (auto baud rate) (LMD[1:0] bits in the RLN3nLMD register = 10_B), set these bits so that the prescaler clock becomes as follows according to the target baud rate.

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4MHz*1
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

Note 1. Use the clock with NSPB bits set to "0011_B" (four samplings).

18.3.3.2 RLN3nLBRP01 — LIN Baud Rate Prescaler 01 Register

Access: RLN3nLBRP01 can be read or written in 16-bit units.
RLN3nLBRP0 can be read or written in 8-bit units.
RLN3nLBRP1 can be read or written in 8-bit units.

Address: RLN3nLBRP01: <RLIN3n_base> + 02_H
RLN3nLBRP0: <RLIN3n_base> + 02_H
RLN3nLBRP1: <RLIN3n_base> + 03_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.32 RLN3nLBRP01 Register Contents

Bit Position	Bit Name	Function
15 to 0	BRP[15:0]	Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000_H to $FFFF_H$

Set the RLN3nLBRP01 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the RLN3nLWBR register by L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units by registers RLN3nLBRP0 and RLN3nLBRP1.

NOTE

In LIN slave mode [auto baud rate], the system automatically sets the result of baud rate correction to the RLN3nLBRP01 register on successful reception of the sync field.

18.3.3.3 RLN3nLSTC — LIN Self-Test Control Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 04_H

Value after reset: 00_H

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

Table 18.33 RLN3nLSTC Register Contents

Bit Position	Bit Name	Function
7 to 0	—	Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.
0	LSTM	LIN Self-Test Mode 0: LIN self test mode is not set. 1: LIN self test mode is set.

The RLN3nLSTC register cancels protection of LIN self-test mode.

Set the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.

When successive writing is completed thus placing LIN self-test mode to be entered, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN self-test mode, see **Section 18.9, LIN Self-Test Mode**.

When reading bits 6 to 1, they return "000000_B", and reading bit 7, it returns an undefined value.

LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.
For exiting LIN self-test mode, see **Section 18.9, LIN Self-Test Mode**.
Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of $A7_H$, 58_H, and 01_H.

18.3.3.4 RLN3nLMD — LIN Mode Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	—	—	LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 18.34 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4	LIOS	LIN Interrupt Output Select 0: RLIN3 interrupt is used. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are used.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 1 0: LIN slave mode with auto baud rate 1 1: LIN slave mode with fixed baud rate

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LRDNFS Bit (LIN Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.
With 0 set, the RLIN3 interrupt is generated from the LIN/UART interface.
With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n reception status interrupt are generated from the LIN/UART interface.
For each interrupt source, see **Section 18.4, Interrupt Sources**.

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.
To use this module as an LIN slave, set these bits to "10_B" (auto baud rate) or "11_B" (fixed baud rate).

18.3.3.5 RLN3nLBFC — LIN Break Field Configuration Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

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

Table 18.35 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	LBLT	Reception Break (Low-Level) Detection Width Setting 0: A break (low-level) is detected in 9.5 or 10 Tbits 1: A break (low-level) is detected in 10.5 or 11 Tbits

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LBLT Bit (Reception Break (Low-Level) Detection Width Setting)

- When RLN3nLMD.LMD is "10_B" (LIN slave mode (auto baud rate))
  - 0: Low-level width of 10 Tbits or longer is detected.
  - 1: Low-level width of 11 Tbits or longer is detected.
- When RLN3nLMD.LMD is "11_B" (LIN slave mode (fixed baud rate))
  - 0: Low-level width of 9.5 Tbits or longer is detected.
  - 1: Low-level width of 10.5 Tbits or longer is detected.

18.3.3.6 RLN3nLSC — LIN Space Configuration Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + $0A_H$

Value after reset: 00_H

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

Table 18.36 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	IBHS[2:0]	Response Space Setting b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

This register is enabled only for response transmission, and disabled for response reception.

Some combinations of the setting values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the inter-byte space of the response transmission.
0 Tbit to 3 Tbits can be set.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS bits set the transmission width of the response space.
0 Tbit to 7 Tbits can be set.

18.3.3.7 RLN3nLWUP — LIN Wake-Up Configuration Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + $0B_H$

Value after reset: 00_H

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

Table 18.37 RLN3nLWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL bits set the low-level width of the wake-up frame transmission.
1 Tbit to 16 Tbits can be set.

18.3.3.8 RLN3nLIE — LIN Interrupt Enable Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 18.38 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SHIE	Successful Header Reception Interrupt Request Enable 0: Disables successful header reception interrupt request. 1: Enables successful header reception interrupt request.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Response/Wake-up Reception Interrupt Request Enable 0: Disables successful Response/wake-up reception interrupt request. 1: Enables successful Response/wake-up reception interrupt request.
0	FTCIE	Successful Response/Wake-up Transmission Interrupt Request Enable 0: Disables successful Response/wake-up transmission interrupt request. 1: Enables successful Response/wake-up transmission interrupt request.

Set the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

SHIE Bit (Successful Header Reception Interrupt Request Enable)

The SHIE bit enables or disables an interrupt request upon successful reception of a header.
With 0 set, the interrupt request for RLIN3n reception complete is not generated when the HTRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n reception complete is generated when the HTRC flag in the RLN3nLST register is set to 1.

ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.
With 0 set, the interrupt request for RLIN3n status is not generated when the ERR flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n status is generated when the ERR flag in the RLN3nLST register is set to 1.
Error types that are interrupt sources are the bit error, frame/response timeout error, framing error, sync filed error, ID parity error, checksum error, and response preparation error.
Detection of the bit error, frame/response timeout error, sync filed error, ID parity error, and framing error can be enabled or disabled using the RLN3nLEDE register.

FRCIE Bit (Successful Response/Wake-up Reception Interrupt Request Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a response or a wake-up frame (input signal low-level width count).
With 0 set, the interrupt request for successful RLIN3n reception is not generated when the FRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for successful RLIN3n reception is generated when the FRC flag in the RLN3nLST register is set to 1.

FTCIE Bit (Successful Response/Wake-up Transmission Interrupt Request Enable)

The FTCIE bit enables or disables an interrupt request upon successful transmission of a response or a wake-up frame.
With 0 set, the interrupt request for successful RLIN3n transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for successful RLIN3n transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

18.3.3.9 RLN3nLEDE — LIN Error Detection Enable Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + $0D_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LTES	IPERE	—	SFERE	FERE	TERE	—	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W

Table 18.39 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6	IPERE	ID Parity Error Detection Enable 0: Disables ID Parity error detection. 1: Enables ID Parity error detection.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SFERE	Sync Field Error Detection Enable 0: Disables Sync Field error detection. 1: Enables Sync Field error detection.
3	FERE	Framing Error Detection Enable*1 0: Disables framing error detection. 1: Enables framing error detection.
2	TERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BERE	Bit Error Detection Enable *1 0: Disables bit error detection. 1: Enables bit error detection.

Note 1. Set FERE bit and BERE bit to 1.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.
With 0 set, the timeout function operates as frame timeout.
With 1 set, the timeout function operates as response timeout.
For details on the timeout error, see **Section 18.7.7, Error Status**.

IPERE Bit (ID Parity Error Detection Enable)

This bit enables or disables detection of the ID parity error.
With 0 set, the ID parity error is not detected.
With 1 set, the ID parity error is detected.
When this bit is set to 1, the detection result is reflected in the IPER flag in the RLN3nLEST register.
For details on the ID parity error, see **Section 18.7.7, Error Status**.

SFERE Bit (Sync Field Error Detection Enable)

This bit enables or disables detection of the sync field error.
With 0 set, the sync field error is not detected.
With 1 set, the sync field error is detected.
Regardless of the setting of this bit, when a sync field error is detected, this module waits for the next header.
When this bit is set to 1, the detection result is reflected in the SFER flag in the RLN3nLEST register.
For details on the sync filed error, see **Section 18.7.7, Error Status**.

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
Set this bit to 1. The detection result of the framing error is indicated in the FER flag in the RLN3nLEST register.
For details on the framing error, see **Section 18.7.7, Error Status**.

TERE Bit (Timeout Error Detection Enable)

The TERE bit enables or disables detection of the frame timeout error or the response timeout error.
With 0 set, the frame timeout error or response timeout error is not detected.
With 1 set, the frame timeout error or response timeout error is detected.
When this bit is set to 1, the detection result is reflected in the TER flag in the RLN3nLEST register.
With the LTES bit, either the frame timeout error or response timeout error can be selected.
The timeout error should not be used in LIN slave mode [auto baud rate] (when the LMD[1:0] bits in the RLN3nLMD register are "10_B").
Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.
For details on the timeout error, see **Section 18.7.7, Error Status**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
Set this bit to 1. The detection result of the bit error is indicated in the BER flag in the RLN3nLEST register.
For details on the bit error, see **Section 18.7.7, Error Status**.

18.3.3.10 RLN3nLCUC — LIN Control Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + $0E_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 18.40 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode. 1: LIN operation mode.
0	OM0	LIN Reset 0: LIN reset mode. 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to transition to LIN wake-up mode or to 03_H to transition to LIN operation mode after exiting LIN reset mode.

In LIN self-test mode, set the RLN3nLCUC register to 03_H after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific LIN operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.
With 0 set, LIN/UART interface enters LIN wake-up mode.
With 1 set, LIN/UART interface enters LIN operation mode.
This bit is valid only when the OMM0 bit in the RLN3nLMST register is 1.
Writing a value to this bit is disabled while the FTS bit in the RLN3nLTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.
With 0 set, LIN/UART interface enters LIN reset mode.
With 1 set, LIN reset mode of LIN/UART interface is canceled.

18.3.3.11 RLN3nLTRC — LIN Transmission Control Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	LNRR	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 18.41 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	LNRR	No LIN Response Request 0: Response for the reception ID 1: No response for the reception ID
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped. 1: Response transmission/reception is started.
0	FTS	LIN Communication Start 0: Header reception/wake-up transmission/reception is stopped. 1: Header reception/wake-up transmission/reception is started.

LNRR Bit (No LIN Response Request)

Set this bit to 1 if no response is to be transmitted/received after receiving the header and checking the received ID.

Once set, this bit is automatically cleared to 0 upon detection of new sync field or transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 04_H using the store instruction.

Do not set this bit and the RTS bit to 1 simultaneously.

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is halted).

When response data of 9 bytes or more is to be transmitted or received, use this bit only after the completion of the header. (Do not use this bit for the second or later data group.)

RTS Bit (Response Transmission/Reception Start)

Set this bit to 1 to start response transmission or reception after receiving the header and checking the received ID.

Once set, this bit is automatically cleared to 0 upon completion of response transmission or reception (including error detection) or transition to LIN reset mode.

Only 1 can be written to this bit; 0 cannot be written.

To write 1 to this bit, write 02_H to the RLN3nLTRC register using the store instruction.

Do not set this bit and the LNRR bit to 1 simultaneously

Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

Writing a value to this bit is disabled when the FTS bit is 0 (header reception or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 upon completion of data group transmission/reception or transition to LIN reset mode.

FTS Bit (LIN Communication Start)

Set this bit to 1 to start header reception or wake-up transmission/reception.
Only 1 can be written to this bit; 0 cannot be written.
Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).

This bit is set to 0 upon completion of wake-up communication and transition to LIN reset mode.

18.3.3.12 RLN3nLMST — LIN Mode Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RLIN3n_base> +11_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 18.42 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: LIN wake-up mode. 1: LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

18.3.3.13 RLN3nLST — LIN Status Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 18.43 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Reception Flag 0: Header reception has not been completed. 1: Header reception has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	FRC	Successful Response/Wake-up Reception Flag 0: Response or wake-up reception has not been completed. 1: Response or wake-up reception has been completed.
0	FTC	Successful Response/Wake-up Transmission Flag 0: Response or wake-up transmission has not been completed. 1: Response or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00_H upon transition to LIN reset mode.

In LIN reset mode, writing a value to this register is disabled. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Successful Header Reception Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value before 1 is written. The HTRC flag is set to 1 upon completion of header reception. Here, an interrupt request for successful RLIN3n reception is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if header reception is completed while this bit is 1, an interrupt is not generated. To clear this bit to 0, write 0 to the bit.
To detect a new header in the response field upon completion of header reception, clear this bit after it is set to 1.

D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not

generated. Write 0 to clear this bit.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when at least one of the flags of the RLN3nLEST register is set to 1). Here, an interrupt request for RLIN3n status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if an error is detected while this bit is 1, an interrupt is not generated. To clear the ERR flag to 0, write 0 to the RPER, IPER, CSER, SFER, FER, TER, and BER flags in the RLN3nLEST register. This clears the ERR flag to 0.

FRC Flag (Successful Response/Wake-up Reception Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FRC flag is set to 1 upon completion of response or wake-up reception. Here, an interrupt request for successful RLIN3n reception is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if response reception or wake-up reception is completed while this bit is 1, an interrupt is not generated. Write 0 to clear this bit.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

FTC Flag (Successful Response/Wake-up Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FTC flag is set to 1 upon completion of response or wake-up transmission. Here, an interrupt request for RLIN3n transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). However, if response transmission or wake-up transmission is completed while this bit is 1, an interrupt is not generated. Write 0 to clear this bit.
When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

18.3.3.14 RLN3nLEST — LIN Error Status Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RPER	IPER	CSER	SFER	FER	TER	—	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W

Table 18.44 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	IPER	ID Parity Error Flag 0: ID parity error has not been detected. 1: ID parity error has been detected.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: Checksum error has been detected.
4	SFER	Sync Field Error Flag 0: Sync field error has not been detected. 1: Sync field error has been detected.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	TER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode.

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The RPER flag is set to 1 upon response preparation error detection. Write 0 to clear this bit.

IPER Flag (ID Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

When the IPERE bit of the RLN3nLEDE register is 1 (ID parity error detection enabled), this bit is set to 1 upon ID parity error detection. Write 0 to clear this bit.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The CSER flag is set to 1 upon checksum error detection. Write 0 to clear this bit.

SFER Flag (Sync Field Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the SFERE bit of the RLN3nLEDE register is 1 (sync field error detection enabled), this bit is set to 1 upon sync field error detection. Write 0 to clear this bit.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled), the FER flag is set to 1 upon framing error detection. Write 0 to clear this bit.

TER Flag (Timeout Error Flag)

Only 0 can be written to the TER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the TERE bit of the RLN3nLEDE register is 1 (frame/response timeout error detection enabled), this flag is set to 1 upon frame timeout error or response timeout error detection. Write 0 to clear this bit.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled), the BER flag is set to 1 upon bit error detection. Write 0 to clear this bit.

18.3.3.15 RLN3nLDFC — LIN Data Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LSS	—	LCS	RCDS	RFDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.45 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Data transmission/reception continues without waiting for reception of the next header.)
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LCS	Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode
4	RCDS	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 byte (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited.

LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received. With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one. With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.
When multi-byte response transmission/reception function is not used, set this bit to “0”.
Set this bit when the RTS bit is 0 (response transmission/reception stopped).

LCS Bit (Checksum Select)

The LCS bit sets the checksum mode.
With 0 set, classic checksum mode is selected.
With 1 set, enhanced checksum mode is selected.
When the timeout error is used (the TERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details on the bit error, see **Section 18.7.7, Error Status**.
When the length of the response field data is 0 bytes (the RFDL bit is 0), do not set this bit to “1” (enhanced).

When response data of 9 bytes or more is to be transmitted or received, do not change the LCS bit setting after the first data group through the last data group.
During transmission or reception of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.
Set this bit when the RTS bit is 0 (response transmission/reception stopped).

RCDS Bit (Response Field Communication Direction Select)

This bit selects the direction of the response field/wake-up signal communication.
With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (input signal low-level width count).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.
Set this bit when the RTS bit in the RLN3nLTRC register is 0 in LIN operation mode (response transmission/reception stopped) or when the FTS bit is 0 in LIN wake-up mode (header reception or wake-up transmission/reception stopped).
When response data of 9 bytes or more is to be transmitted or received, do not change this bit setting after the first data group through the last data group.

RFDL[3:0] Bits (Response Field Length Select)

The RFDL bits set the length of the response field data.
The data length can be 0 to 8 bytes excluding the checksum size.
Set these bits when the RTS bit in the RLN3nLTRC register is 0 (response transmission/reception stopped).
When response data of 9 bytes or more is to be transmitted or received, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit in the RLN3nLDFC register is 1) include the checksum.

18.3.3.16 RLN3nLIDB — LIN ID Buffer Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.46 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7, 6	IDP[1:0]	Parity Setting Stores the parity bits (P0 and P1) received in the ID field.
5 to 0	ID[5:0]	ID Setting Stores the 6-bit ID value received in the ID field.

The value in the RLN3nLIDB register is enabled after the completion of header reception. In LIN mode (LIN operation mode, LIN wake-up mode), writing to this register is disabled.

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted before communication. After completion of frame transmission/ reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 18.9, LIN Self-Test Mode**.

IDP[1:0] Bits (Parity Setting)

The IDP bits store the parity bits (P0 and P1) received in the ID field of the LIN frame. IDP0 is for P0 and IDP1 is for P1.

When the IPERE bit in the RLN3nLEDE register is 1 (ID parity detection enabled), the received value and the value calculated internally are checked. If they do not match, IPER (ID parity error flag) is set.

ID[5:0] Bits (ID Setting)

The ID bits store the 6-bit ID value received in the ID field of the LIN frame.

18.3.3.17 RLN3nLCBR — LIN Checksum Buffer Register

Access: This register is a read-only register that can be read in 8-bit units. However, in LIN self-test mode, this register can be read and written in 8-bit units.

Address: <RLIN3n_base> + 16_H

Value after reset: 00_H

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

Table 18.47 RLN3nLCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the transmitted or received checksum data.

In LIN operation mode, this register operates as follows:

- When the RCDS bit in the RLN3nLDFC register is 1 (transmission):
  The value transmitted can be read from the register. Read the value after transmission is completed.
  Writing to this register is invalid.
- When the RCDS bit in the RLN3nLDFC register is 0 (reception):
  The value received can be read from the register. Read the value after reception is completed.
  Writing to this register is invalid.

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for the other data groups.

In LIN self-test mode, this register operates as follows:

- When the RCDS bit in the RLN3nLDFC register is 1 (transmission):
  After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RCDS bit in the RLN3nLDFC register is 0 (reception):
  Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 18.9, LIN Self-Test Mode**.

Set the RLN3nLCBR register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

18.3.3.18 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

Access: This register can be read or written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H
RLN3nLDBR2: <RLIN3n_base> + 19_H
RLN3nLDBR3: <RLIN3n_base> + $1A_H$
RLN3nLDBR4: <RLIN3n_base> + $1B_H$
RLN3nLDBR5: <RLIN3n_base> + $1C_H$
RLN3nLDBR6: <RLIN3n_base> + $1D_H$
RLN3nLDBR7: <RLIN3n_base> + $1E_H$
RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Value after reset: 00_H

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

Table 18.48 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or holds the received data. Setting range: 00_H to FF_H

- For response transmission:
  The RLN3nLDBRb registers set the data to be transmitted in the response field.
  Set these registers when the RTS bit in the RLN3nLTRC register is 0 (response transmission/reception is halted).
- For response reception:
  The RLN3nLDBRb registers hold the data received in the response field.
  The received data is overwritten. If an error is detected, the data prior to reception interruption is stored in the register.
  Do not read these registers when the RTS bit is 1 (response transmission/reception is started)

In LIN self-test mode, the operation is as follows.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see **Section 18.9, LIN Self-Test Mode**.

18.3.4 UART Related Registers

18.3.4.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

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

Table 18.49 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 samplings 0 1 0 1: 6 samplings 0 1 1 0: 7 samplings 0 1 1 1: 8 samplings 1 0 0 0: 9 samplings 1 0 0 1: 10 samplings 1 0 1 0: 11 samplings 1 0 1 1: 12 samplings 1 1 0 0: 13 samplings 1 1 0 1: 14 samplings 1 1 1 0: 15 samplings 1 1 1 1: 16 samplings Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 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
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB bits select the number of sampling in one Tbit (reciprocal of the baud rate).
In UART mode, it is possible to set the NSPB bits from 6 samplings to 16 samplings.

LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS bits select the frequency division ratio for the prescaler.
The LIN communication clock source is divided by this prescaler.

18.3.4.2 RLN3nLBRP01 — UART Baud Rate Prescaler 01 Register

Access: RLN3nLBRP01 register can be read or written in 16-bit units.
RLN3nLBRP0 register can be read or written in 8-bit units.
RLN3nLBRP1 register can be read or written in 8-bit units.

Address: RLN3nLBRP01: <RLIN3n_base> + 02_H
RLN3nLBRP0: <RLIN3n_base> + 02_H
RLN3nLBRP1: <RLIN3n_base> + 03_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.50 RLN3nLBRP01 Register Contents

Bit Position	Bit Name	Function
15 to 0	BRP[15:0]	Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000_H to $FFFF_H$

Set the RLN3nLBRP01 register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS bits (prescaler clock select bits) in the RLN3nLWBR register by L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units by the registers RLN3nLBRP0 and RLN3nLBRP1.

18.3.4.3 RLN3nLMD — UART Mode Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 08_H

Value after reset: 00_H

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

Table 18.51 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	LRDNFS	UART Reception Data Noise Filter Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 0 1: UART mode

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

LRDNFS Bit (UART Reception Data Noise Filter Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD bits select the LIN/UART interface mode.
To use the LIN/UART interface as an UART, set these bits to 01_B.

18.3.4.4 RLN3nLBFC — UART Configuration Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	UTPS	URPS	UPS[1:0]		USBLS	UBOS	UBLS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.52 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	UTPS	UART Output Polarity Switch 0: Transmit data normal output 1: Transmit data with inverted output
5	URPS	UART Input Polarity Switch 0: Reception data normal output 1: Reception data inverted output
4, 3	UPS[1:0]	UART Parity Select 00: Parity disabled 01: Even parity 10: 0 Parity 11: Odd parity
2	USBLS	UART Stop Bit length Select 0: Stop bit:1 bit 1: Stop bit: 2 bits
1	UBOS	UART Transfer Format Order Select 0: LSB First 1: MSB First
0	UBLS	UART Character Length Select 0: UART 8 bits communication 1: UART 7 bits communication

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

UTPS Bit (UART Output Polarity Switch)

Sets the output polarity for UART communication.
With 0 set, transmit data is output without inversion.
With 1 set, inverted transmit data is output.
The setting of this bit is valid for all the bits of the UART frame.
In half-duplex communication, this setting should match the setting of URPS bit.

URPS Bit (UART Input Polarity Switch)

Sets the input polarity for UART communication.
With 0 set, receive data is input without inversion.
With 1 set, receive data is input with inversion.
The setting of this bit is valid for all the bits of the UART frame.
In half-duplex communication, this setting should match the setting of UTPS bit.
When setting this bit to “1” and expansion bit reception ((with expansion bit comparison) or (with data comparison)) is performed, set the inverse of the expected value to the UEBDL bit in the

RLN3nLUOR1 register and RLN3nLIDB register to enable comparison of the inverted values of the received values.

UPS[1:0] Bits (UART Parity Select)

Sets the UART parity.

- When these bits are set to “00_B”, data is transmitted/received without the parity.

  [Transmission]

  A parity bit is not added to transmit data.

  [Reception]

  Data is received without parity processing. Therefore, a parity error does not occur.

- When these bits are set to “01_B”, data is transmitted/received with the even parity.

  [Transmission]

  If the number of 1s in transmit data is odd, “1” is added to the parity bit. If the number of 1s in transmit data is even, “0” is added to the parity bit.

  [Reception]

  If the number of 1s in receive data including the parity bit is odd, a parity error occurs.

- When these bits are set to “10_B”, data is transmitted/received with 0 parity.

  [Transmission]

  Regardless of the number of 1s in transmit data, “0” is added to the parity bit.

  [Reception]

  The value of the parity bit is not judged. Therefore, no parity error occurs.

- When these bits are set to “11_B”, data is transmitted/received with the odd parity.

  [Transmission]

  If the number of 1s in transmit data is odd, “0” is added to the parity bit. If the number of 1s in transmit data is even, “1” is added to the parity bit.

  [Reception]

  If the number of 1s in receive data including the parity bit is even, a parity error occurs.

USBLS Bit (UART Stop Bit Length Select)

Sets the stop bit length of data for UART communication.
With 0 set, stop bit length of 1 bit is selected.
With 1 set, stop bit length of 2 bits is selected.

UBOS Bit (UART Transfer Format Select)

Sets the bit order of data for UART communication.
With 0 set, LSB first is selected.
With 1 set, MSB first is selected.

UBLS Bit (UART Character Length Select)

Sets the character length of one frame for UART communication.
With 0 set, the character length is 8 bits.
With 1 set, the character length is 7 bits.
When the character length of one frame is 9 bits (the UEBE bit in the RLN3nLUOR1 register is 1), the setting of this bit is invalid.

18.3.4.5 RLN3nLSC — UART Space Configuration Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + $0A_H$

Value after reset: 00_H

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

Table 18.53 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the space between the UART frames when transmitting data from the UART buffer.
0 to 3 Tbits can be set.
Set IBS[1:0] bits to "00_B" when UART buffer is not used.

When data is transferred from the UART transmission data register (RLN3nLUTDR) and the UART wait transmission data register (RLN3nLUWTDR), the setting of these bits is ignored.Set these bits to "00_B".

18.3.4.6 RLN3nLEDE —UART Error Detection Enable Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + $0D_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FERE	OERE	—	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R/W

Table 18.54 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	OERE	Overrun Error Detection Enable 0: Disables overrun error detection. 1: Enables overrun error detection.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (LIN reset mode).

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
When this bit is set to 1, the detection result is indicated in the FER flag in the RLN3nLEST register.
For details on the framing error, see **Section 18.8.5, Error Status**.

OERE Bit (Overrun Error Detection Enable)

This bit enables or disables detection of the overrun error.
With 0 set, the overrun error is not detected.
With 1 set, the overrun error is detected.
When this bit is set to 1, the detection result is reflected in the OER flag in the RLN3nLEST register.
For details on the overrun error, see **Section 18.8.5, Error Status**.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
When this bit is set to 1, the detection result is indicated in the BER flag in the RLN3nLEST register.
In full-duplex communication, do not set this bit to “1”.
Do not set this register when the NSPB bits in the RLN3nLWBR register are 0101_B (6 samplings) and the LRDNFS bit in the RLN3nLMD register is 0 (noise filtering is enabled).

For details on the bit error, see **Section 18.8.5, Error Status**.

18.3.4.7 RLN3nLCUC — UART Control Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + $0E_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 18.55 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OM0	LIN Reset 0: LIN reset mode. 1: LIN reset mode is canceled.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to reset mode or canceling reset mode.
With 0 set, reset mode.
With 1 set, reset mode is canceled.

18.3.4.8 RLN3nLTRC — UART Transmission Control Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

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

Table 18.56 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RTS	UART Buffer Transmission Start 0: UART Buffer transmission is stopped. 1: UART Buffer transmission is started.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

RTS Bit (UART Buffer Transmission Start)

When transmitting data from the UART buffer, set this bit to “1”.
Only 1 can be written to this bit; 0 cannot be written.
Write to this bit when the UTOE bit in the RLN3nLUOER register is 1 (transmission enabled) and the UTS bit in the RLN3nLST register is 0 (transmission is not in progress).
Once set, regardless of errors, this bit is automatically cleared to 0 upon completion of the number of data transmission specified by the MDL bit in the RLN3nLDFC register. Moreover, this bit is automatically cleared to 0 upon transition to reset mode.
Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
When writing 1 to this bit while the UTSW bit in the RLN3nLDFC register is 1 (when UART buffer transmission is requested, the start of transmission is delayed until the stop bit of reception data is completed), write only during the reception of stop bit.

18.3.4.9 RLN3nLMST — UART Mode Status Register

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RLIN3n_base> + 11_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 18.57 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	OMM0	LIN Reset Status Monitor 0: LIN reset mode. 1: Not in LIN reset mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

18.3.4.10 RLN3nLST — UART Status Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	URS	UTS	ERR	—	—	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 18.58 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	URS	Reception Status Flag 0:Reception is not operated. 1:Reception is operated.
4	UTS	Transmission Status Flag 0: Transmission is not operated. 1: Transmission is operated.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2, 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	FTC	Successful UART Buffer Transmission Flag 0: UART buffer transmission has not been completed. 1: UART buffer transmission has been completed.

The RLN3nLST register is automatically cleared to "00_H" upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains "00_H". To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

URS Flag (Reception Status Flag)

At the start of the reception, this flag is set to 1.
The reception is started under the following condition.

- When the start bit is detected

At the end of reception, this flag is cleared to 0. While reception is stopped, this flag retains 0.
The reception is ended under the following condition.

- Sampling point of the first bit of the stop bits

UTS Flag (Transmission Status Flag)

At the start of the transmission, this flag is set to 1. During the transmission, this flag retains 1.
The transmission is started under the following conditions.

- When transmission data is set to the RLN3nLUTDR or RLN3nLUWTDR register
- When the RTS bit in the RLN3nLTRC register is set to 1

This flag is cleared to 0 at the end of transmission.
The transmission is ended under the following conditions.

- When transmission of data set in the RLN3nLUTDR or RLN3nLUWTDR register is completed and next data is not set
- When transmission from UART buffer is completed (when the RTS bit in the RLN3nLTRC register is cleared to 0)

ERR Flag (Error Detection Flag)

This flag is set to 1 upon detection of an error, detection of an expansion bit, or upon ID matching (when at least one of the flags of the RLN3nLEST register is 1). At this time, an interrupt request for RLIN3n status is generated. However, when this bit is 1, an interrupt is not generated upon detection of an error, detection of an expansion bit, or upon ID matching. To clear the bit to 0, write 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register.

FTC Flag (Successful UART Buffer Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that it was before 1 was written.
Regardless of errors, this bit is set to 1 upon completion of transmission of the number of data specified by the MDL bit in the RLN3nLDFC register from the UART buffer. At this time, an interrupt request for RLIN3n transmission is generated. Write 0 to clear this flag.

18.3.4.11 RLN3nLEST — UART Error Status Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	UPER	IDMT	EXBT	FER	OER	—	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W

Table 18.59 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	UPER	Parity Error Flag 0: Parity error has not been detected. 1: Parity error has been detected.
5	IDMT	ID Matching Flag 0: The receive data does not match with the ID value. 1: The receive data matches with the ID value.
4	EXBT	Expansion Bit Detection Flag 0: Expansion Bit has not been detected. 1: Expansion Bit has been detected.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	OER	Overrun Error Flag 0: Overrun error has not been detected. 1: Overrun error has been detected.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to, and the value of 00_H is held. To clear certain bits in this register, write 0 to those bits, and write 1 to the bits not to be cleared by using the store instruction.

UPER Flag (Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.
This flag is set to 1 upon parity error detection. Write 0 to clear this flag.

IDMT Flag (ID Matching Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.

The IDMT flag becomes 1 when all the following conditions are met:

- The UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enabled)
- The UECD bit in the RLN3nLUOR1 register is 0 (expansion bit comparison enabled)

- The UEBDCE bit in the RLN3nLUOR1 register is 1 (data comparison after expansion bit is detected)
  - The received expansion bit and the value of the UEBDL bit of the RLN3nLUOR1 register are match.
  - The value of the 8-bit of the received data excluding the expansion bit and the value of the RLN3nLIDB register match.

Write 0 to clear this flag.

EXBT Flag (Expansion Bit Detection Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.
When the UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enable), if the received expansion bit matches the UEBDL bit in the RLN3nLUOR1 register, this flag is set to 1.
Write 0 to clear this flag.

FER Flag (Framing Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that it was before 1 was written.
The FER flag is set to 1 upon framing error detection while the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled). Write 0 to clear this flag.

OER Flag (Overrun Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value.
The OER flag is set to 1 upon overrun error detection while the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). Write 0 to clear this flag.

BER Flag (Bit Error Flag)

Only 0 can be written to the this flag; when 1 is written, the bit retains the value that it was before 1 was written.
The BER flag is set to 1 when the transmitted data and the data monitored by the receive pin do not match while the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled).
Write 0 to clear this flag.

18.3.4.12 RLN3nLDFC — UART Data Field Configuration Register

Access: This register can be read or written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

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

Table 18.60 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	UTSW	Transmission Start Wait 0: When UART buffer transmission is requested, transmission is started immediately. 1: When UART buffer transmission is requested, transmission is not started until reception of the stop bit is completed.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	MDL[3:0]	UART Buffer Data Length Select b3 b0 0 0 0 0: 9 data 0 0 0 1: 1 data 0 0 1 0: 2 data 0 0 1 1: 3 data 0 1 0 0: 4 data 0 1 0 1: 5 data 0 1 1 0: 6 data 0 1 1 1: 7 data 1 0 0 0: 8 data 1 0 0 1: 9 data Settings other than the above are prohibited.

UTSW Bit (Transmission Start Wait)

This bit controls the transmission start timing of UART buffer.
With 0 set, transmission is started as soon as the start of UART buffer transmission is requested.
With 1 set, transmission is started after the completion of the stop bit reception.
Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.
This bit is enabled when the RTS bit in the RLN3nLTRC register is set to 1. In addition, writing a value to this bit is disabled when the RTS bit is 1 (UART buffer transmission started).
Set this bit to 1 only to switch from reception to transmission in half-duplex communication.

MDL[3:0] Bits (UART Buffer Data Length Select)

These bits specify the data length of the UART buffer.
Writing a value to these bits is disabled when the RTS bit in the RLN3nLTRC register is 1 (UART buffer transmission started).

18.3.4.13 RLN3nLIDB — UART ID Buffer Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

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

Table 18.61 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7 to 0	ID[7:0]	ID value that is referred for the expansion bit data comparison is set

ID Bits (ID)

When the UEBE bit in the RLN3nLUOR1 register is set to 1 (expansion bit enabled), the UECD bit is set to 0 (expansion bit comparison enabled), and the UEBDCE bit is set to 1 (data comparison after expansion bit is detected), set the value to be compared with the received data. Write to the RLN3nLIDB register when the URS bit in the RLN3nLST register is 0 (receive operation is not in progress).

18.3.4.14 RLN3nLUDB0 — UART Data Buffer 0 Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 17_H

Value after reset: 00_H

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

Table 18.62 RLN3nLUDB0 Register Contents

Bit Position	Bit Name	Function
7 to 0	UDB[7:0]	Sets the data to be transmitted. Setting range: 00_H to FF_H

When transmitting 9-byte data from the UART buffer (the RLN3nLDFC.MDL bit is "0_H" or" 9_H"), set the first data to be transmitted.

Write to the RLN3nLUDB0 register when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 18.63, Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format, shows the bit arrangement according to the settings for communication format.

For details about the UART buffer, see **Section 18.8.1.2, UART Buffer Transmission**, **(1) UART Buffer Transmission**.

Table 18.63 Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format

	RLN3nLUDB0							
	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

18.3.4.15 RLN3nLDBRb — UART Data Buffer b Register (b = 1 to 8)

Access: This register can be read or written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> +18_H
RLN3nLDBR2: <RLIN3n_base> + 19_H
RLN3nLDBR3: <RLIN3n_base> + $1A_H$
RLN3nLDBR4: <RLIN3n_base> + $1B_H$
RLN3nLDBR5: <RLIN3n_base> + $1C_H$
RLN3nLDBR6: <RLIN3n_base> + $1D_H$
RLN3nLDBR7: <RLIN3n_base> + $1E_H$
RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Value after reset: 00_H

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

Table 18.64 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted. Setting range: 00_H to FF_H

This register specifies the data transmitted from the UART buffer.

Write to these registers when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 18.65, Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format, shows the bit arrangement according to the set communication format.

For details about the UART buffer, see **Section 18.8.1.2, UART Buffer Transmission**, **(1) UART Buffer Transmission**.

Table 18.65 Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format

	RLN3nLDBRb							
	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7-bit; MSB first	—	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
8-bit; LSB first	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
8-bit; MSB first	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

18.3.4.16 RLN3nLUOER — UART Operation Enable Register

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 20_H

Value after reset: 00_H

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

Table 18.66 RLN3nLUOER Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	UROE	Reception Enable 0: Disables reception. 1: Enables reception.
0	UTOE	Transmission Enable 0: Disables transmission. 1: Enables transmission.

The RLN3nLUOER register is automatically cleared to 00_H upon transition to LIN reset mode.
In LIN reset mode, this register cannot be written to.
In LIN reset mode, the register retains 00_H.

UROE Bit (Reception Enable)

The UROE bit enables or disables reception.
With 0 set, reception is disabled.
With 1 set, reception is enabled.
Do not clear this bit during reception. If the communication is suspended during reception, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to transition to the LIN reset mode. However, the transmit operation is also suspended at this time.

Do not set this bit to 1 when data transmission from the UART buffer is in progress.

UTOE Bit (Transmission Enable)

The UTOE bit enables or disables transmission.
With 0 set, transmission is disabled.
With 1 set, transmission is enabled.
Do not clear this bit during transmission. If the communication is suspended during transmission, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to transition to the LIN reset mode. However, the receive operation is also suspended at this time.

18.3.4.17 RLN3nLUOR1 — UART Option Register 1

Access: This register can be read or written in 8-bit units

Address: <RLIN3n_base> + 21_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	UECD	UTIGTS	UEBDCE	UEBDL	UEBE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 18.67 RLN3nLUOR1 Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	UECD	Expansion Bit Comparison Disable 0: Enables expansion bit comparison. 1: Disables expansion bit comparison.
3	UTIGTS	Transmission Interrupt Generation Timing Select 0: Transmission interrupt is generated at the start of transmission. 1: Transmission interrupt is generated at the completion of transmission.
2	UEBDCE	Expansion Bit Data Comparison Enable 0: Disables data comparison after an expansion bit is detected. 1: Enables data comparison after an expansion bit is detected.
1	UEBDL	Expansion Bit Detection Level Select 0: Selects expansion bit value 0 as the expansion bit detection level. 1: Selects expansion bit value 1 as the expansion bit detection level.
0	UEBE	Expansion Bit Enable 0: Disables expansion bit operation. 1: Enables expansion bit operation.

UECD Bit (Expansion Bit Comparison Disable)

The UECD bit enables or disables comparison between the received expansion bit and the UEBDL bit value when the UEBE bit is 1 (expansion bit operation is enabled).
With 0 set, comparison between the received expansion bit and the UEBDL bit value is enabled when the expansion bit is received.
With 1 set, comparison between the received expansion bit and the UEBDL bit value is disabled when the expansion bit is received.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.
Do not set this bit to 1 when the UEBDCE bit is 1 (data comparison after expansion bit is detected).

UTIGTS Bit (Transmission Interrupt Generation Timing Select)

The UTIGTS bit sets the generation timing of the transmission interrupt.
With 0 set, the transmission interrupt is generated at the start of transmission.
With 1 set, the transmission interrupt is generated at the completion of transmission.
When transmission from the UART buffer is performed with 0 set, the transmission interrupt is generated only at the start of the transmission of the last data of the data length set with the MDL bits in the RLN3nLDFC register.
When transmission from the UART buffer is performed with 1 set, the transmission interrupt is

generated only at the completion of the transmission of the last data of the data length set with the MDL bits in the RLN3nLDFC register.

UEBDCE Bit (Expansion Bit Data Comparison Enable)

After an expansion bit is detected, this bit enables or disables the comparison between the 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register.
With 0 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is disabled.
With 1 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the received value in the RLN3nLURDR register and the value of the RLN3nLIDB register is enabled.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Do not set this bit to 1 when the UEBE bit is 0 (expansion bit operation disabled).
Do not set this bit to 1 when the UECD bit is 1 (expansion bit comparison disabled).
Do not set this bit to 1 when the UART buffer is used.

UEBDL Bit (Expansion Bit Detection Level Select)

The UEBDL bit sets the level to be detected as the expansion bit when the UEBE bit is 1 (expansion bit operation is enabled) and the UECD bit is 0 (comparison of the expansion bit is enabled).
With 0 set, expansion bit value 0 is the level to be detected as the expansion bit.
With 1 set, expansion bit value 1 is the level to be detected as the expansion bit.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.

UEBE Bit (Expansion Bit Enable Bit)

The UEBE bit enables or disables expansion bit operation.
With 0 set, expansion bit operation is disabled.
With 1 set, expansion bit operation is enabled.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.

18.3.4.18 RLN3nLUTDR — UART Transmission Data Register

Access: RLN3nLUTDR register can be read or written in 16-bit units.
RLN3nLUTDRL register can be read or written in 8-bit units.
RLN3nLUTDRH register can be read or written in 8-bit units.

Address: RLN3nLUTDR: <RLIN3n_base> + 24_H
RLN3nLUTDRL: <RLIN3n_base> + 24_H
RLN3nLUTDRH: <RLIN3n_base> + 25_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UTD[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.68 RLN3nLUTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	UTD[8:0]	Sets the data to be transmitted. Setting range: 000_H to $1FF_H$

The RLN3nLUTDR register sets the data to be transmitted from the transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

This register can be accessed in 8 bits.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

Also, do not write data to this register when a transmission request is being generated due to write access to the RLN3nLUWTDR register.

When transmitting data continuously, do not set another piece of transmission data in this register before the generation of transmission interrupt.

The table below shows the bit arrangement according to the set communication format.

Table 18.69 Bit Arrangement of the RLN3nLUTDR Register According to Each Communication Format

	RLN3nLUTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

18.3.4.19 RLN3nLURDR — UART Reception Data Register

Access: RLN3nLURDR register is a read-only register that can be read in 16-bit units.
RLN3nLURDRL register is a read-only register that can be read in 8-bit units.
RLN3nLURDRH register is a read-only register that can be read in 8-bit units.

Address: RLN3nLURDR: <RLIN3n_base> + 26_H
RLN3nLURDRL: <RLIN3n_base> + 26_H
RLN3nLURDRH: <RLIN3n_base> + 27_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	URD [8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.70 RLN3nLURDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned.
8 to 0	URD [8:0]	Stores the received data Setting range: 000_H to $1FF_H$

The RLN3nLURDR allows the reception data to be read from the receive data register.

When the UROE bit in the RLN3nLUOER register is 1, the reception data is stored in this register and can be read out.

This register is updated when the stop bit of the receive data is received.

This register is also updated when an error is caused by the parity or stop bit.

However, the value of this register is not updated upon occurrence of an overrun error when the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). The value of this register is updated upon occurrence of an overrun error when the OERE bit is 0 (overrun error detection disabled).

Read this register upon occurrence of a receive error (overrun error, framing error, parity error) when the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). If the next data is received without reading this register, an overrun error occurs.

This register can be accessed in 8 bits.

However, during expansion bit use (UEBE bit of the RLN3nLUOR1 register is 1 (expansion bit operation enabled), do not attempt 8-bit access.

The table below shows the bit arrangement according to the set communication format.

Table 18.71 Bit Arrangement of the RLN3nLURDR Register According to Each Communication Format

	RLN3nLURDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

18.3.4.20 RLN3nLUWTDR — UART Wait Transmission Data Register

Access: RLN3nLUWTDR register can be read or written in 16-bit units.
RLN3nLUWTDRL register can be read or written in 8-bit units.
RLN3nLUWTDRH register can be read or written in 8-bit units.

Address: RLN3nLUWTDR: <RLIN3n_base> + 28_H
RLN3nLUWTDRL: <RLIN3n_base> + 28_H
RLN3nLUWTDRH: <RLIN3n_base> + 29_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UWTD[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.72 RLN3nLUWTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	UWTD[8:0]	Sets the data to be transmitted from the wait transmit data register after waiting for the stop bit reception to be completed. Setting range: 000_H to $1FF_H$

The RLN3nLUWTDR register sets the data to be transmitted from the UART wait transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

Use this register only to switch from reception to transmission in half-duplex communication.

The user should write to this register only while the stop bit is being received.

Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.

When this register is read, the RLN3nLUTDR register value is actually read.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

The table below shows the bit arrangement according to the set communication format.

Table 18.73 Bit Arrangement of the RLN3nLUWTDR Register According to Each Communication Format

	RLN3nLUWTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

18.4 Interrupt Sources

The LIN/UART interface generates four types of interrupt requests.

- RLIN3n transmission interrupt
- RLIN3n successful reception interrupt
- RLIN3n status interrupt
- RLIN3n interrupt

Setting the LIOS bit in the RLN3nLMD register to 0 allows to perform logical OR operation on all of the interrupt sources, outputting the interrupt request from the RLIN3n interrupt.

Setting the LIOS bit in the RLN3nLMD register to 1 allows to output the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, or RLIN3n status interrupt depending on the interrupt source.

Table 18.74 lists the sources for each interrupt.

Table 18.74 Interrupt Sources

		LIOS bit in RLN3nLMD register is 0	LIOS bit in RLN3nLMD register is 1*1		
		RLIN3n Interrupt	RLIN3n Transmission Interrupt	RLIN3n Successful Reception Interrupt	RLIN3n Status Interrupt
LIN mode	LIN master mode	• Successful frame transmission • Successful frame reception • Successful wake-up transmission • Successful wake-up reception • Successful header transmission • Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error	• Successful frame transmission • Successful wake-up transmission • Successful header transmission	• Successful wake-up reception • Successful wake-up reception	• Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error
	LIN slave mode	• Successful response transmission • Successful response reception • Successful wake-up transmission • Successful wake-up reception • Successful header reception • Bit error • Frame/response timeout error • Framing error • Sync field error • Checksum error • ID parity error • Response preparation error	• Successful response transmission • Successful wake-up transmission	• Successful response reception • Successful wake-up reception • Successful header reception	• Bit error • Frame/response timeout error • Framing error • Sync field error • Checksum error • ID parity error • Response preparation error
UART mode		—	• Transmission start/ successful transmission	• Successful reception • Expansion bit mismatch	• Bit error • Overrun error • Framing error • Expansion bit match • ID match • Parity error

Note 1. The LIOS bit setting is valid in LIN Mode. In UART mode, setting the LIOS bit is not required.

In LIN mode, each interrupt request is output when the corresponding bit in the RLN3nLIE register is 1 (interrupt is enabled) and the corresponding flag in the RLN3nLST register is 1.

18.5 Modes

The LIN/UART interface provides the following four modes, depending upon the specific function to be performed:

- LIN reset mode
- LIN mode
  - LIN master mode
  - LIN slave mode [auto baud rate]
  - LIN slave mode [fixed baud rate]
- UART mode
- LIN self-test mode

Figure 18.2 shows mode transitions. **Table 18.75** describes mode transition conditions. **Table 18.76** lists operations available in each mode.

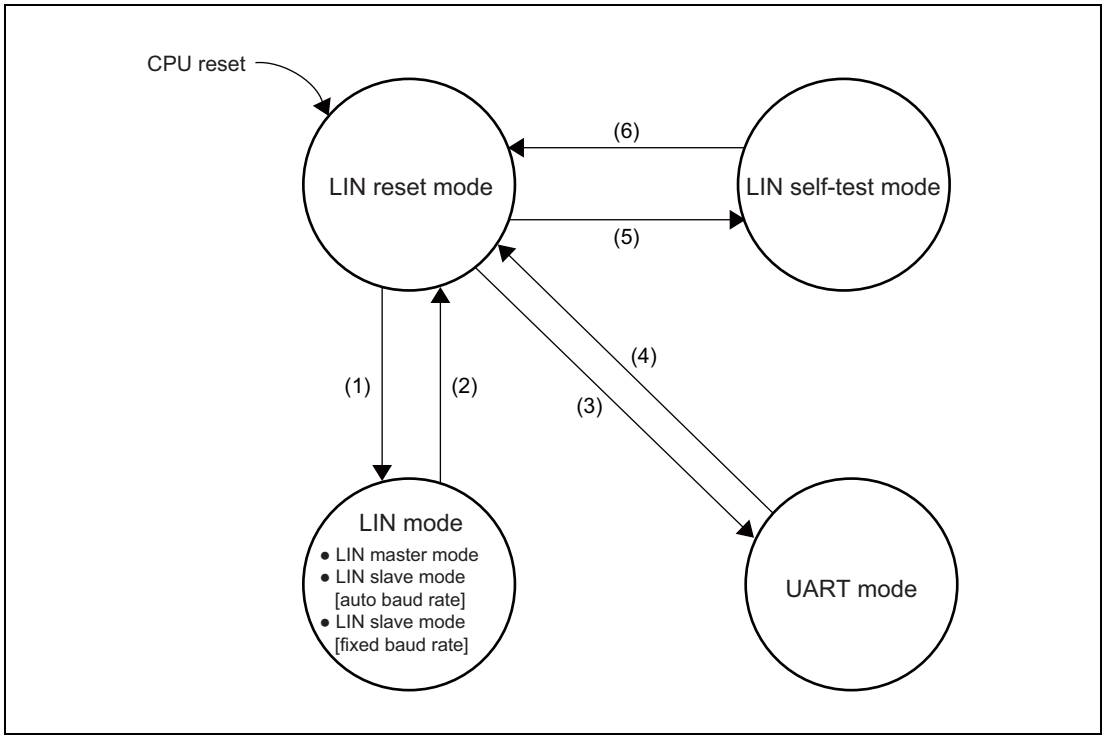


Figure 18.2 Mode Transitions

Table 18.75 Transition Condition of Each Mode

	Mode Transition			Transition Condition
(1)	LIN reset mode	→	LIN mode	
			• LIN master mode	• LMD bits in RLN3nLMD register = 00_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B or 11_B
			• LIN slave mode [auto baud rate]	• LMD bits in RLN3nLMD register = 11_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B or 11_B
			• LIN slave mode [fixed baud rate]	• LMD bits in RLN3nLMD register = 10_B and OM1 and OM0 bits of RLN3nLCUC register = 01_B or 11_B
(2)	LIN mode	→	LIN reset mode	OM0 bit in RLN3nLCUC register = 0_B
(3)	LIN reset mode	→	UART mode	LMD bits in RLN3nLMD register = 01_B and OM0 bit in RLN3nLCUC register = 1_B
(4)	UART mode	→	LIN reset mode	OM0 bit in RLN3nLCUC register = 0_B
(5)	LIN reset mode	→	LIN self-test mode	See **Section 18.9, LIN Self-Test Mode**.
(6)	LIN self-test mode	→	LIN reset mode	See **Section 18.9, LIN Self-Test Mode**.

Table 18.76 Operations Available in Each Mode

LIN Mode			
LIN Master Mode	**LIN Slave Mode [auto baud rate] LIN Slave Mode [fixed baud rate]**	**UART Mode**	**LIN Self-Test Mode**
Header transmission Response transmission Response reception Wake-up transmission Wake-up reception Error detection	Header reception Response transmission Response reception Wake-up transmission Wake-up reception Error detection	UART transmission UART reception Error detection	Self test

Whether mode has transitioned to LIN reset mode, the LIN mode, or the UART mode can be verified by reading the LMD bits in the RLN3nLMD register and the OMM0 bit in the RLN3nLMST register.

For a description of the LIN self-test mode, see **Section 18.9, LIN Self-Test Mode**.

18.6 LIN Reset Mode

Setting the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) causes a transition to LIN reset mode. The change to LIN reset mode can be verified by determining that the OMM0 bit in the RLN3nLMST register has been set to 0 (LIN reset mode). In this mode, the LIN communication and the UART communication functions are halted.

From LIN reset mode, transitions to LIN mode, UART mode, and LIN self-test mode can be made.

When the mode changes to LIN reset mode, the following registers are initialized to their reset values, and as long as LIN reset mode is in effect, they retain their initial values.

- RLN3nLTRC register
- RLN3nLST register
- RLN3nLEST register
- RLN3nLUOER register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN3nLWBR register
- RLN3nLBRP0 register
- RLN3nLBRP1 register
- RLN3nLMD register
- RLN3nLBFC register
- RLN3nLSC register
- RLN3nLWUP register
- RLN3nLIE register
- RLN3nLEDE register
- RLN3nLDFC register
- RLN3nLIDB register
- RLN3nLCBR register
- RLN3nLUDB0 register
- RLN3nLDBRb register (b = 1 to 8)
- RLN3nLUOR1 register
- RLN3nLUTDR register
- RLN3nLURDR register
- RLN3nLUWTDR register

18.7 LIN Mode

LIN mode can operate in the following submodes: LIN master mode, LIN slave mode [auto baud rate], and LIN slave mode [fixed baud rate].

In LIN master mode, the following operations can be performed: header transmission, response transmission, response reception, wake-up transmission, wake-up reception, and error detection. In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 00_B (LIN master mode) and the OM1 and OM0 bits in the RLN3nLCUC register to either 01_B or 11_B sets LIN master mode, turning the OMM1 and OMM0 bits in the RLN3nLMST register to either 01_B to 11_B.

In LIN slave mode [auto baud rate] and LIN slave mode [fixed baud rate], header reception, response transmission, response reception, wake-up transmission, wake-up reception, and error detection can be performed.

The LIN slave mode [auto baud rate] allows automatic detection of the break field and the sync field, and sets a baud rate based on the results of measurement of a sync field. The baud rate can be set to 1 kbps to 20 kbps.

Set the LPRS[2:0] bits in the RLN3nLWBR register so that the prescaler clock (with the frequency of the LIN communication clock source divided by the prescaler) becomes as follows according to the target baud rate.

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4MHz*1
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

Note 1. Use the clock with NSPB[3:0] bits in the RLN3nLWBR register set to "0011_B" (four samplings).

LIN slave mode [fixed baud rate] allows automatic detection of the break field, sync field, and ID field at a baud rate that is set in advance by the baud rate generator.

In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 10_B (LIN slave mode [auto baud rate]) and setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B or 11_B sets LIN slave mode [auto baud rate]; and setting the LMD bits in the RLN3nLMD register to 11_B (LIN slave mode [fixed baud rate]), and setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B or 11_B sets LIN slave mode [fixed baud rate], turning the OMM1 and OMM0 bits in the RLN3nLMST register to 01_B or 11_B.

When changing a submode to another submode within LIN mode, a transition to LIN reset mode should first be made and change the LMD bits in the RLN3nLMD register.

The LIN mode provides the following two operation modes:

- LIN operation mode
- LIN wake-up mode

Figure 18.3 shows the transition of operation modes. **Table 18.77** describes the transition conditions of operation modes.

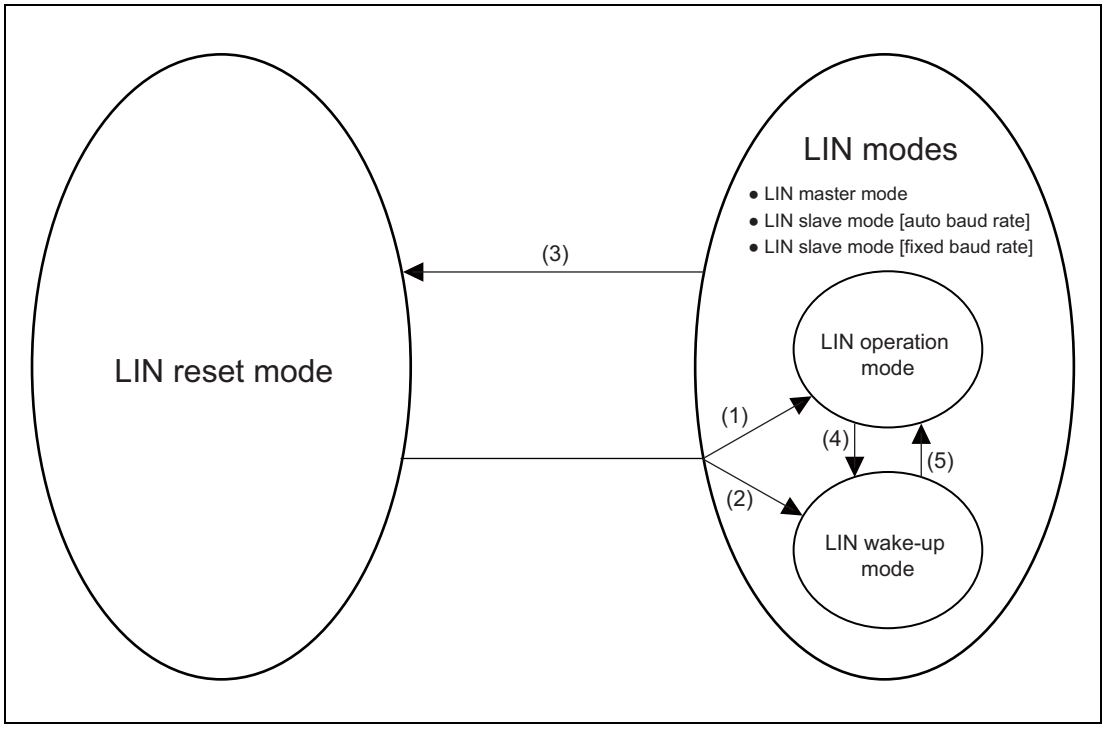


Figure 18.3 Transition of Operation Modes

Table 18.77 Transition Condition for Operation Mode

	Operation Mode Transition			Transition Condition
(1)	LIN reset mode	→	LIN mode • LIN operation mode	LMD bits in RLN3nLMD register = 00_B or 10_B or 11_B and OM1 and OM0 bits in RLN3nLCUC register = 11_B
(2)	LIN reset mode	→	LIN mode • LIN wake-up mode	LMD bits in RLN3nLMD register = 00_B or 10_B or 11_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B
(3)	LIN mode • LIN operation mode • LIN wake-up mode	→	LIN reset mode	OM0 bit in RLN3nLCUC register = 0_B
(4) *1	LIN mode • LIN operation mode	→	LIN mode • LIN wake-up mode	OM1 and OM0 bits in RLN3nLCUC register = 01_B
(5) *1	LIN mode • LIN wake-up mode	→	LIN mode • LIN operation mode	OM1 and OM0 bits in RLN3nLCUC register = 11_B

Note 1. Transition between LIN operation mode and LIN wake-up mode cannot be made when communication is in progress (when the FTS bit in the RLN3nLTRC register is 1).

(1) LIN Operation Mode

In LIN operation mode, frame processing (header transmission, header reception, response transmission, response reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 11_B changes the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 11_B. Communication settings should be performed after the OMM1 and OMM0 bits have become 11_B.

(2) LIN Wake-up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B changes the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 01_B. Communication settings should be performed after the OMM1 and OMM0 bits have become 01_B.

18.7.1 LIN Master Mode

18.7.1.1 Header Transmission

Figure 18.4 shows the operation of the LIN/UART interface (LIN master mode) in header transmission. **Table 18.78** describes the processing in header transmission.

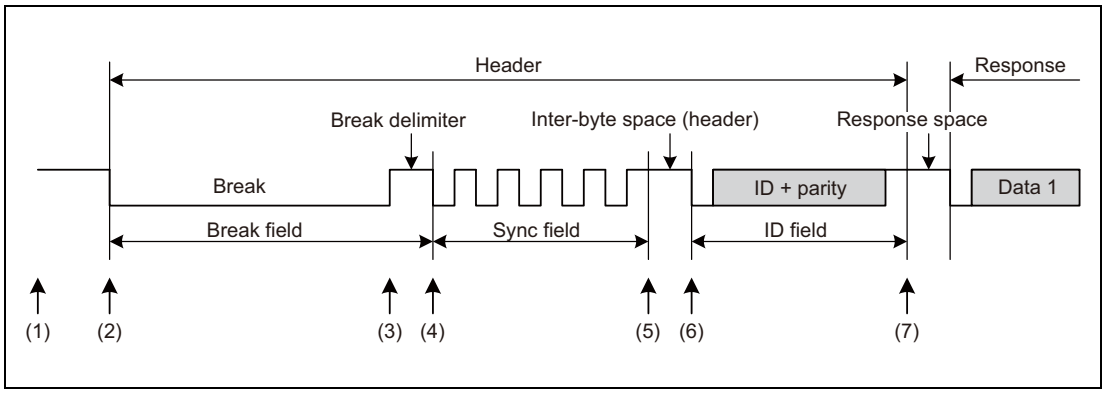


Figure 18.4 Operation in Header Transmission

Table 18.78 Processing in Header Transmission

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Changes the LIN/UART interface to the LIN master mode: LIN operation mode • Sets information on the frame to be transmitted (ID, parity, data length, response direction, checksum method, and transmission data)	Waits for the setting of the FTS bit in the RLN3nLTRC register by software (idle)
(2)	Sets the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started)	Transmits a break.
(3)	Waits for an interrupt request	Transmits a break delimiter.
(4)		Transmits a sync field (55_H).
(5)		Transmits an inter-byte space (header).
(6)		Transmits an ID field.
(7)		Sets a successful header transmission flag.

NOTE

For information about error detection conditions, see **Section 18.7.7, Error Status**.

18.7.1.2 Response Transmission

Figure 18.5 shows the operation of the LIN/UART interface (LIN master mode) in response transmission. **Table 18.79** describes the processing in response transmission.

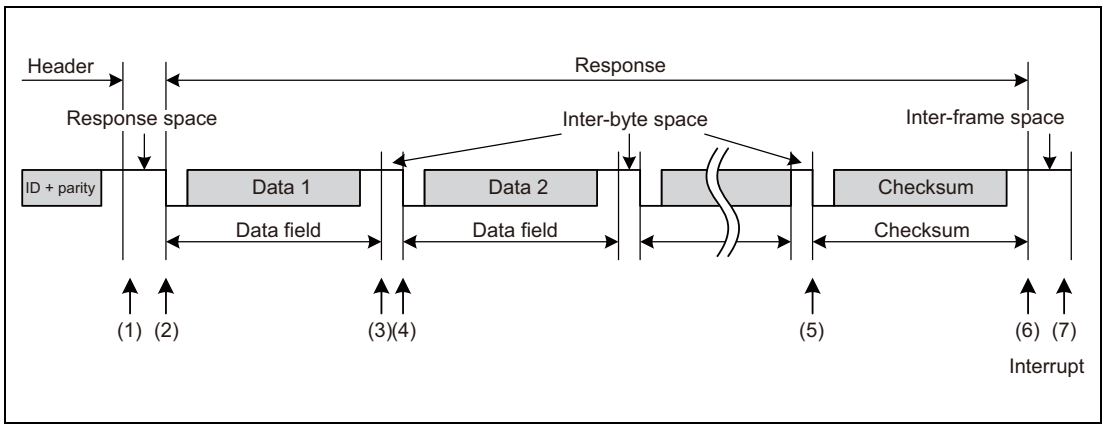


Figure 18.5 Operation in Response Transmission

Table 18.79 Processing in Response Transmission

	Software Processing	LIN/UART Interface Processing
(1)	[When in frame separate mode] • Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/ reception started) [When not in frame separate mode] • Waits for an interrupt request	[When in frame separate mode] • Waits for the setting of the RTS bit in the RLN3nLTRC register to 1 by software. • When the bit is set to 1, sends a response space. [When not in frame separate mode] • Sends a response space.
(2)	Waits for an interrupt request	Transmits data 1.
(3)		Transmits an inter-byte space.
(4)		• Transmits data 2. • Transmits an inter-byte space • Transmits data 3. • Transmits an inter-byte space (Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : :
(5)		Transmits the checksum.
(6)		• Sets a successful frame/wake-up transmission flag. • Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped) [When in frame separate mode] • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception is halted).
(7)	• Processing after communication Checks the RLN3nLST register, and clears flags.	Idle

NOTE

For information about error detection conditions, see **Section 18.7.7, Error Status**.

18.7.1.3 Response Reception

Figure 18.6 shows the operation of the LIN/UART interface (LIN master mode) in response reception. **Table 18.80** describes the processing in response reception.

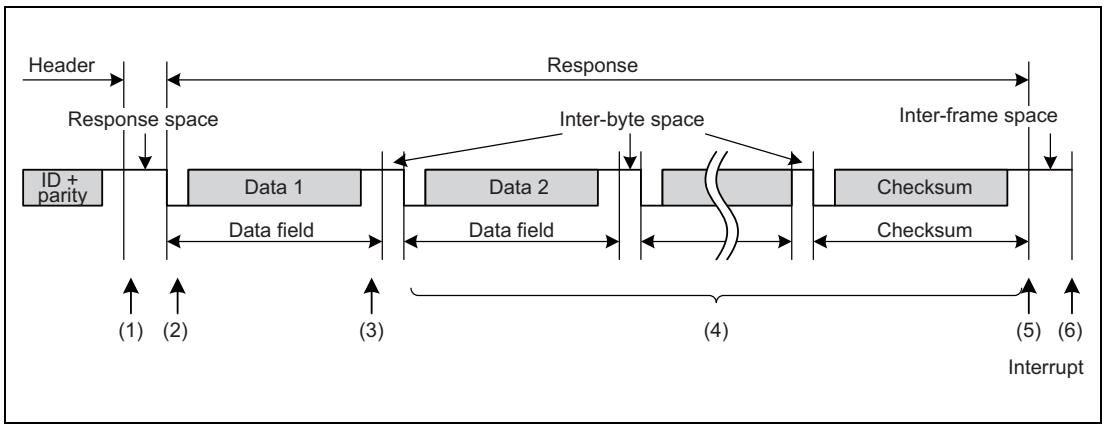


Figure 18.6 Operation in Response Reception

Table 18.80 Processing in Response Reception

	Software Processing	LIN/UART Interface Processing
(1)	Waits for an interrupt request (no processing)	Waits for detection of a start bit.
(2)		Receives data 1 when the start bit is detected.
(3)		Sets the successful data 1 reception flag.
(4)		• Receives data 2 when the start bit is detected. • Receives data 3 when the start bit is detected. Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : : • Receives the checksum when the start bit is detected.
(5)		• Determines the checksum. • Sets the successful frame/wake-up reception flag. • Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped).
(6)	• Processing after communication Reads the received data. Checks the RLN3nLST register, and clears flags.	Idle

NOTE

For information about error detection, see **Section 18.7.7, Error Status**.

18.7.2 LIN Slave Mode

18.7.2.1 Header Reception

Figure 18.7 shows the operation of the LIN/UART interface (LIN slave mode) in header reception.
Table 18.81 provides processing in header reception.

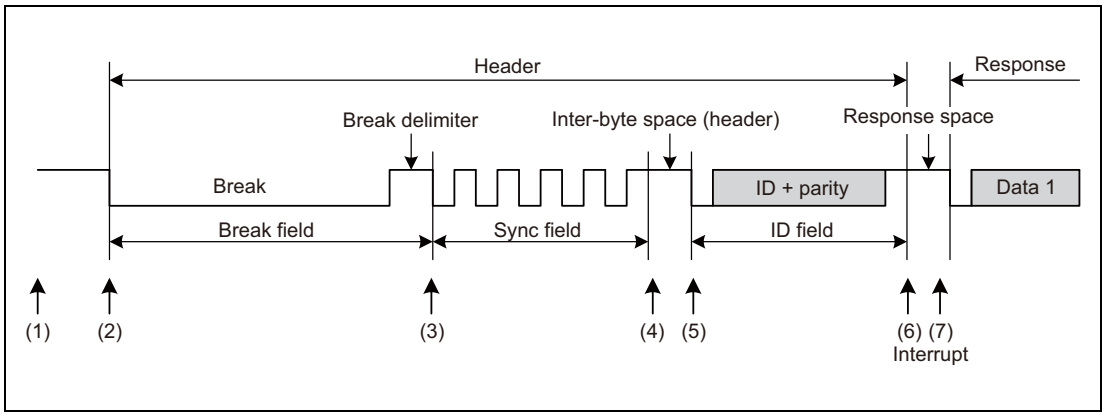


Figure 18.7 Operation in Header Reception

Table 18.81 Processing in Header Reception

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Changes the LIN/UART interface to the LIN slave mode: LIN operation mode • Sets the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/ reception started)	Waits for the setting of the FTS bit in the RLN3nLTRC register by software.
(2)	Waits for an interrupt request.	Waits for detection of break field
(3)		Detects a break field. (LIN slave mode [fixed baud rate]). For details about the break field detection timing in the case of LIN slave mode [auto baud rate], see [Auto Baud Rate Correction Function].)
(4)		• Detects a sync field (55_H) • Sets the baud rate generator (in the case of LIN slave mode [auto baud rate]) • Clears the no-response request bit (LNRR bit).
(5)		• Receives an ID field. • Checks an ID parity bit
(6)		Sets a header reception complete flag.
(7)	• Checks the RLN3nLST register, and clears flags. • Checks the RLN3nLIDB register, and prepares a response.	• Completes a header reception process. • Waits for a response request.

NOTE

The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the following Sync field and ID field) continues regardless of whether an error occurred. For information about error detection conditions, see **Section 18.7.7, Error Status**.

[Auto Baud Rate Correction Function]

In LIN slave mode [auto baud rate], the system always measures the low-level widths that are received. If the first “Low level” width is 10 times (if the BLT bit of the RLN3nLBFC register is “0”) or 11 times (if the BLT bit of the RLN3nLBFC register is “1”) or greater than the bit width calculated from the average of the starting 2 bits (the period of the consecutive falling edges from the beginning of the sync field) of the sync field, the system concludes that the detection of break field was successful, the system verifies that the data in the sync field is 55_H. If the data in the sync field is indeed 55_H and the system judges that sync field reception was successful, the system automatically sets the baud rate correction result to the RLN3nLBRP01 register.

If data is received up to the ID field without error, a successful header reception interrupt is generated at the stop bit position.

On the other hand, if the data in the sync field is not 55_H and the system judges that sync field reception failed, the system sets the sync field error flag and an error interrupt is generated. In that case, baud rate correction is not performed and the LIN/UART interface waits for the detection of the next break field (low level).

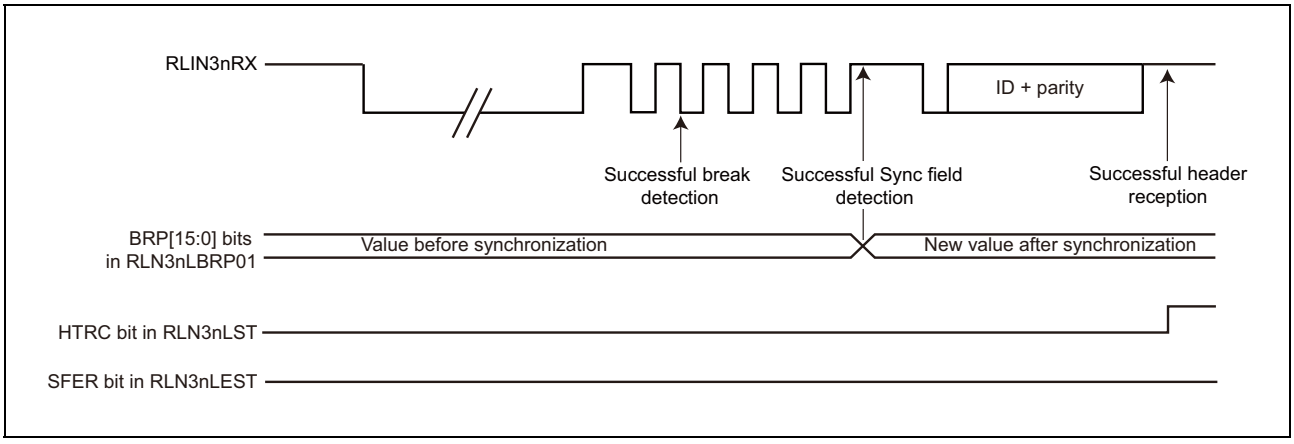


Figure 18.8 Header Reception in LIN Slave Mode [Auto Baud Rate] (in Normal Operation)

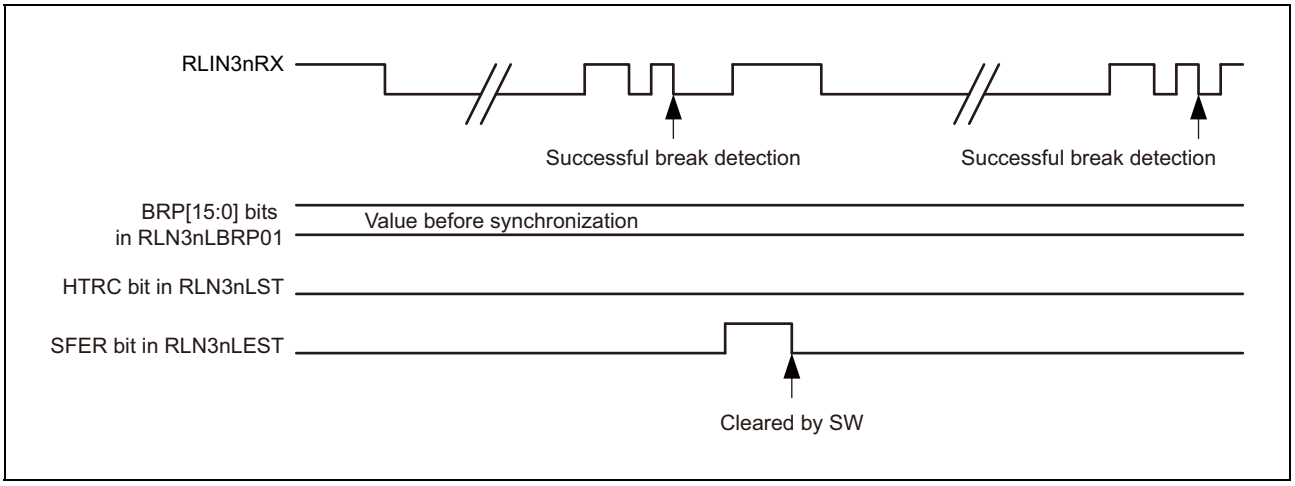


Figure 18.9 Header Reception in LIN Slave Mode [Auto Baud Rate] (Sync Field Error)

18.7.2.2 Response Transmission

Figure 18.10 shows the operation of the LIN/UART interface (LIN slave mode) in response transmission. **Table 18.82** describes the processing in response transmission.

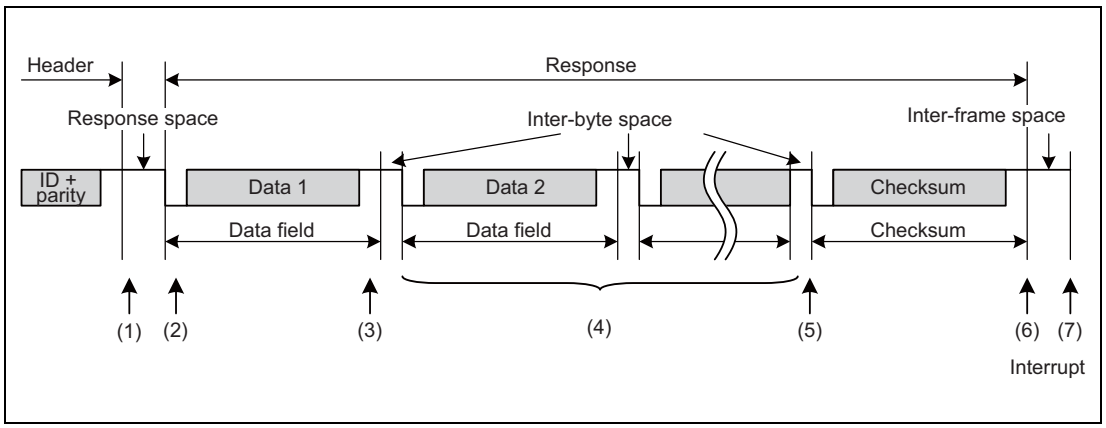


Figure 18.10 Operation in Response Transmission

Table 18.82 Processing in Response Transmission

	Software Processing	LIN/UART Interface Processing
(1)	• Sets the RLN3nLDFC register. • Sets the RLN3nLDBRb registers.(b = 1 to 8) • Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/ reception started)	• Waits for the setting of the RTS or LNRR bit of the RLN3nLTRC register by software • Transmits the response space after the RTS bit of the RLN3nLTRC register is set to 1
(2)	Waits for an interrupt request.	Transmits data 1.
(3)		Transmits the inter-byte space.
(4)		• Transmits data 2. • Transmits an inter-byte space • Transmits data 3. • Transmits an inter-byte space (Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : :
(5)		Transmits the checksum.
(6)		• Sets the successful response/wake-up transmission flag. • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).
(7)	• Processing after communication Checks the RLN3nLST register, and clears flags.	• Completes the response transmission process. • Waits for a new break.

NOTE

- For information about error detection, see **Section 18.7.7, Error Status**.
- The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the following Sync field and ID field) continues regardless of whether an error occurred.

18.7.2.3 Response Reception

Figure 18.11 shows the operation of the LIN/UART interface (LIN slave mode) in response reception. **Table 18.83** describes the processing in response reception.

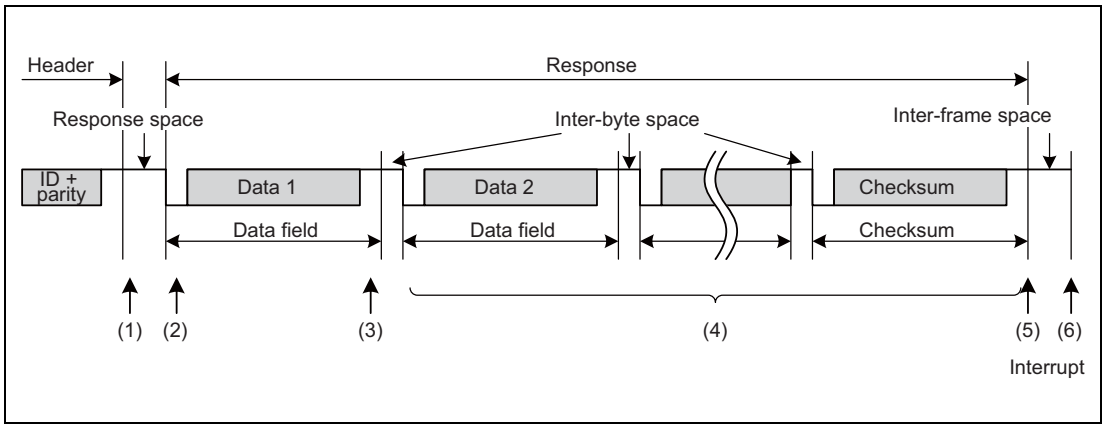


Figure 18.11 Operation in Response Reception

Table 18.83 Processing in Response Reception

	Software Processing	LIN/UART Interface Processing
(1)	• Sets the RLN3nLDFC register. • Sets the response transmission/ reception start bit (RTS bit) to 1.	• Waits for the setting by software of the response transmission/ reception start bit (RTS bit) or the no-response request bit (LNRR bit). • Waits for detection of the start bit.
(2)	Waits for an interrupt request.	Receives data 1 when the start bit is detected.
(3)		Sets the successful data 1 reception flag.
(4)		• Receives data 2 when the start bit is detected. • Receives data 3 when the start bit is detected. Repeats as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : : • Receives the checksum when the start bit is detected.
(5)		• Determines the checksum. • Sets a successful response/wake-up reception flag or error flag. • Sets the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).
(6)	• Processing after communication Reads the received data. Checks the RLN3nLST register, and clears flags.	• Completes the response process. • Waits for a new break.

NOTE

- For information about error detection conditions, see **Section 18.7.7, Error Status**.
- The LIN/UART interface allows reception of break fields during frame transmission/reception. In that case, a framing error, bit error or other error may be detected at the stop bit position of the frame before the break field is received, and a status interrupt may occur as a result. However, reception of a new header (the following Sync field and ID field) continues regardless of whether an error occurred.

18.7.2.4 No-response Request

Figure 18.12 shows the operation of the LIN/UART interface (LIN slave mode) when no response is requested. **Table 18.84** shows the processing that occurs when no response is requested.

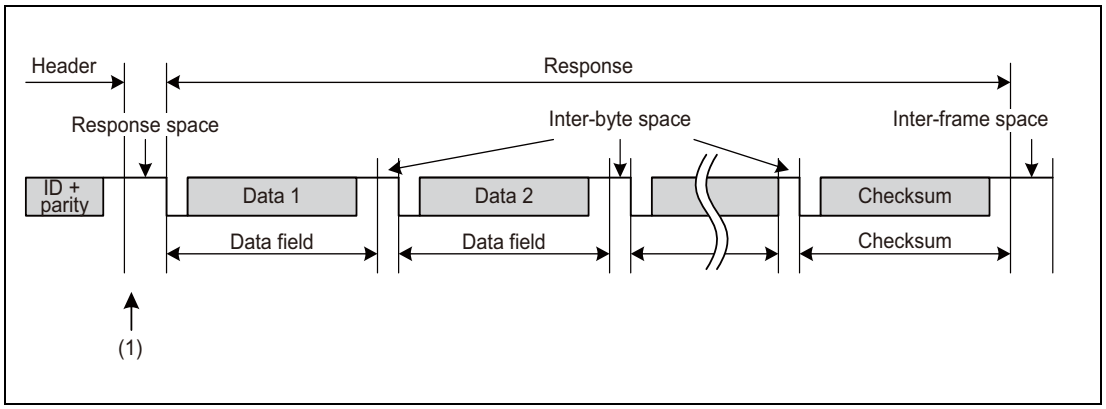


Figure 18.12 Operation when No Response is Requested

Table 18.84 Processing when No Response is Requested

Software Processing	LIN/UART Interface Processing
(1)	• Waits for setting of the no-response request bit (LNRR bit) by software
• Sets the no-response request bit (LNRR bit) to 1.	• Completes the frame reception process
	• Waits for a new break

18.7.3 Data Transmission/Reception

18.7.3.1 Data Transmission

One bit of data is transmitted per 1 Tbit.

The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data are compared bit by bit, and the results are stored in the BER flag in the RLN3nLEST register (see **Section 18.7.7, Error Status**).

In LIN master mode and LIN slave mode [fixed baud rate], 1 Tbit is generated to be 16fLIN, and thus the sampling point for received data is at the 13th clock cycle (81.25% position).

In LIN slave mode [auto baud rate], if 1 Tbit is generated to be 4fLIN, the sampling point for received data is at the third clock cycle (75% position). If 1 Tbit is generated to be 8fLIN, the sampling point for received data is at the 7th clock cycle (87.5% position).

Figure 18.13 shows an example of data transmission timing.

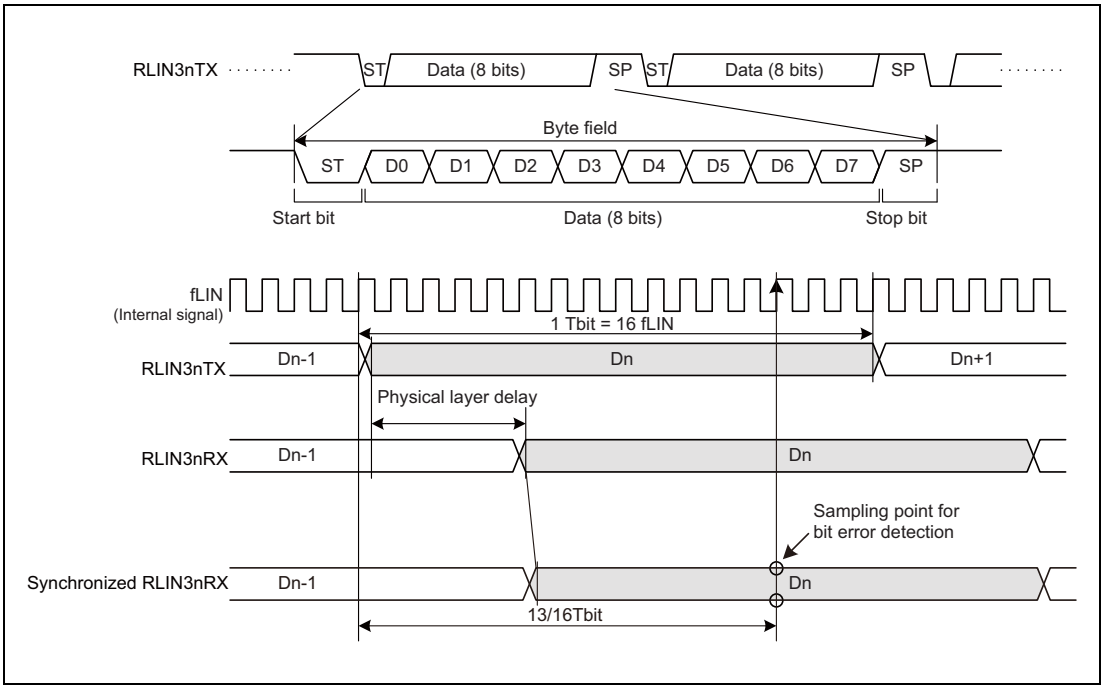


Figure 18.13 Example of Data Transmission Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])

18.7.3.2 Data Reception

Data reception is performed by using the synchronized RLIN3nRX signal (an internal signal) that is the input from the RLIN3nRX pin synchronized with prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized RLIN3nRX signal is low level. The falling edge is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted is fixed to low level or if a high level is detected on re-sampling.

After the start bit is detected, the system samples 1 bit per Tbit.

The LIN/UART interface has a noise filter function for reception data. If the LRDNFS bit in the RLN3nLMD register is 0, the LIN/UART interface uses a noise filter, and the value determined by a 3-sampling majority rule on prescaler clocks is used as the sampling value. If the LRDNFS bit in the RLN3nLMD register is 1, the LIN/UART interface does not use a noise filter, and the value of the synchronized RLIN3nRX value at the sampling position is used as the sampling value.

Figure 18.14 shows an example of data reception timing.

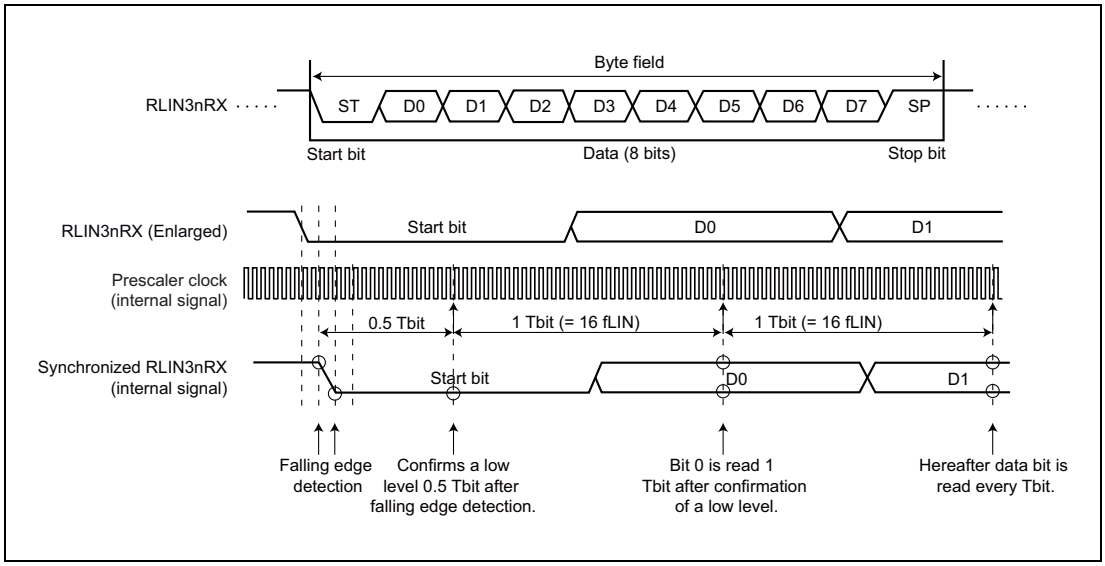


Figure 18.14 Example of Data Reception Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])

18.7.4 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN/UART interface sends or receives data continuously.

18.7.4.1 Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR8 are sequentially transmitted to data areas 1 to 8 of the LIN frame. In the case of a 4-bytes transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR4 are transmitted to data areas 1 to 4 of the LIN frame, but the contents of registers RLN3nLDBR5 to RLN3nLDBR8 are not transmitted. The transmitted checksum data is stored in the RLN3nLCBR register.

Figure 18.15 shows the LIN transmission processing and the required buffers.

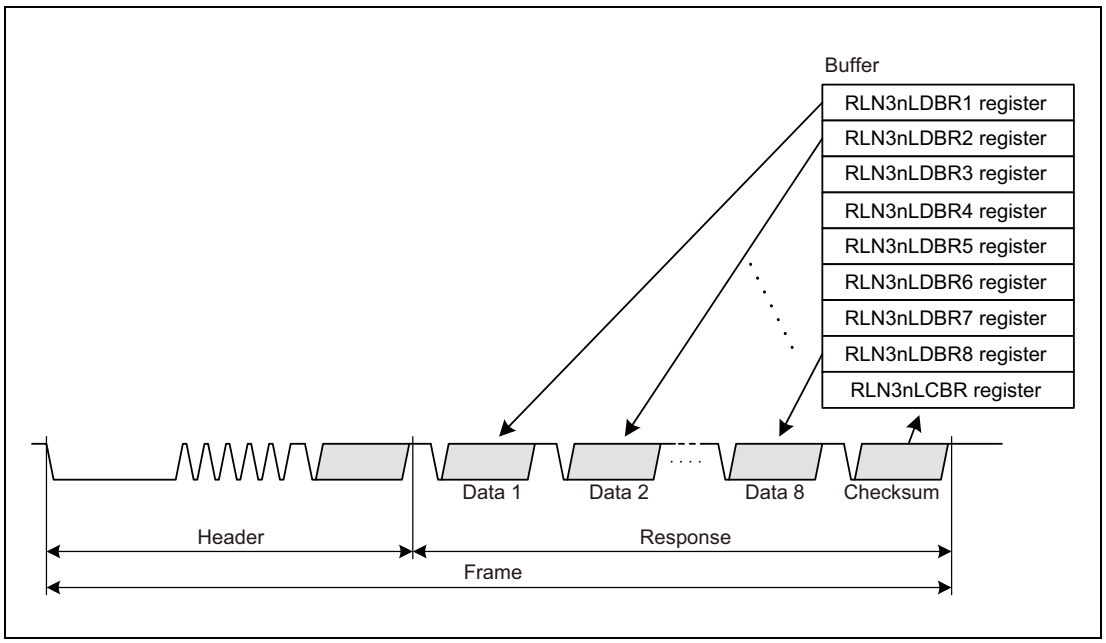


Figure 18.15 LIN Transmission Processing and Required Buffer

[Frame Separate Mode]

Setting the FSM bit in the RLN3nLDFC register to 1 turns on the frame separate mode.

In frame separate mode, a header and a response are transmitted when prompted by separate transmission start requests.

When the transmission of a header is finished, the HTRC flag in the RLN3nLST register is set to 1 (successful header transmission).

Use frame separate mode when transmitting or receiving response data of 9 bytes or more in LIN master mode.

18.7.4.2 Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR8, respectively, upon reception of a stop bit. In the case of a 4-byte reception, the contents of data areas 1 to 4 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR4, respectively; no data is stored in registers RLN3nLDBR5 to RLN3nLDBR8. Also, the received checksum data is stored in the RLN3nLCBR register.

Figure 18.16 shows the LIN reception processing and the required buffers.

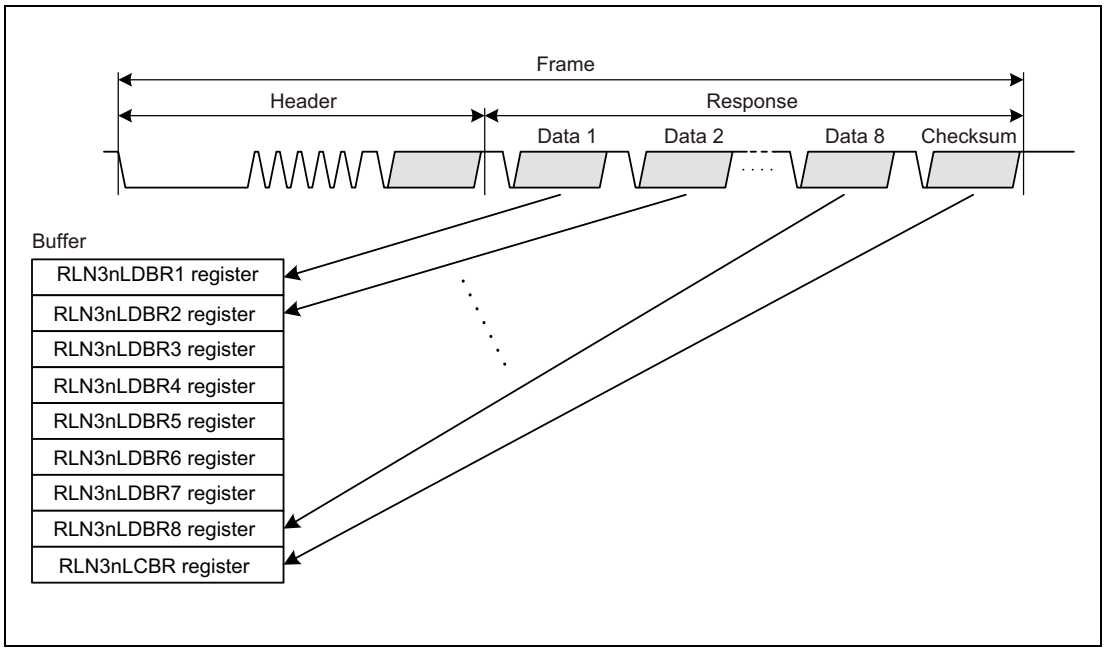


Figure 18.16 LIN Reception Processing and Required Buffer

[Reception of Data 1]

When the reception of the first byte of data is finished, the D1RC flag in the RLN3nLST register is set to 1 (successful data 1 reception).

18.7.4.3 Multi-Byte Response Transmission/Reception Function

Normally in LIN communications, a response is 9 bytes or less including a checksum field; however, responses of 10 bytes or more can also be transmitted and received.

In such a case, the bit error, framing error, response preparation error detection, and auto checksum functions are enabled.

If the data length is longer than 8 bytes, the LSS bit in RLN3nLDFC register should be set to 1 (indicating that the next data group to be transmitted or received is not the last data group) in the first data group (variable from 0 to 8 bytes) before transmitting or receiving the data group. After the transmission or reception, the user should determine whether the next data group is the last data group. If it is the last data group, the LSS bit in the RLN3nLDFC register should be set to 0 (indicating that the next data group to be transmitted or received is the last data group), and a checksum should be appended to the last data group.

By changing the RFDL bit in RLN3nLDFC register settings when the RTS bit in RLN3nLTRC register is 0, the user can change the data length for each data group.

When performing multi-byte response transmission/reception in LIN master mode, set the FSM bit in RLN3nLDFC register to 1 (frame separate mode).

NOTE

In LIN slave mode, the LIN/UART interface can detect a new break field during the transmission or reception of a response.

18.7.5 Wake-up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

18.7.5.1 Wake-up Transmission

In LIN wake-up mode, setting the RFT bit in the RLN3nLDFC register to 1 (LIN master mode: response transmission), or setting the RCDS bit in the RLN3nLDFC register to 1 (LIN slave mode: response transmission), and then the FTS bit in the RLN3nLTRC register to 1 (frame transmission, header reception or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low width of the wake-up signal should be set using the WUTL[3:0] bits in the RLN3nLWUP register.

However, if the LWBR0 bit of the RLN3nLWBR register in LIN master mode is 1 (LIN2.x), the LIN system clock (fLIN) becomes low level width at fa regardless of the setting of the LCKS bit of the RLN3nLMD register. By setting the WUTL[3:0] bits of the RLN3nLWUP register to 0100_B (5 Tbits), 260 µs low width can be output in LIN wake-up mode regardless of the setting of the LCKS bit of the RLN3nLMD register.

If a wake-up low-level width is output without any bit error, the FTC flag in the RLN3nLST register is set to 1 (successful frame response or wake-up transmission); when the FTCIE bit in the RLN3nLIE register is 1 (successful frame response/wakeup transmission interrupt enabled), an interrupt request for RLIN3n transmission is generated.

If RLN3nLEDE.BERE is set and a bit error is detected, wake-up transmission is canceled and the BER flag in the RLN3nLEST register is set to 1 (bit error detection).
When RLN3nLEDE.PBERE is set in LIN master mode, set RLN3nLEST.PBER flag to 1 (physical bus error detection) at the same time of a bit error.

Figure 18.17 shows the wake-up transmission timing.

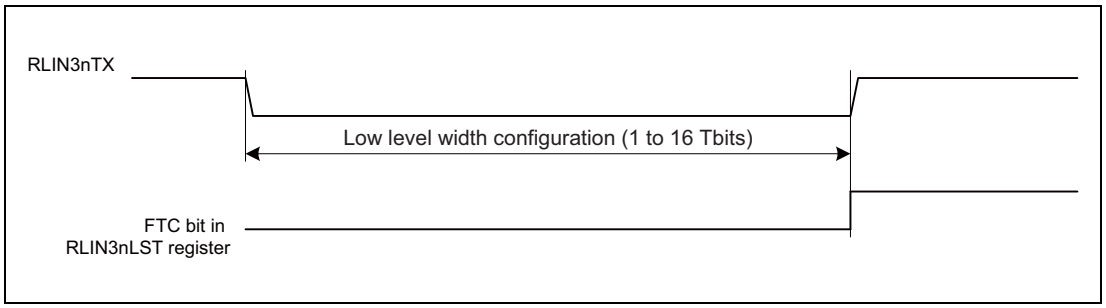


Figure 18.17 Wake-up Transmission Timing

18.7.5.2 Wake-up Reception

To detect a wake-up signal, use the input signal low-level width count function. The input signal low level width count function measures the low level width of the input signal to the RLIN3nRX pin, using the same sampling point as data reception. This allows the 2.5-Tbit or longer low-level width of the input signal of fLIN to be measured.

In LIN master mode, by setting the LWBR0 bit in the RLN3nLWBR register, operation can be executed without changing the baud rate generator setting when switching between LIN operation mode and LIN wake-up mode.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 sets the LIN system clock (fLIN) to fa regardless of the setting of the LCKS bit in the RLN3nLMD register. (The LCKS bit is not changed). By setting the baud rate to 19200bps while fa is selected, an input signal with a low-level width of 130 us or longer can be measured in LIN wake-up mode regardless of the setting of the LCKS bit in the RLN3nLMD register.

When using the wake-up reception function, in LIN wake-up mode set the RFT bit in the RLN3nLDFC register to 0 (LIN master mode: response reception), or the RCDS bit in the RLN3nLDFC register to 0 (LIN slave mode: response reception), and then set the FTS bit in the RLN3nLTRC register to 1 (frame transmission (header reception) or wake-up transmission/reception started).

When the low level width to be measured is reached, the FRC flag in the RLN3nLST register is set to 1 (successful frame response/wake-up reception). If the FRCIE bit in the RLN3nLIE register is 1 (successful frame response or wake-up reception interrupt enabled), an interrupt request for successful RLIN3n reception is generated.

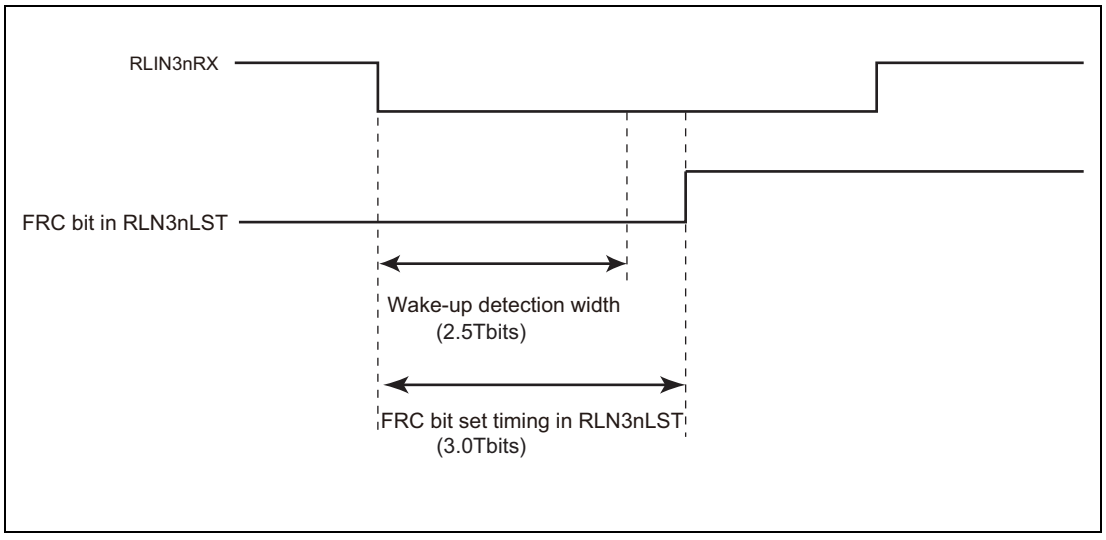


Figure 18.18 Input Signal Low level Count Function

18.7.5.3 Wakeup Collision

If the master node and the slave node transmit wakeup signals simultaneously, a collision will occur on the LIN bus, though a collision of wakeup signals is not detected by the LIN/UART interface.

18.7.6 Status

In LIN mode operation, the LIN/UART interface can detect seven types of statuses.

The four statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, and successful header transmission/reception, can generate interrupt requests.

Table 18.85 shows the types of statuses available in LIN master mode. **Table 18.86** lists the types of statuses available in LIN slave mode [auto baud rate] and in LIN slave mode [fixed baud rate].

Table 18.85 Types of Statuses in LIN Master Mode

Status	Status Set Condition	Status Clear Condition	Operation Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode.	All modes	OMM0 bit in RLN3nLMST register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if actually the LIN/UART interface enters LIN operation mode.	After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if actually the LIN/UART interface enters LIN wake-up mode.	• LIN operation mode • LIN wake-up mode	OMM1 bit in RLN3nLMST register	—
Frame/wake-up transmission end	When a frame (header transmission + response transmission), a wake-up signal, or a data group is transmitted successfully.	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FTC flag in RLN3nLST register	√
Frame/wake-up reception end	When a frame (header transmission + response reception), a wake-up signal, or a data group is received successfully.	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FRC flag in RLN3nLST register	√
Error detection	If any of the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software*1 • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	ERR flag in RLN3nLST register	√
Data 1 reception end	The RFT bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte of each data group is received successfully.*2	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	LIN operation mode	D1RC flag in RLN3nLST register	—
Header transmission end	When a header field is transmitted successfully.	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	LIN operation mode	HTRC flag in RLN3nLST register	√

Note 1. In LIN wake-up mode or LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, or BER flag in the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000_B (0-byte + checksum).

Table 18.86 Types of Statuses in LIN Slave Mode

Status	Status Set Condition	Status Clear Condition	Operation Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN–reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode.	After the OM0 bit of the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode.	All modes	OMM0 bit in RLN3nLMST register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if actually the LIN/UART interface enters LIN operation mode.	• After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if actually the LIN/UART interface enters LIN wake-up mode.	• LIN operation mode • LIN wake-up mode	OMM1 bit in RLN3nLMST register	—
Response/ wake-up transmission end	When a response field, a wake-up signal, or a data group is transmitted successfully.	• When cleared by software • Transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FTC flag in RLN3nLST register	√
Response/ wake-up reception end	When a response field, a wake-up signal, or a data group is received successfully.	• When cleared by software • Transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FRC flag in RLN3nLST register	√
Error detection	If any of the RPER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	• When cleared by software*1 • Transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	ERR flag in RLN3nLST register	√
Data 1 reception end	The RCDS bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte for each data group is received successfully.*2	• When cleared by software • Transition to LIN reset mode	LIN operation mode	D1RC flag in RLN3nLST register	—
Header reception end	When a header field is received successfully.	• When cleared by software • Transition to LIN reset mode	LIN operation mode	HTRC flag in RLN3nLST register	√

Note 1. In LIN wake-up mode or LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, IPER flag, CSER flag, SFER flag, FER flag, TER flag, or BER flag in the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000_B (0-byte + checksum).

18.7.7 Error Status

18.7.7.1 LIN Master Mode

(1) Types of Error Statuses

The LIN/UART interface can detect six types of error statuses in LIN master mode. The condition of these error statuses can be verified by checking the corresponding bits in the RLN3nLEST register.

All error statuses are interrupt factors.

Table 18.87 shows the types of error statuses.

Table 18.87 Types of Error Statuses in LIN Master Mode

Status	Error Detection Condition	Operation Mode Capable of Error Detection	Communication	Enable/ Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match *1*2	• LIN operation mode • LIN wake-up mode	Cancel	Enabled	BER flag in RLN3nLEST register
Physical bus error	• LIN bus detected a high level when sending a break • LIN bus detected a low level when sending a break delimiter • LIN bus detected a high level when sending a wake-up	• LIN operation mode • LIN wake-up mode	Cancel	Enabled	PBER flag in RLN3nLEST register
Timeout error	A frame or response transmission/reception is not completed within a given time*3	LIN operation mode	Cancel	Enabled	FTER flag in RLN3nLEST register
Framing error	In response field reception, a stop bit of each data byte is low level	LIN operation mode	Cancel	Enabled	FER flag in RLN3nLEST register
Checksum error	In response field reception, the result of checksum indicates an error	LIN operation mode	—	Disabled	CSER flag in RLN3nLEST register
Response preparation error	One of the following conditions occurs in frame separate mode during a multi-byte response reception: • After header transmission is complete, the first byte of receive data is received before a response transmission/reception request is set. • After the previous data group reception is complete, the first byte of receive data is received before a transmission/reception request for the next data group is set.	LIN operation mode	Cancel	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, processing is stopped after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is suspended immediately after the bit which had the error is sent. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is canceled after the error-causing bit is sent.

Note 2. In a multi-byte response transmission, bit errors are also detected between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLN3nLDFC register) and the checksum selection (the CSM bit in the RLN3nLDFC register), and can be calculated from the following formula.
When the FSM bit in the RLN3nLDFC register is 1 (frame separation mode), the timeout time is that for eight bytes until the RTS bit of the RLN3nLTRC register is set. Once the RTS bit is set, the timeout time is re-set to the time based on the response field data length (the RFDL[3:0] bits in the RLN3nLDFC register).

[Frame timeout]

When classic is selected (when the CSM bit in RLN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
When enhanced is selected (when the CSM bit in RLN3nLDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]
The aforementioned timeout time is a time longer than the TFRAME_MAX of LIN Specification Package Revision 1.3 when classic is selected, or the TFRAME_MAX of LIN Specification Package Revision 2.x when enhanced is selected.

[Response timeout]

Timeout time = (number of data bytes + 1) × 14 [Tbit]

When an error is detected, the timeout error detection function stops.

The error status is cleared when the next communication is started (when the FTS bit in the RLN3nLTRC register is set), by software, or at a transition to LIN reset mode.

(2) Target Time Area for LIN Error Detection

Figure 18.19 shows the time domain in which the LIN/UART interface in LIN master mode performs monitoring for error detection.

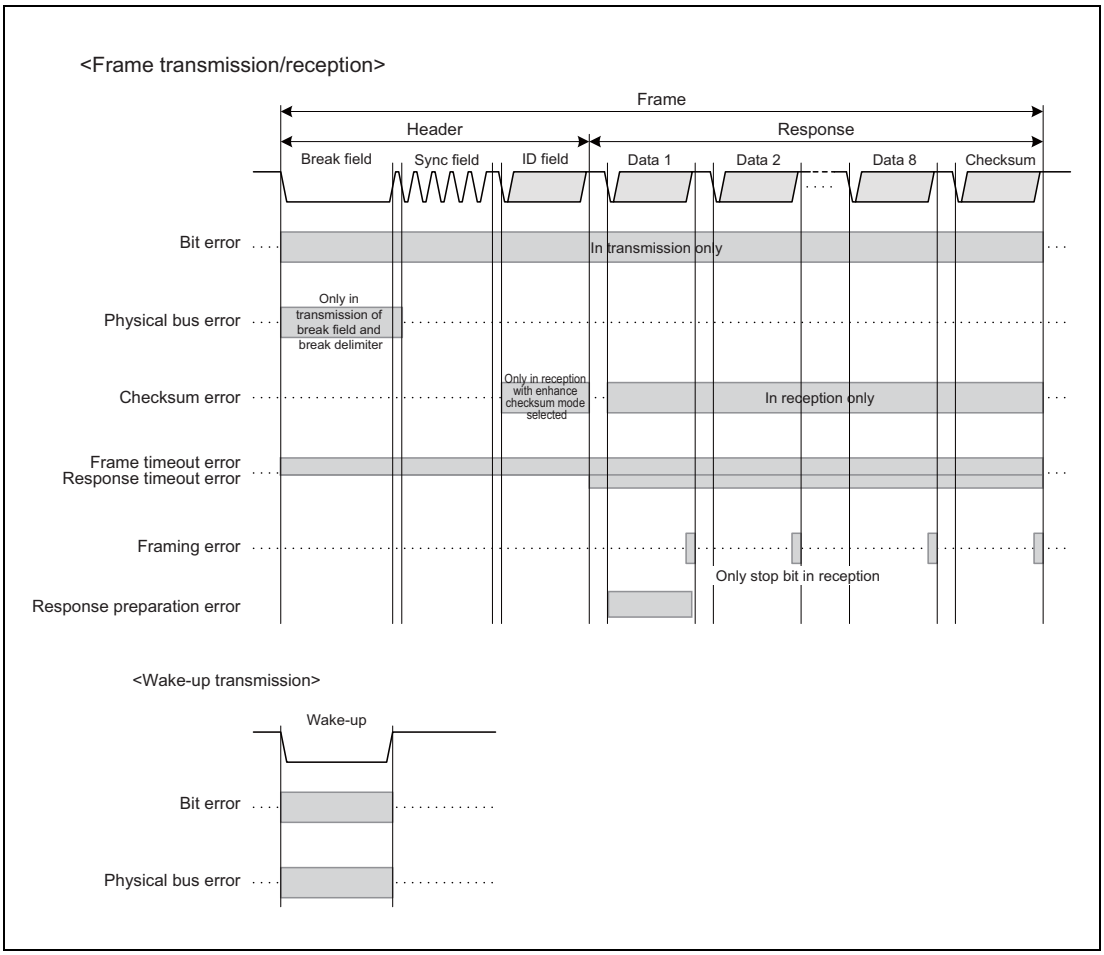


Figure 18.19 Target Time Area for LIN Error Detection (LIN Master Mode)

18.7.7.2 LIN Slave Mode

(1) Types of Error Statuses

The LIN/UART interface can detect seven types of error statuses in LIN slave mode [auto baud rate] or in LIN slave mode [fixed baud rate]. These error statuses can be verified by checking the corresponding bits in the RLN3nLEST register.

Table 18.88 shows the types of error statuses.

Table 18.88 Types of Error Statuses in LIN Slave Mode

Status	Error Detection Condition	Operation Mode Capable of Error Detection	Communication	Enable/ Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match*1 *2	• LIN operation mode • LIN wake-up mode	Cancel	Enabled	BER flag in RLN3nLEST register
Timeout error	A frame or response transmission/reception is not completed within a given time*3	LIN operation mode	Cancel	Enabled	TER flag in RLN3nLEST register
Framing error	In frame reception, a stop bit of each data byte is low level	LIN operation mode	Cancel	Enabled	FER flag in RLN3nLEST register
Sync field error	If the width of the break low level is greater than the width set by the LBLT bit in the RLN3nLBFC register and the sync field is not 55_H	LIN operation mode	Cancel	Enabled*4	SFER flag in RLN3nLEST register
Checksum error	In response field reception, the result of checksum indicates an error	LIN operation mode	—*5	Disabled	CSER flag in RLN3nLEST register
ID parity error	If the received ID parity bit does not match the value that is automatically calculated by the LIN/ UART interface	LIN operation mode	Cancel	Enabled	IPER flag in RLN3nLEST register
Response preparation error	• After the reception of a header, response preparation is not completed in time before the first byte of reception data is received. • In multi-byte response reception, the reception preparation for the next data group is not completed in time before the first byte of the next data group reception data is received.	LIN operation mode	Cancel	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, processing is stopped after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is suspended immediately after the bit which had the error is sent. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is canceled after the error-causing bit is sent.

Note 2. In a multi-byte response transmission, bit errors are also detected between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLN3nLDFC register) and the checksum selection (the LCS bit in the RLN3nLDFC register), and this can be calculated according to the following formula. The timeout period until the RTS or LNRR bit of the RLN3nLTRC register is set is 8 data bytes. When the RTS bit is set, the timeout time is reset to the time based on the response field data length (RFDL[3:0] bits of the RLN3nLDFC register). When the LNRR bit is set, the timeout function stops.

[Frame timeout]
When classic is selected (when the CSM bit in RLN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
When enhanced is selected (when the CSM bit in RLN3nLDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is a time longer than the TFRAME_MAX of LIN Specification Package Revision 1.3 when classic is selected, or the TFRAME_MAX of LIN Specification Package Revision 2.x when enhanced is selected.

[Response timeout]
Timeout time = (number of data bytes + 1) × 14 [Tbit]
When an error is detected, timeout error detection function stops.

Note 4. Only reflection of the result to the SFER flag can be enabled/disabled. Error detection cannot be enabled/disabled.

Note 5. Checksum judgment is performed upon completion of response frame reception. In case of an error, the successful reception flag is not set to 1.

The error status is cleared by software or at a transition to LIN reset mode.

(2) Target Time Area for LIN Error Detection

Figure 18.20 shows the time domain in which the LIN/UART interface in slave mode performs monitoring for error detection.

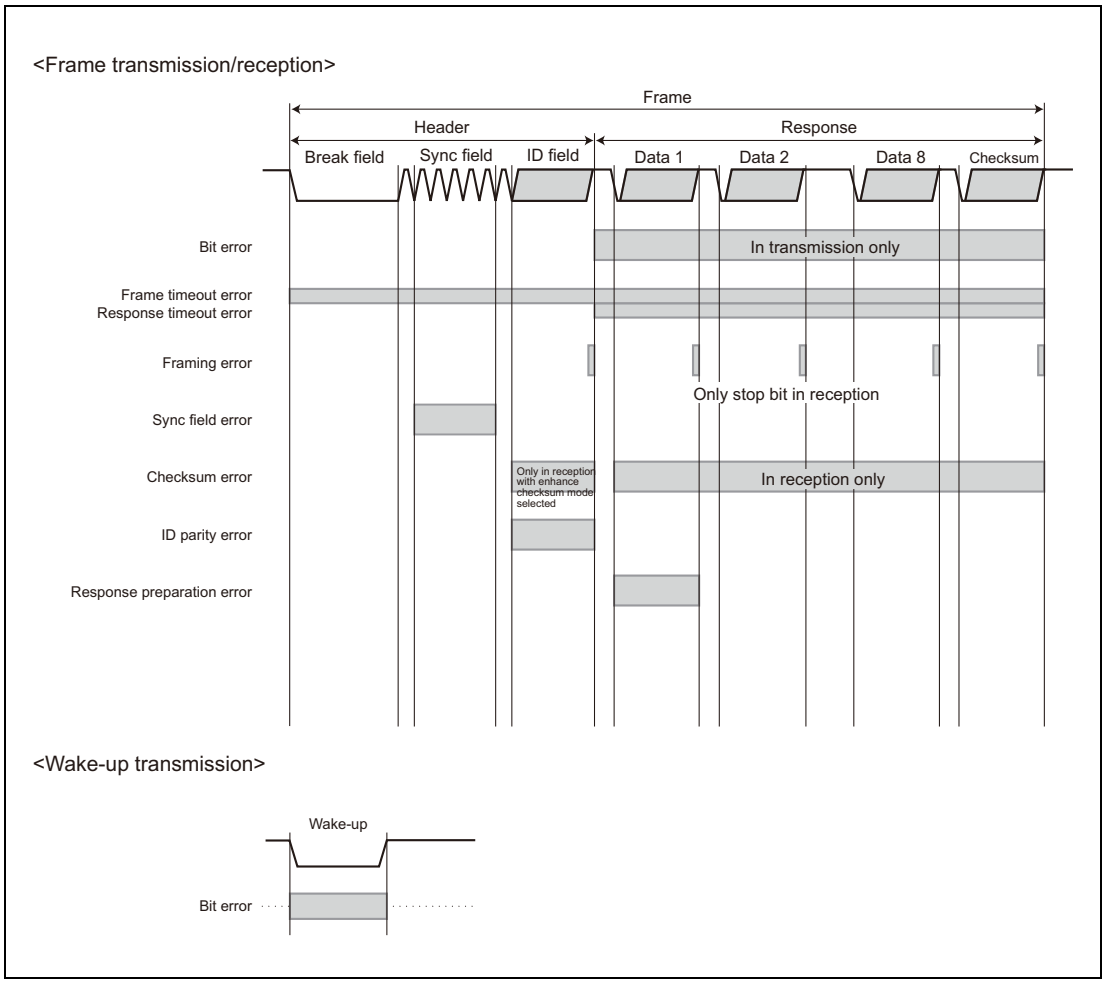


Figure 18.20 Target Time Area for LIN Error Detection (LIN Slave Mode)

18.8 UART Mode

In LIN reset mode, setting the LMD bits in the RLN3nLMD register to 01_B (UART mode) and the OM0 bit in the RLN3nLCUC register to 1 changes the mode to UART mode, turning the OMM0 bit in the RLN3nLMST register to 1.

18.8.1 Transmission

Figure 18.21 shows LIN/UART interface (in UART mode) transmission operations; **Table 18.89** shows LIN/UART interface (in UART mode) transmission processing.

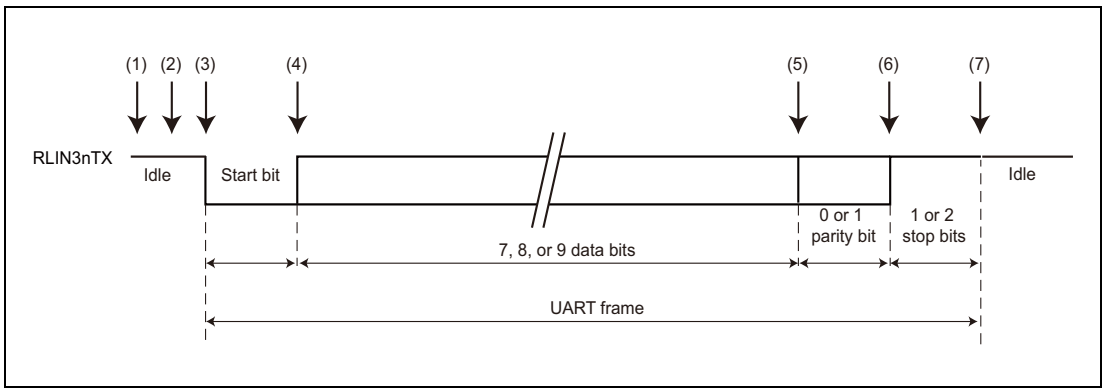


Figure 18.21 LIN/UART Interface (in UART Mode) Transmission Operation

Table 18.89 LIN/UART Interface (UART Mode) Transmission Processing (1/2)

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Enables error detection. • Sets data format. • Sets an interrupt generation timing. • Clears the LIN/UART interface from LIN reset mode. • Sets the transmit enable bit (UTOE bit) to 1.	• Waits for a transmission trigger (RLN3nLUTDR register) by software.
(2)	• Sets the transmit data to the UART transmit data register (RLN3nLUTDR) or UART wait transmit data register (RLN3nLUWTDR).	• Sets the transmit status flag.
(3)	• Waits an interrupt request.	• Transmits a start bit (for switching between transmission and reception in half duplex communication, transmits a start bit after receiving 1 stop bit. For details about this function, see **Section 18.8.1.4, Transmission Start Wait Function**.
	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)]	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)]
(4)	• When transmitting data continuously, sets another piece of transmission data in the UART transmit data register (RLN3nLUTDR register), waits for the generation of an interrupt request.	• Outputs a transmission interrupt. Transmits the data set in the UART (for wait) transmit data register.
(5)		Transmits a parity bit when parity is used.
(6)		Transmits 1 or 2 stop bits.

Table 18.89 LIN/UART Interface (UART Mode) Transmission Processing (2/2)

	Software Processing	LIN/UART Interface Processing
(7)	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • If another item of transmission data is set, goes to step (3).	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • If another piece of transmission data is set, goes to step (3). • If another piece of transmission data is not set, clears the transmit status flag.
	[When the UTIGTS bit is 1 (a transmission interrupt is generated upon end of transmission)] • When transmitting data continuously, goes to step (2).	[When the UTIGTS bit is 1 (a transmission interrupt is generated upon end of transmission)] • Generates RLIN3n transmission interrupt request. • Clears the transmission status flag.

18.8.1.1 Continuous Transmission

The LIN/UART interface (in UART mode) can transmit multiple sets of data continuously by using the RLN3nLUTDR register. **Figure 18.22** shows an operation example where the transmission interrupt generation timing is the start of transmission and an operation example where the transmission interrupt generation timing is the end of transmission.

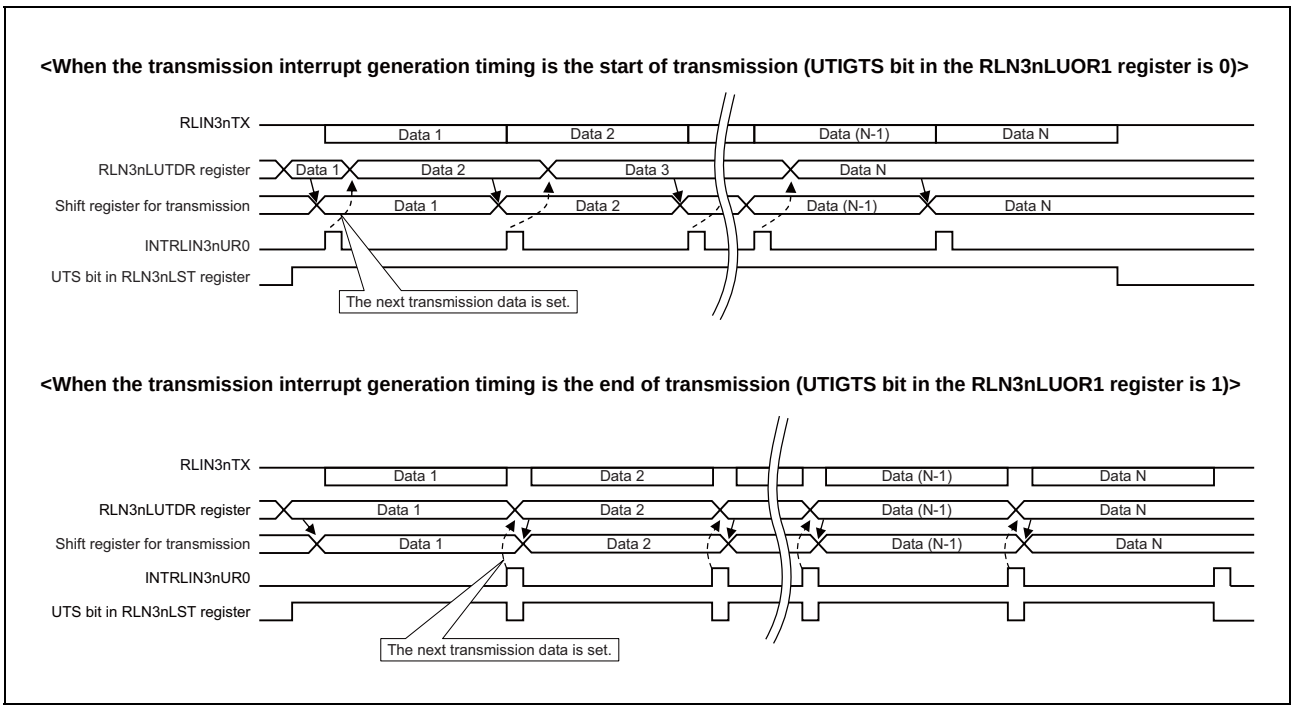


Figure 18.22 Operation Example of LIN/UART Interface (UART Mode) Continuous Transmission

An interrupt can be generated at the end of a transmission by changing the UTIGTS bit in the RLN3nLUOR1 register from 0 to 1 after the start of transmission of final data, provided that the transmission interrupt generation timing is the start of transmission and the end of transmission of final data needs to be known.

18.8.1.2 UART Buffer Transmission

The LIN/UART interface (in UART mode) has a maximum of nine bytes of UART buffers, and thus it is capable of performing continuous transmissions through the use of UART buffers.

Figure 18.23 shows the UART buffer transmission operation of the LIN/UART interface (in UART mode). **Table 18.90** describes the UART buffer transmission processing.

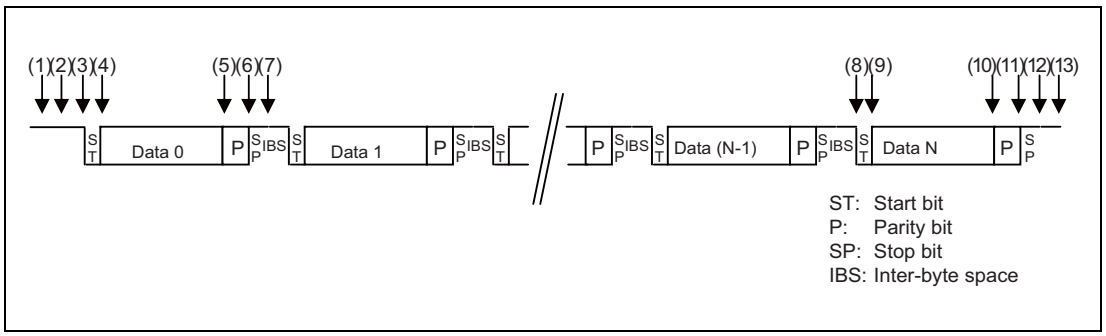


Figure 18.23 UART Buffer Transmission of LIN/UART Interface (in UART Mode)

Table 18.90 UART Buffer Transmission Processing of LIN/UART Interface (in UART Mode) (1/2)

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables error detection • Sets data format • Sets an interrupt generation timing to the end of transmission. • Clears the LIN/UART interface from LIN reset mode. • Sets the transmit enable bit (UTOE bit) to 1	• Waits for a transmission trigger (RTS bit) by software
(2)	• Sets the UART buffer data length and whether the system must wait for the start of transmission. • Sets the transmission data in the UART data 0 buffer register (RLN3nLUDB0) and the LIN data buffer b register (RLN3nLDBRb). (b =1 to 8) • Sets the UART buffer transmission start bit (RTS).	• Sets the transmit status flag.
(3)	Waits for an interrupt request.	Transmits a start bit. (When switching from reception to transmission during half-duplex communication, transmits the start bit upon completion of the stop bit for reception. For details about this function, see **Section 18.8.1.4, Transmission Start Wait Function**.)
(4)		Transmits the data set in the UART data buffer 0 register (RLN3nLUDB0) and the LIN/UART data buffer b register (RLN3nLDBRb).
(5)		Transmits a parity bit when parity is used.
(6)		Transmits 1 or 2 stop bits (When the number of data set in UART buffer data length select bits is 1, proceeds to (12).)
(7)		Transmits an inter-byte space (idle).
		Repeats steps (3) to (7) until number of data set in the UART buffer data length select bits -1 is reached.

Table 18.90 UART Buffer Transmission Processing of LIN/UART Interface (in UART Mode) (2/2)

	Software Processing	LIN/UART Interface Processing
(8)		Transmits a start bit.
(9)		Transmits the data set in the LIN/UART data buffer b register (RLN3nLDBRb).
(10)		Transmits a parity bit when parity is used.
(11)		Transmits 1 or 2 stop bits.
(12)		• Sets the successful buffer transmission flag. • Clears the UART buffer transmit start bit (RTS). • A transmission interrupt request signal. • Clears the transmission status flag.
(13)	• Checks the RLN3nLST register, and clears flags • In the case of continuous data transmission, goes to step (2).	

(1) UART Buffer Transmission

For a 9-byte transmission, the contents stored in the RLN3nLUDB0 and RLN3nLDBR1 to RLN3nLDBR8 registers are transmitted to data areas 0 to 8. The RLN3nLUDB0 register is used only if 9-byte transmission is set. In other cases, the RLN3nLDBR1 to RLN3nLDBR8 registers are selected depending upon the length of data involved. For a 4-byte transmission, the contents stored in the RLN3nLDBR1 to RLN3nLDBR4 registers are transmitted to data areas 1 to 4, but the contents of the RLN3nLDBR5 to RLN3nLDBR8 registers are not transmitted. An RLIN3n transmission interrupt is generated after the number of data specified in the MDL [3:0] bits of the RLN3nLDFC register is transmitted. The spaces between transmission data items can be set in the IBS bit in the RLN3nLSC register.

Figure 18.24 shows a 9-byte UART buffer and the transmission processing.

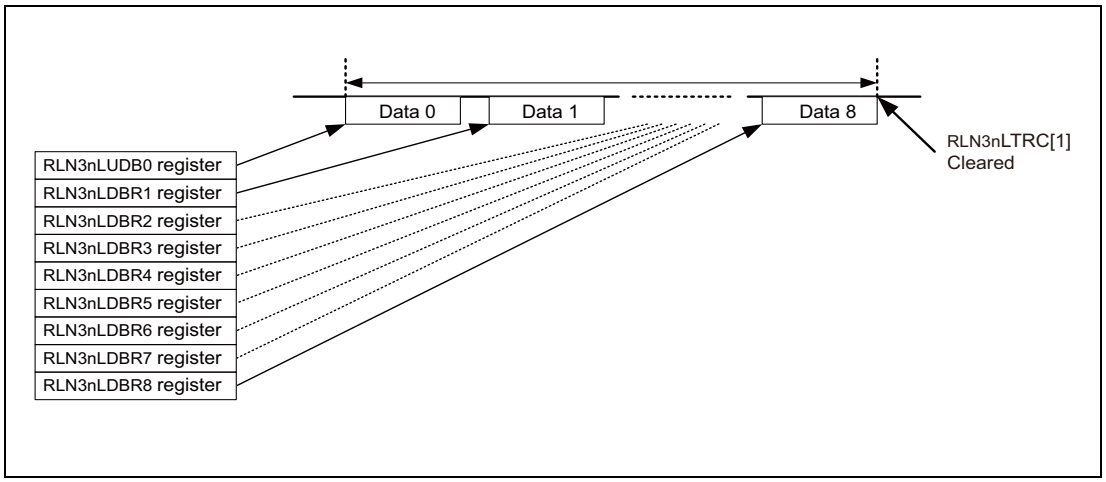


Figure 18.24 UART Buffer and Transmission Processing (for 9-Byte Transmission)

18.8.1.3 Data Transmission

One bit of data is transmitted per Tbit.

In half-duplex communication, if the BERE bit in the RLN3nLEDE register is 1 (bit error detection enabled), the transmission data and the input pin level are compared bit by bit during data transmission, and the results are stored in the BER flag in the RLN3nLEST register (see **Section 18.8.5, Error Status**). The timing at which the input pin is sampled during data transmission can vary depending upon the settings of the LPRS[2:0] and NSPB[3:0] bits in the RLN3nLWBR register.

The bit error detection timing in UART mode is shown in **Table 18.91**.

Table 18.91 Error Detection Timing in UART Mode

Sampling Count Per Bit	Bit Error Detection Timing
6 samples	3rd clock cycle + 1 prescaler clock
7 samples	4th clock cycle + 1 prescaler clock
8 samples	4th clock cycle + 1 prescaler clock
9 samples	5th clock cycle + 1 prescaler clock
10 samples	5th clock cycle + 1 prescaler clock
11 samples	6th clock cycle + 1 prescaler clock
12 samples	6th clock cycle + 1 prescaler clock
13 samples	7th clock cycle + 1 prescaler clock
14 samples	7th clock cycle + 1 prescaler clock
15 samples	8th clock cycle + 1 prescaler clock
16 samples	8th clock cycle + 1 prescaler clock

Example of Data Transmission Timing (when 1 Tbit = 16 samplings) is shown in **Figure 18.25**

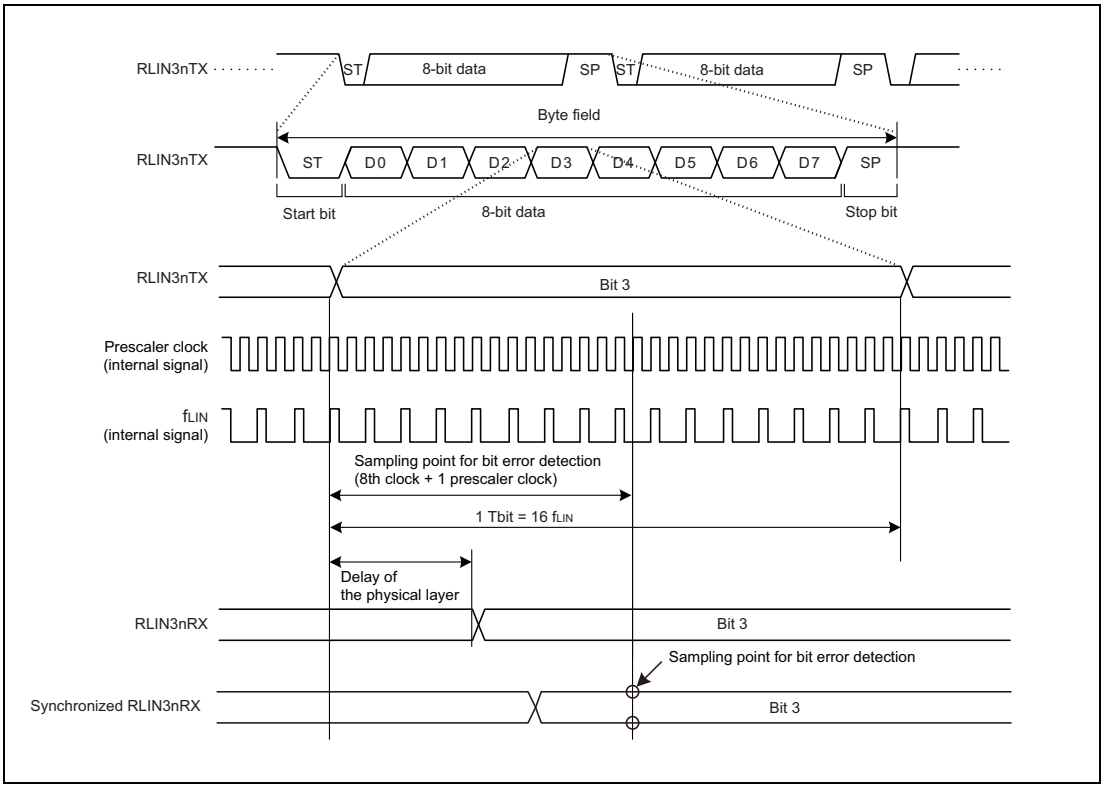


Figure 18.25 Example of Data Transmission Timing (When 1 Tbit = 16 samplings)

18.8.1.4 Transmission Start Wait Function

For performing half-duplex communication, the LIN/UART interface (in UART mode) has the function of securing the reception stop bit length when switching from reception to transmission.

If it is desired to delay the start of transmission until the stop bits for the reception are completed, set data in the RLN3nLUWTDR register, which is used only for the wait function, instead of setting transmission data in the RLN3nLUTDR register as a start-of-transmission request. When transmitting from the UART buffer, set 1 (UART buffer transmission started) in the RTS bit in the RLN3nLTRC register with 1 set in the UTSW bit in the RLN3nLDFC register.

In such a case, the LIN/UART interface delays the start of transmission until the stop bits of reception data are completed.

Note that even if the UART stop bit length selection bit (USBLS) in RLN3nLBFC register is 1 (stop bits = 2 bits), there is only a 1-bit delay.

Figure 18.26 shows the operation of transmission wait function.

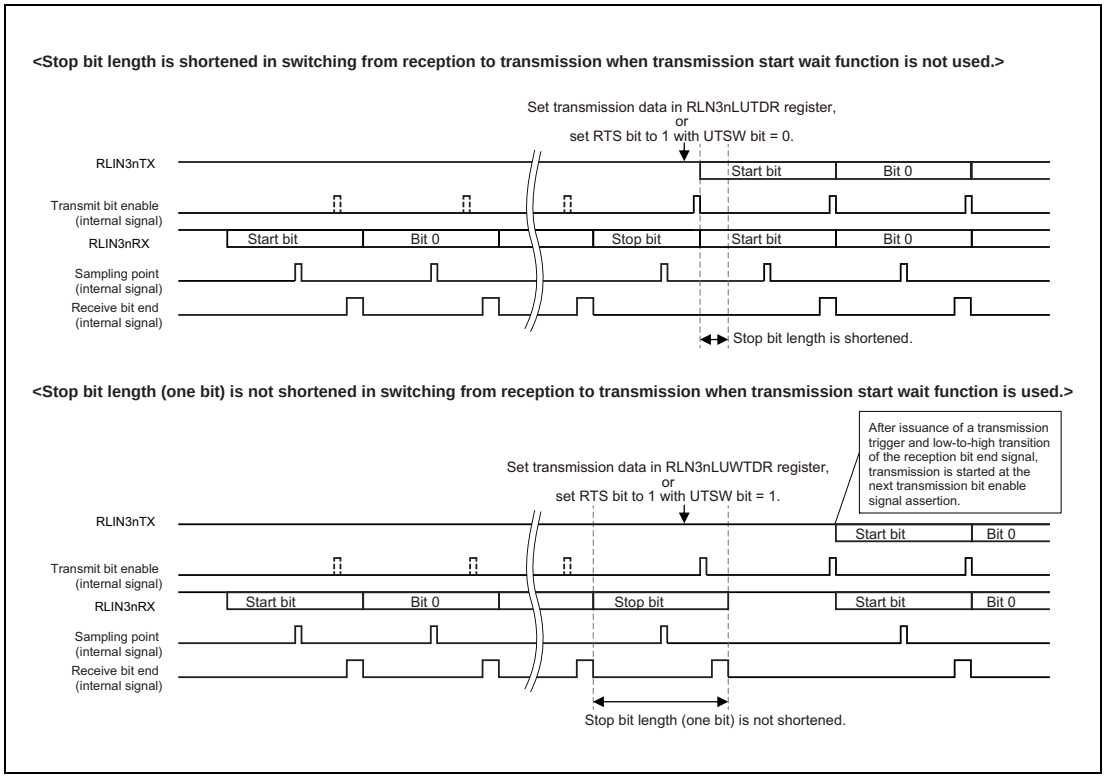


Figure 18.26 Case When Transmit Data Is Set While Stop Bits Are Being Received

18.8.2 Reception

Figure 18.27 shows the LIN/UART interface (in UART mode) reception operation.**Table 18.92** shows the LIN/UART interface (in UART mode) reception processing.

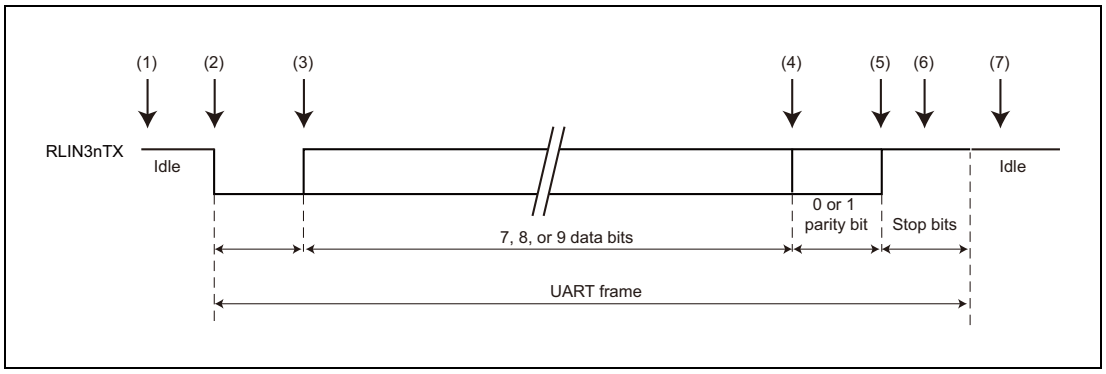


Figure 18.27 LIN/UART Interface (in UART Mode) Reception Operation

Table 18.92 LIN/UART Interface (in UART Mode) Reception Processing

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Enables error detection. • Sets data format. • Clears the LIN/UART interface from LIN reset mode. • Sets the receive enable bit (UROE bit) to 1.	• Waits for the reception to be enabled by software. • Waits for detection of a start bit.
(2)	Waits for an interrupt request.	• Waits for a falling edge from the reception pin, and detects a start bit. • Sets the reception status flag.
(3)		Receives data.
(4)		Receives a parity bit when parity is used.
(5)		Receives only 1 stop bit.
(6)		• Generates a successful RLIN3n reception interrupt request. • Clears the reception status flag.
(7)	Checks the RLN3nLST register, and clears flags	Waits for a falling edge from the reception pin.

18.8.2.1 Data Reception

Data reception is performed by using the synchronized RLIN3nRX (an internal signal) that is the input from the RLIN3nRX pin synchronized with the prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, resampling is performed 0.5 Tbits later when the number of sampling per 1 Tbit is even and {(the number of sampling + 1) / 2} / (the number of sampling) Tbits later when the number is odd. If the synchronized RLIN3nRX signal is low level, the bit is recognized as a start bit. The bit is not recognized as a start bit if the RLIN3nRX signal after the reset is de-asserted is fixed to low level or if a high level is detected during the resampling.

After the start bit is detected, 1 bit is sampled per Tbit.

However, when the BERE bit in the RLN3nLEDE register is 1, the sampling point is the same as the bit error detection timing.

The LIN/UART interface has a noise filtering function for received data. If the LRDNFS bit in the RLN3nLMD register is 0, the noise filter is used. For a sampling value, the value determined by a 3-samplings majority rule by the prescaler clock is used. If the LRDNFS bit in the RLN3nLMD register is 1, the noise filter is not used. In this case, for a sampling value, the synchronized RLIN3nRX value at the sampling position is used as is.

Figure 18.28 shows an example of data reception timing.

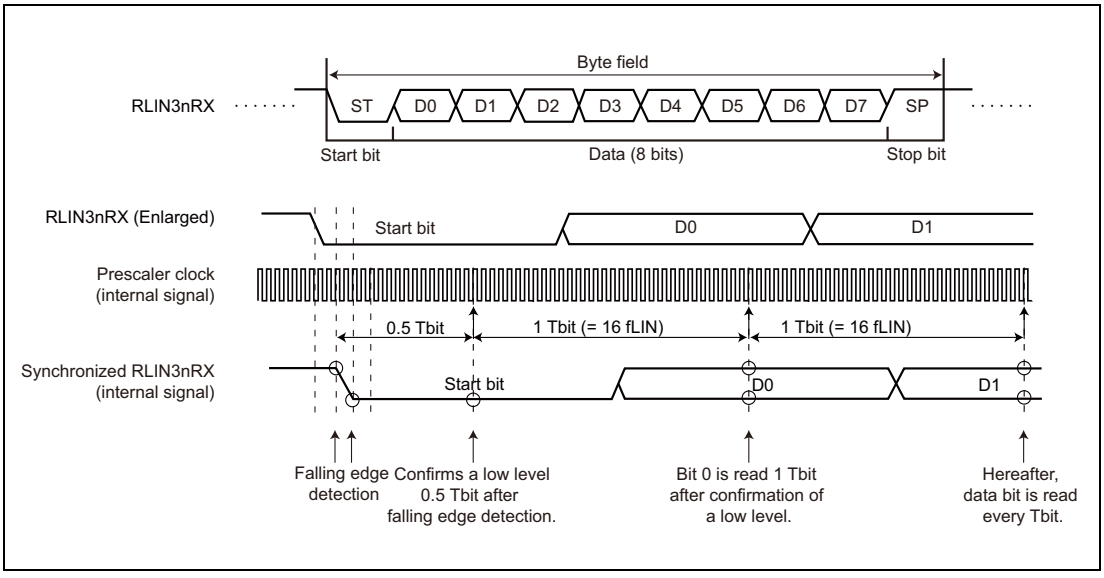


Figure 18.28 Example of Data Reception Timing (When Sampling Count is 16 in 1 Tbit)

18.8.3 Expansion Bits

The LIN/UART interface (in UART mode) can transmit and receive 9-bit long data by setting the UEBE bit in the RLN3nLUOR1 register to 1.

18.8.3.1 Expansion Bit Transmission

The LIN/UART interface (in UART mode) can transmit 9-bit long data when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1 and by writing the 9-bit data to either the UART transmission data register (RLN3nLUTDR) or the UART wait transmission data register (RLN3nLUWTDR).

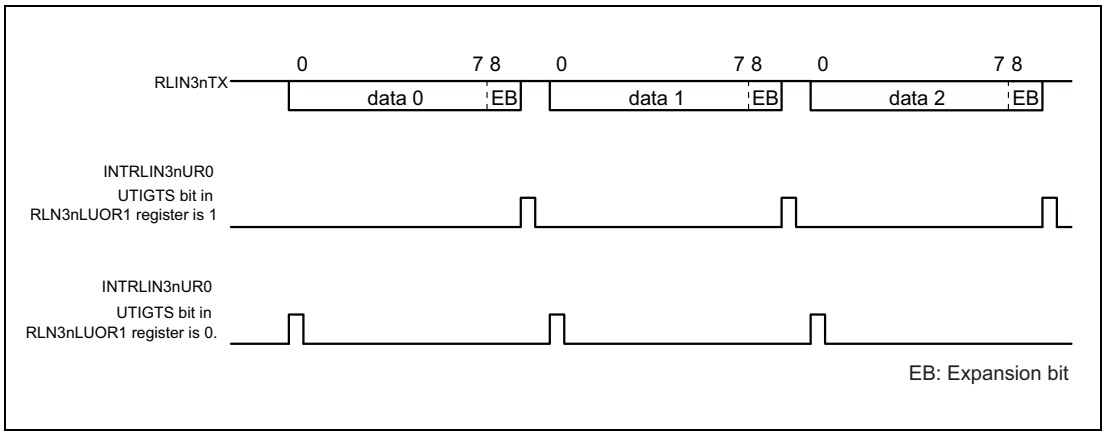


Figure 18.29 Transmission Example When Expansion Bit is Enabled (LSB First)

18.8.3.2 Expansion Bit Reception

With the LIN/UART interface (in UART mode), 9-bit data can always be received without requiring a comparison of expansion bits, provided that the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 1, and the expansion bit data comparison enable bit (UEBDCE) is 0. Irrespective of the particular setting of the expansion bit detection level selection bit (UEBDL) in the UART option register 1 (RLN3nLUOR1), a successful RLIN3n reception interrupt is generated when 9-bit data is received.

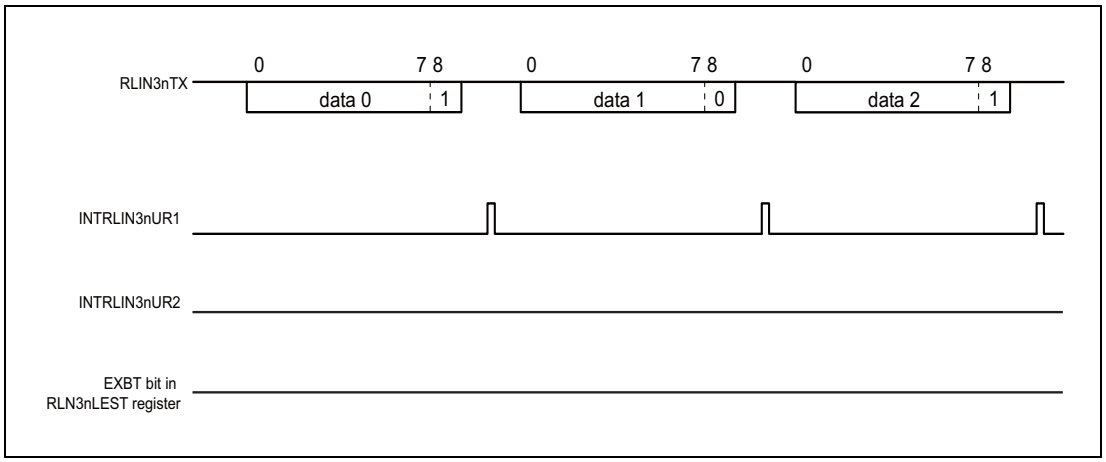


Figure 18.30 Expansion Bit Reception Example (LSB First)

18.8.3.3 Expansion Bit Reception (with Expansion Bit Comparison)

The LIN/UART interface (in UART mode) can compare received expansion bits and the UEBDL bits when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 0.

If the level that was set in the expansion bit detection level selection bit (UEBDL) is detected, an RLIN3n status interrupt request is generated upon completion of data reception, and the expansion bit detection flag (EXBT) in the LIN error status register (RLN3nLEST) is set. If the reversed value of an expansion bit detection level is detected, successful RLIN3n reception interrupt request is generated. In either case, the received data is stored in the UART reception data register (RLN3nLURDR), unless there was an overrun error.

Figure 18.31 shows an example when the expansion bit detection level selection bit (UEBDL) is set to 0.

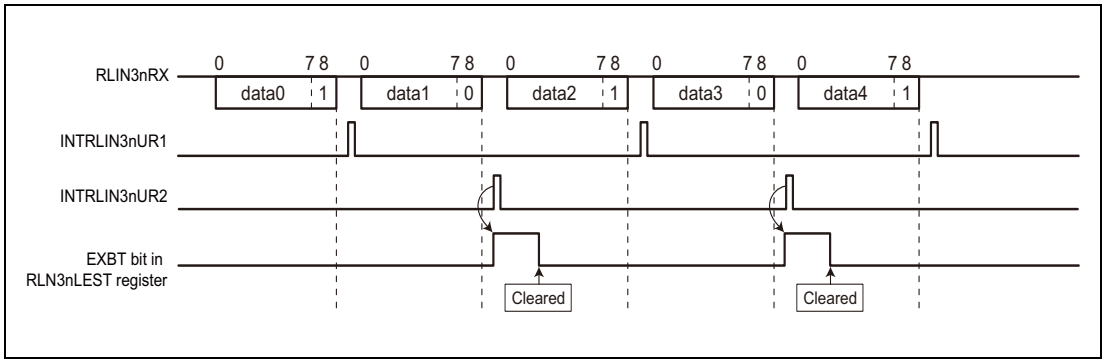


Figure 18.31 Expansion Bit Reception Example (with Expansion Bit Comparison) (LSB First, UEBDL = 0)

NOTE

- If a reception error (parity error, framing error, or overrun error) occurs in received data 0, 2, or 4 (if a reversed value of an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. In this case, a successful RLIN3n reception interrupt is not generated.
- If a reception error (parity error, framing error, or overrun error) occurs in received data 1 or 3 (if an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. If the overrun error occurs, the expansion bit detection flag (EXBT) is also set.

18.8.3.4 Expansion Bit Reception (with Data Comparison)

If the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 1, and if the level that was set by the expansion bit detection level selection bit (UEBDL) is detected, the LIN/UART interface (in UART mode) compares the 8 bits, excluding the expansion bit in the received data, with the a pre-set RLN3nLIDB register value.

If the result of the comparison is a match, the LIN/UART interface performs the following operations:

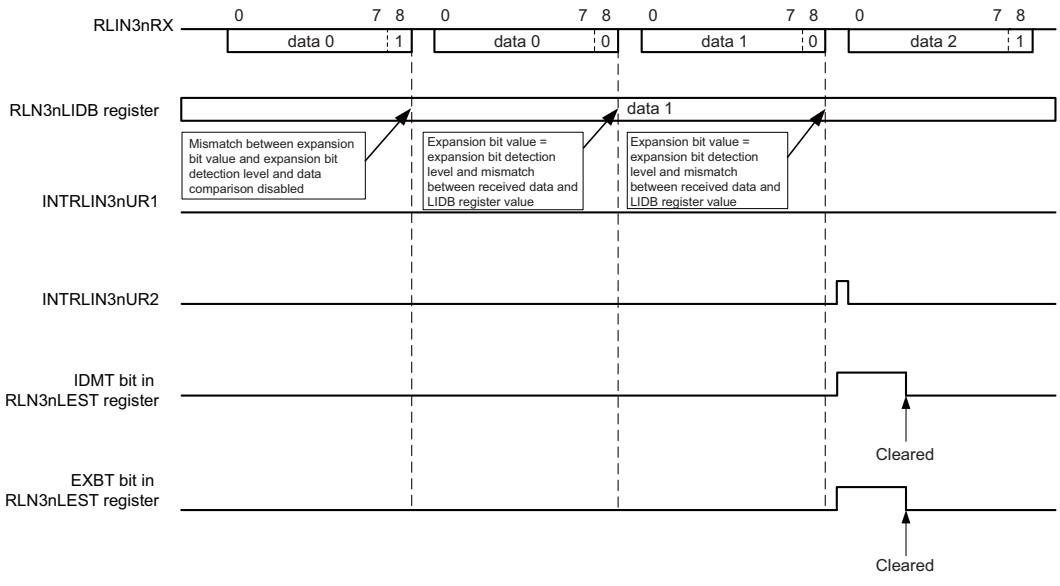
- Generates an RLIN3n status interrupt
- Sets an expansion bit detection flag (EXBT)
- Sets an ID match flag (IDMT)
- Stores the received data in the UART reception data register (RLN3nLURDR)

Even when the result of the comparison is a match, successful RLIN3n reception interrupt is not generated.

If the result of the comparison is not a match, no successful RLIN3n reception interrupt or RLIN3n status interrupt is generated, and the EXBT and IDMT flags are not set to 1. The received data is not stored in the UART reception data register (RLN3nLURDR).

When changing the UEBDCE bit to 0, make the change before the reception of another set of data is finished.

Figure 18.32 shows an example when the expansion bit detection level selection bit (UEBDL) is set to 0.



Note: If a reception error (parity error, framing error, or overrun error) occurs, an RLIN3n status interrupt is generated, and the error flag is updated. If the overrun error occurs and the result of comparison is a match, the EXBT and IDMT are also set to 1.

Figure 18.32 Expansion Bit Reception Example (with Data Comparison) (LSB First, UEBDL = 0)

18.8.4 Status

In UART mode, the LIN/UART interface can detect five types of statuses.

Two statuses, successful UART buffer transmission and error detection, can generate interrupt requests.

Table 18.93 shows the types of statuses available in UART mode.

Table 18.93 Types of Statuses in UART Mode

Status	Status Set Condition	Status Clear Condition	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN-reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode.	OMM0 bit in RLN3nLMST register	—
Successful UART buffer transmission	• When the UTIGTS bit in the RLN3nLUOR1 register is 0 (transmission interrupt request is generated upon start of transmission), the transmission of the last data of the data length set by the MDL bit in the RLN3nLDFC register is started. • When the UTIGTS bit in the RLN3nLUOR1 register is 1 (transmission interrupt request is generated upon end of transmission), the transmission of the data length set by the MDL bit in the RLN3nLDFC register is ended.	• When cleared by software • After transition to LIN reset mode	FTC flag in RLN3nLST register	√
Error detection	If any of the UPER flag, IDMT flag, EXBT flag, FER flag, OER flag, and BER flag in the RLN3nLEST register is set to 1 (error detected).	• When cleared by software*1 • After transition to LIN reset mode	ERR flag in RLN3nLST register	√
Transmission status	• When data is written to the RLN3nLUTDR or RLN3nLUWTDR register. • When 1 is written to the RTS bit in the RLN3nLTRC register.	• When the transmission of the data specified in the RLN3nLUTDR or RLN3nLUWTDR register is completed, but the next transmission data is not specified. • When the transmission of the data in the UART buffer is completed and the RTS bit in the RLN3nLTRC register is cleared. • After transition to LIN reset mode	UTS flag in RLN3nLST register	—
Reception status	• When a start bit is detected.	• When a sampling point for stop bits is detected • After transition to LIN reset mode	URS flag in RLN3nLST register	—

Note 1. Writing a 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register when the LIN reset mode is being canceled sets the ERR flag in the RLN3nLST register to 0.

18.8.5 Error Status

Types of Error Statuses

In UART mode, the LIN/UART interface can detect four types of errors and two types of statuses. The condition of these statuses can be verified by using the corresponding bits in the RLN3nLEST register.

Table 18.94 shows available status types.

Table 18.94 Types of Statuses in UART Mode

Status	Error Detection Condition	Communication	Enable/ Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data monitored on the receive pin do not match*1	Continues until the transmission of the set transmission data is finished.	Enabled	BER flag in RLN3nLEST register
Overrun error	After received data is stored in the RLN3nLURDR register, another data item is received before the data is read. (In this case, no data is stored in the RLN3nLURDR register).	— (Reception is finished by the time this error is detected)	Enabled	OER flag in RLN3nLEST register
Framing error	When the first stop bit is low level in the reception processing.	— (Reception is finished by the time this error is detected)	Enabled	FER flag in RLN3nLEST register
Parity error	The received parity value fails to match the parity value calculated from the received data	Continues until the data reception is finished.	Disabled *2	UPER flag in RLN3nLEST register
Expansion bit detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register.	—	Enabled	EXBT flag in RLN3nLEST register
ID match detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register and the 8-bit receive data excluding the expansion bit matches the value of the RLN3nLIDB register.	—	Enabled	IDMT flag in RLN3nLEST register

Note 1. In the case of transmission from the UART buffer, bit errors are detected even in the space between UART frames (inter-byte space).

Note 2. Setting the UPS[1:0] bits in the RLN3nLBFC register to 10_B (0 parity) disables the checking of parity bit values. In this case, no parity error is generated.

The error status is cleared by software or at a transition to LIN reset mode.

18.9 LIN Self-Test Mode

The LIN/UART interface provides a LIN self-test mode.

When the LIN/UART interface enters the LIN self-test mode, RLIN3nTX and RLIN3nRX are disconnected from external pins and RLIN3nTX and RLIN3nRX are connected to the LIN/UART interface internally. Therefore, the frame transmitted from RLIN3nTX is looped back to RLIN3nRX. The LIN self-test mode can perform tests exclusively in LIN mode.

The self-test can be performed in the following four modes:

- LIN master self-test mode (transmission): Header transmission and response transmission
- LIN master self-test mode (reception): Header transmission and response reception
- LIN slave self-test mode (transmission): Header reception and response transmission
- LIN slave self-test mode (reception): Header reception and response reception

In LIN self-test mode, operation is performed at the fastest baud rate, regardless of the setting of the baud rate generator.

Regardless of the setting of the baud rate related registers, the baud rate setting is the LIN communication clock source/16 [bps]. (The NSPB bits in the RLN3nLWBR register should be set to 0000_B or 1111_B.)

In addition, in LIN self-test mode, the following functions are not supported.

- LIN wake-up mode
- Frame separate mode
- Multi-byte response transmission/reception
- LIN slave mode (Auto baud rate)
- Frame/response timeout error

Do not use these functions.

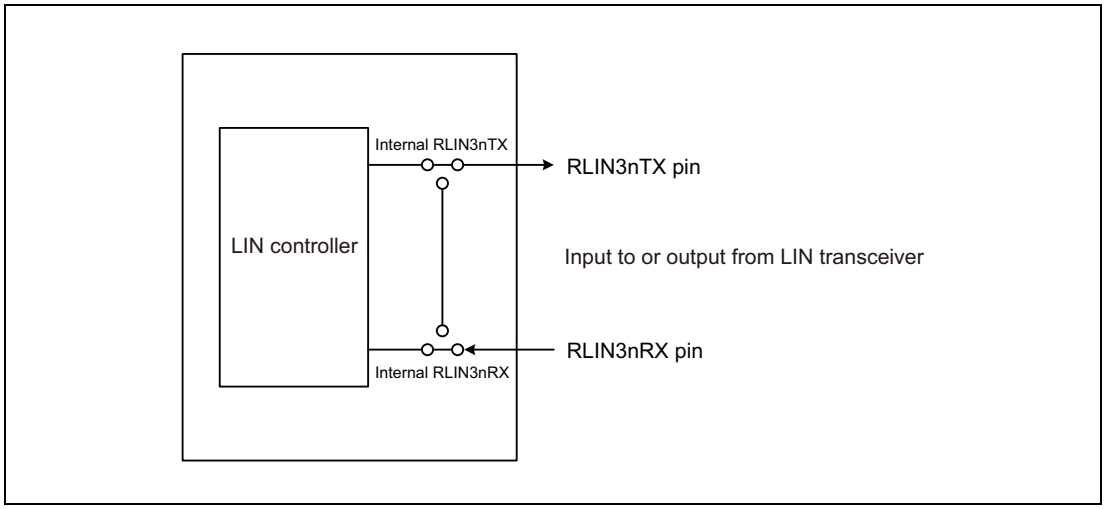


Figure 18.33 Connection in LIN Reset Mode, LIN Mode, and UART Mode

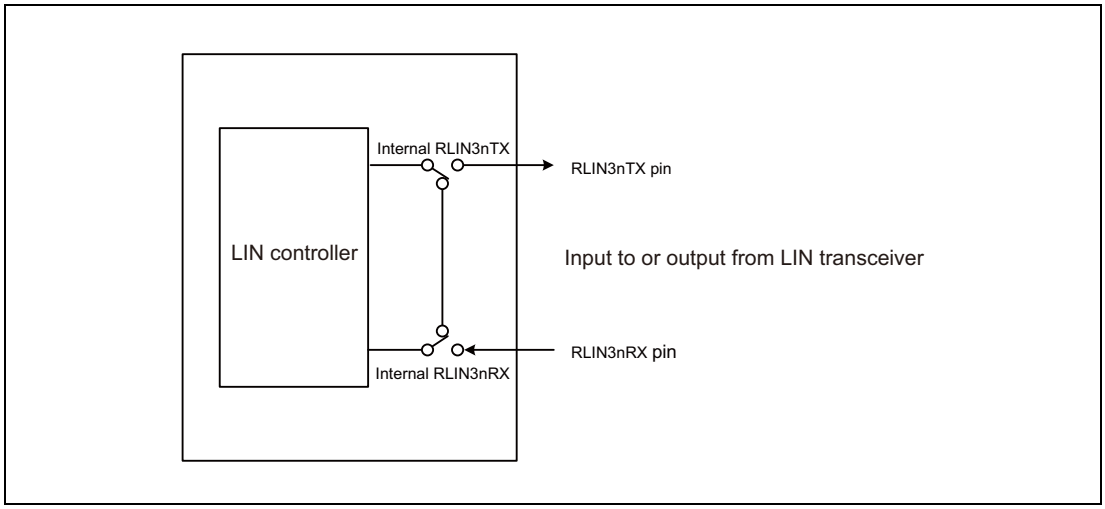


Figure 18.34 Connection in LIN Self-Test Mode

18.9.1 Transitioning to LIN Self-Test Mode

Writing to the RLN3nLSTC register makes a transition to the LIN self-test mode.
The LSTM bit in the RLN3nLSTC register set to 1 indicates that the mode has transitioned to the LIN self-test mode.

To transition to LIN self-test mode, be sure to execute a specific sequence. In that sequence, information must be written three times consecutively to the LIN self-test control register as shown below:

- Transition to LIN reset mode
  Set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode).
  Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).
- Select a LIN mode
  LMD bits in RLN3nLMD = 00_B (LIN master mode) or 11_B (LIN slave mode [fixed baud rate])
- 1st write: RLN3nLSTC register = $1010\ 0111_B$ ($A7_H$)
- 2nd write: RLN3nLSTC register = $0101\ 1000_B$ (58_H)
- 3rd write: RLN3nLSTC register = $0000\ 0001_B$ (01_H)
- Verify the transition to LIN self-test mode
  Read the LSTM bit in the RLN3nLSTC register; verify that it is 1 (LIN self-test mode).

If the key of the first write ($A7_H$) is written twice by mistake, the transition to LIN self-test mode is canceled. The above sequence should be retried from the step of first write. In addition, if a write to another LIN-related register is performed during transition to LIN self-test mode (three consecutive write operations to the RLN3nLSTC register), the transition is also canceled.

18.9.2 Transmission in LIN Master Self-Test Mode

To execute a self-test on LIN master transmission, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 $xxxx_B$*1
  RLN3nLBRP0 register = xxxx $xxxx_B$*1
  RLN3nLBRP1 register = xxxx $xxxx_B$*1
  RLN3nLMD register = 00xx $xx00_B$*1
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 $xxxx_B$*2
  RLN3nLEDE register = x000 $x0xx_B$
- Set the break field and space related registers.
  RLN3nLBFC register = 00xx $xxxx_B$
  RLN3nLSC register = 00xx $0xxx_B$
- Release from the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the transmit frame related registers.
  RLN3nLDFC register = 00x1 $xxxx_B$
  RLN3nLIDB register = xxxx $xxxx_B$
  RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx $xxxx_B$
- Header transmission → response transmission started
  Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/ reception started).
  The LIN master self-test mode (transmission) is executed. In this mode, interrupt are generated, and status and error status are also updated. The checksum is automatically calculated by the LIN/ UART interface. To suspend the LIN master self-test mode (transmission) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the transmission fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1 register and the LCKS bit in the RLN3nLMD register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 7, Exceptions and Interrupts**.

Note 3. When the successful header transmission interrupt and the successful frame transmission interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame transmission interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).
The time required from the setting of the successful header transmission flag to the setting of

the successful frame/wake-up transmission flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/frequency of LIN communication clock source × 16

18.9.3 Reception in LIN Master Self-Test Mode

To execute a self-test on LIN master reception, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 $xxxx_B$*1
  RLN3nLBRP0 register = xxxx $xxxx_B$*1
  RLN3nLBRP1 register = xxxx $xxxx_B$*1
  RLN3nLMD register = 00xx $xx00_B$*1
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 $xxxx_B$*2
  RLN3nLEDE register = x000 $x0xx_B$
- Set the break field and space related registers.
  RLN3nLBFC register = 00xx $xxxx_B$
  RLN3nLSC register = 00xx $0xxx_B$*1
- Release from the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the reception frame related registers.
  RLN3nLDFC register = 00x0 $xxxx_B$
  RLN3nLIDB register = xxxx $xxxx_B$
  RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx $xxxx_B$
  RLN3nCBR register = xxxx $xxxx_B$
  Since the checksum value to be transmitted is not automatically calculated, users must calculate it and set it to the RLN3nLCBR register. A checksum test can be performed by setting an incorrect checksum value here.
- Header transmission → response reception started
  Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/ reception started).
  The LIN master self-test mode (reception) is executed. In this mode, interrupt are generated, and status and error status are also updated. To suspend the LIN master self-test mode (reception) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the reception fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1

register, the LCKS bit in the RLN3nLMD register, and the IBS bit in the RLN3nLSC register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 7, Exceptions and Interrupts**.

Note 3. When the successful header transmission interrupt and the successful frame reception interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame reception interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).
The time required from the setting of the successful header transmission flag to the setting of the successful frame/wake-up reception flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/LIN communication clock source × 16

18.9.4 Transmission in LIN Slave Self-Test Mode

To execute a self-test on LIN slave transmission, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 xxx0_B*1
  RLN3nLBRP0 register = xxxx xxxx$_B$*1
  RLN3nLBRP1 register = xxxx xxxx$_B$*1
  RLN3nLMD register = 00x x0011_B
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 xxxx$_B$*2
  RLN3nLEDE register = xx0x x$00x_B$
- Set the break field and space related registers.
  RLN3nLBFC register = 0000 000x$_B$*3
  RLN3nLSC register = 00xx 0001_B
- Release from the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the transmit frame related registers.
  RLN3nLDFC register = 00x1 xxxx$_B$
  RLN3nLIDB register = xxxx xxxx$_B$
  RLN3nLDBR1 to RLN3nLDBR8 registers = xxxx xxxx$_B$
- Header reception → response transmission started
  Set the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/ reception started).
  (The header reception and the response transmission are executed in this order, without manipulating the RTS bit in the RLN3nLTRC register.)
  The LIN slave self-test mode (transmission) is executed. In this mode, interrupt are generated, and status and error status are also updated.
  The checksum is automatically calculated by the LIN/UART interface. To suspend the LIN slave self-test mode (transmission) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a

reversed value because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.

- If the transmission fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, and the RLN3nLBRP1 register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 7, Exceptions and Interrupts**

Note 3. According to the setting of this register, 9.5-Tbit or 10.5-Tbit width break is output from the internal RLIN3nTX.

Note 4. When the successful header reception interrupt and the successful response transmission interrupt are used in the same interrupt processing, if the software processing of the successful header reception interrupt is not completed before the generation of the successful response transmission interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header reception interrupt enabled).
The time required from the setting of the successful header reception flag to the setting of the successful response/wake-up transmission flag is calculated by using the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/frequency of LIN communication clock source × 16

18.9.5 Reception in LIN Slave Self-Test Mode

To execute a self-test on LIN slave reception, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 $xxx0_B$*1
  RLN3nLBRP0 register = xxxx $xxxx_B$*1
  RLN3nLBRP1 register = xxxx $xxxx_B$*1
  RLN3nLMD register = 00xx 0011_B
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 $xxxx_B$*2
  RLN3nLEDE register = xx0x $x00x_B$
- Set the break field and space related registers.
  RLN3nLBFC register = 0000 $000x_B$*3
  RLN3nLSC register = 00xx 0001_B*1
- Release from the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the reception frame related registers.
  RLN3nLDFC register = 00x0 $xxxx_B$
  RLN3nLIDB register = xxxx $xxxx_B$
  RLN3nLDBR1 to RLN3nLDBR8 registers = xxxx $xxxx_B$
  RLN3nCBR register = xxxx $xxxx_B$
  Since the checksum value to be transmitted is not automatically calculated, users must calculate it and set it to the RLN3nLCBR register. A checksum test can be performed by setting an incorrect checksum value here.
- Header reception → response reception started
  Set the FTS bit in the RLN3nLTRC register to 1 (header reception or wake-up transmission/ reception started).
  (Without any setting of the RTS bit in the RLN3nLTRC register, the header reception and the response reception are executed in this order.)
  The LIN slave self-test mode (reception) is executed. In this mode, interrupt are generated, and status and error status are also updated. To suspend the LIN slave self-test mode (reception) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is stored as a reversed value because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the reception fails to complete due to an error, the corresponding error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Arbitrary value can be specified.

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The LPRS bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1 register, and the IBS bit in the RLN3nLSC register.
Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 7, Exceptions and Interrupts**.

Note 3. According to the setting of this register, 9.5-Tbit or 10.5-Tbit width break is output from the internal RLIN3nTX.

Note 4. When the successful header reception interrupt and the successful response reception interrupt are used in the same interrupt processing, if the software processing of the successful header reception interrupt is not completed before the generation of the successful response reception interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header reception interrupt enabled).
The time required from the setting of the successful header reception flag to the setting of the successful response/wake-up reception flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/frequency of LIN communication clock source × 16

18.9.6 Terminating LIN Self-Test Mode

To terminate LIN self-test mode, perform the procedure below:

- Write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register.
  If the OMM1 and OMM0 bits in the RLN3nLMST register are not 11_B, write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register. After confirming that the OMM1 and OMM0 bits in the RLN3nLMST register are set to 11_B, transition to LIN reset mode.
- Verify the cancelation of LIN self-test mode.
  Read the LSTM bit in the RLN3nLSTC register; confirm that it is 0 (not in LIN self-test mode).
- Verify the transition to LIN reset mode.
  Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).

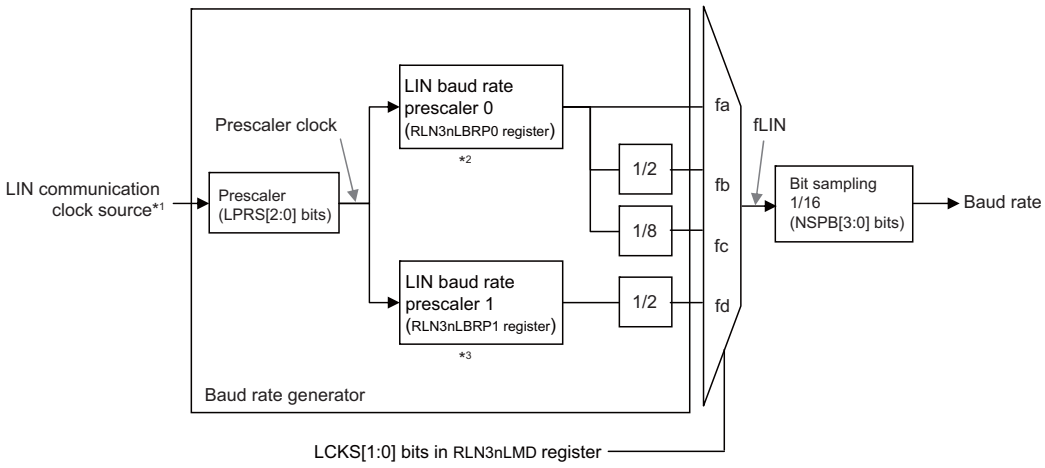
18.10 Baud Rate Generator

The prescaler clock is obtained by frequency-dividing the LIN communication clock source by the prescaler, and the LIN system clock (fLIN) is obtained by frequency-dividing the prescaler clock by the baud rate generator. The clock obtained by frequency-dividing the LIN system clock (fLIN) by the number of samplings is the baud rate. The reciprocal of this baud rate is called the bit time (Tbit).

The LIN/UART interface has two types of baud rate generators. The baud rate generator to be used is switched according to the mode.

18.10.1 LIN Master Mode

Figure 18.35 shows a block diagram of baud rate generation in LIN master mode.



Note 1. For the LIN communication clock source, see **Section 11.5.3, Clock Domain Settings**.
Note 2. When the value in RLN3nLBRP0 register is N (N = 0 to 255), the clock frequency is divided by N + 1.
Note 3. When the value in RLN3nLBRP1 register is M (M = 0 to 255), the clock frequency is divided by M + 1.

Figure 18.35 Block Diagram of Baud Rate Generation in LIN Master Mode

By setting the RLN3nLBRP0 register so that fa is 307200 Hz (= 19200 × 16), the resulting system clock frequencies are fa = 19200 × 16, fb = 9600 × 16, and fc = 2400 × 16. These system clock frequencies are divided by 16 in the bit timing generator, enabling bit rates of 19200 bps, 9600 bps and 2400 bps, to be generated. Also, by setting the RLN3nLBRP1 register so that fd is 166672 Hz (= 10417 × 16), the resulting system clock frequency is fd = 10417 × 16. This system clock frequency is divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

The formula for calculating the baud rate is described below.

Baud rate of LIN master

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)

÷ (RLN3nLBRP0+ 1) ÷ 16 [bps] (When fa is selected for f_{LIN})

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)

÷ (RLN3nLBRP0 + 1) ÷ 2 ÷ 16 [bps] (When fb is selected for f_{LIN})

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)

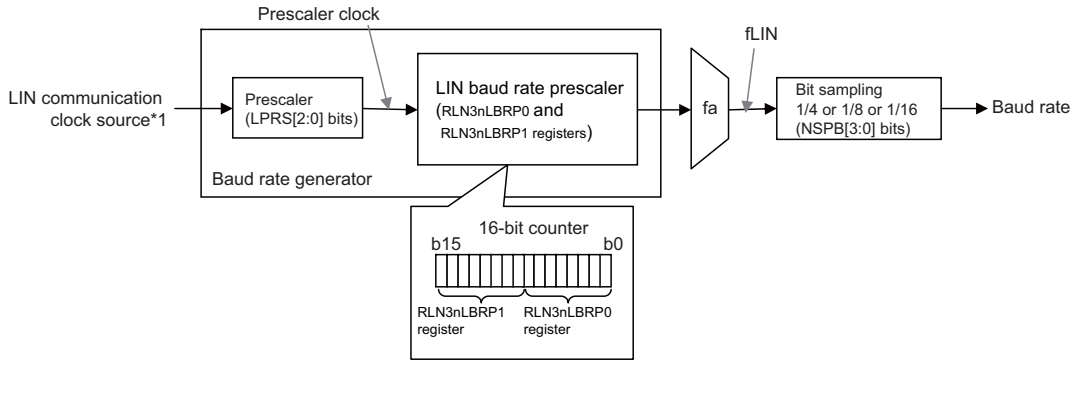
÷ (RLN3nLBRP0 + 1) ÷ 8 ÷ 16 [bps] (When fc is selected for f_{LIN})

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)

÷ (RLN3nLBRP1 + 1) ÷ 2 ÷ 16 [bps] (When fd is selected for f_{LIN})

18.10.2 LIN Slave Mode

Figure 18.36 shows a block diagram of baud rate generation in LIN slave mode.



Note 1. For the LIN communication clock source, see **Section 11.5.3, Clock Domain Settings**.

Figure 18.36 Block Diagram of Baud Rate Generation in LIN Slave Mode

In LIN slave mode (auto baud rate), the baud rate can be set to 1 kbps to 20 kbps. Set the prescaler clock as follows according to the target baud rate:

[Target baud rate]	[Prescaler clock]
1 kbps to 20 kbps	: 4MHz *1
1 kbps to 2.4 kbps (excluding 2.4 kbps)	: 4MHz
2.4 kbps to 20 kbps	: 8 MHz to 12 MHz

Note 1. Use the clock with NSPB[3:0] bits in the RLN3nLWBR register set to "0011_B" (four samplings).

The formula for baud rate is described below.

Baud rate of LIN slave

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)

÷ (RLN3nLBRP01 + 1) ÷ 16 [bps] ([Fixed baud rate])

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)

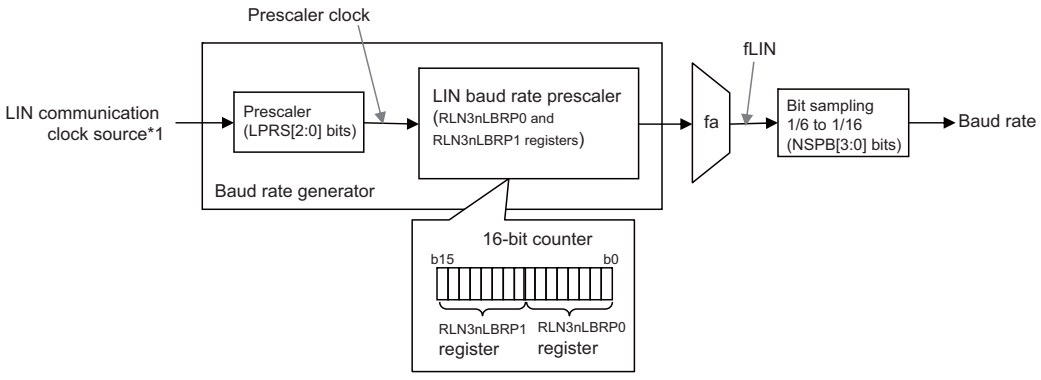
÷ (RLN3nLBRP01 + 1) ÷ 4 or 8 [bps] ([Auto baud rate])

NOTE

For a LIN slave with fixed baud rate, set the NSPB[3:0] bits to "0000_B" (16 samples) or "1111_B" (16 samples). For a LIN slave with auto baud rate, set the NSPB[3:0] bits to "0011_B" (4 samples) or "0100_B" (8 samples).

18.10.3 UART Mode

Figure 18.37 shows a block diagram of baud rate generation in UART mode.



Note 1. For the LIN communication clock source, see **Section 11.5.3, Clock Domain Settings**.

Figure 18.37 Block Diagram of Baud Rate Generation in UART Mode

UART baud rate is calculated with the following formula:

UART baud rate
= {LIN communication clock source frequency} × (RLN3nLWBR.LPRS[2:0] selection clock) ÷ (RLN3nLBRP01 + 1) ÷ {RLN3nLWBR.NSPB[3:0] selection count} [bps]

18.11 Noise Filter

The LIN/UART interface has a noise filter for reducing erroneous receiving of data due to noise. By setting the LRDNFS bit in the RLN3nLMD register to 0 (use the noise filter), the noise filter is enabled. The noise filter samples the level of the synchronized RLIN3nRX with the prescaler clock, and outputs the sampling value determined by a 3-samplings majority rule. The value of each bit of the receive data is determined based on the noise filter output.

Figure 18.38 shows the configuration of the noise filter, **Figure 18.39** shows an example of a noise filter circuit, and **Figure 18.40** shows the determination of the received data when the noise filter is used.

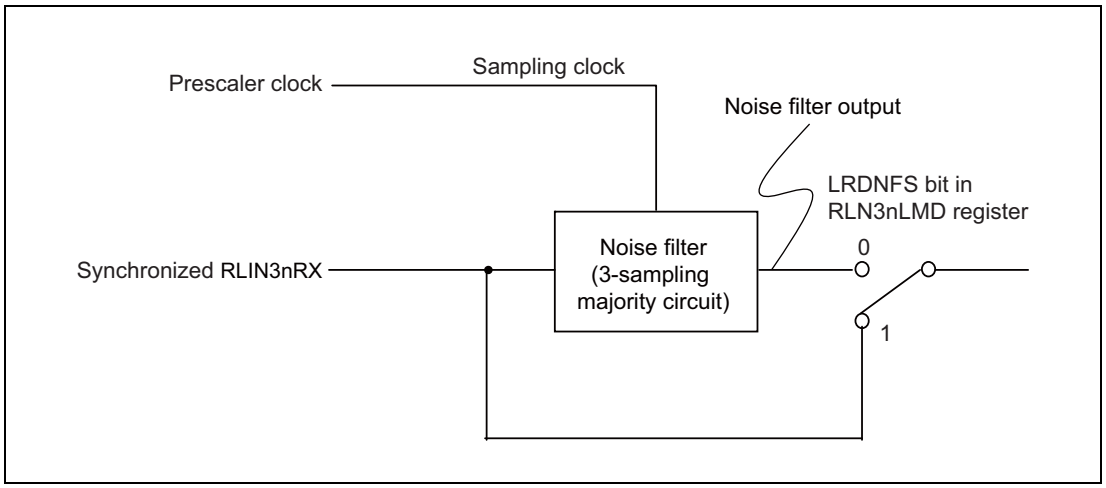


Figure 18.38 Configuration of Noise Filter

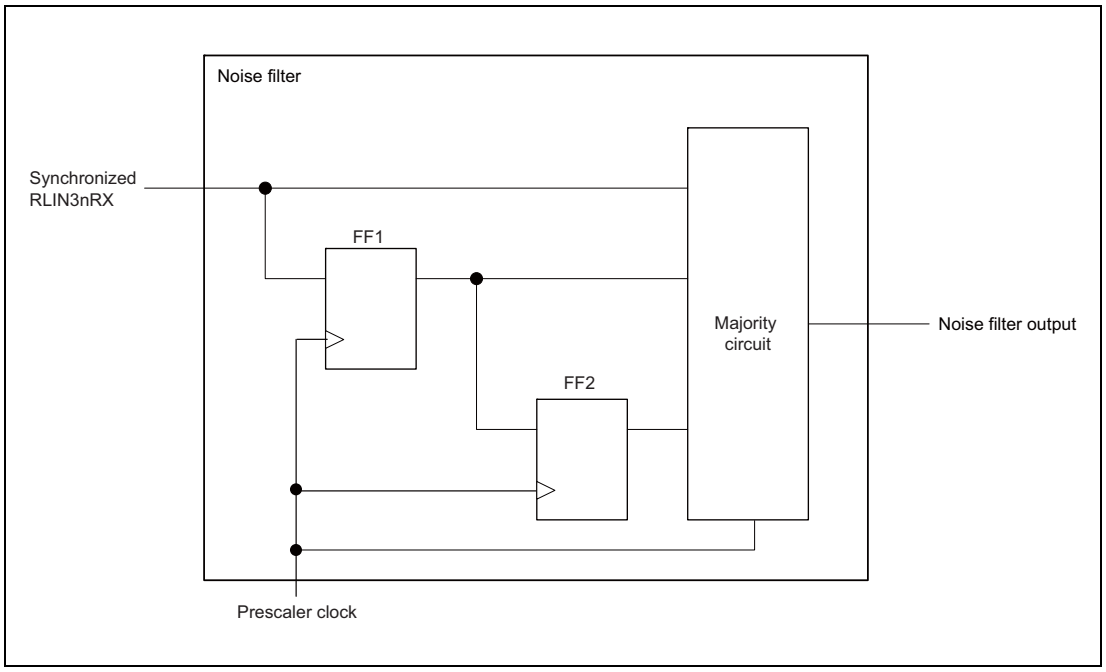


Figure 18.39 Example of Noise Filter Circuit

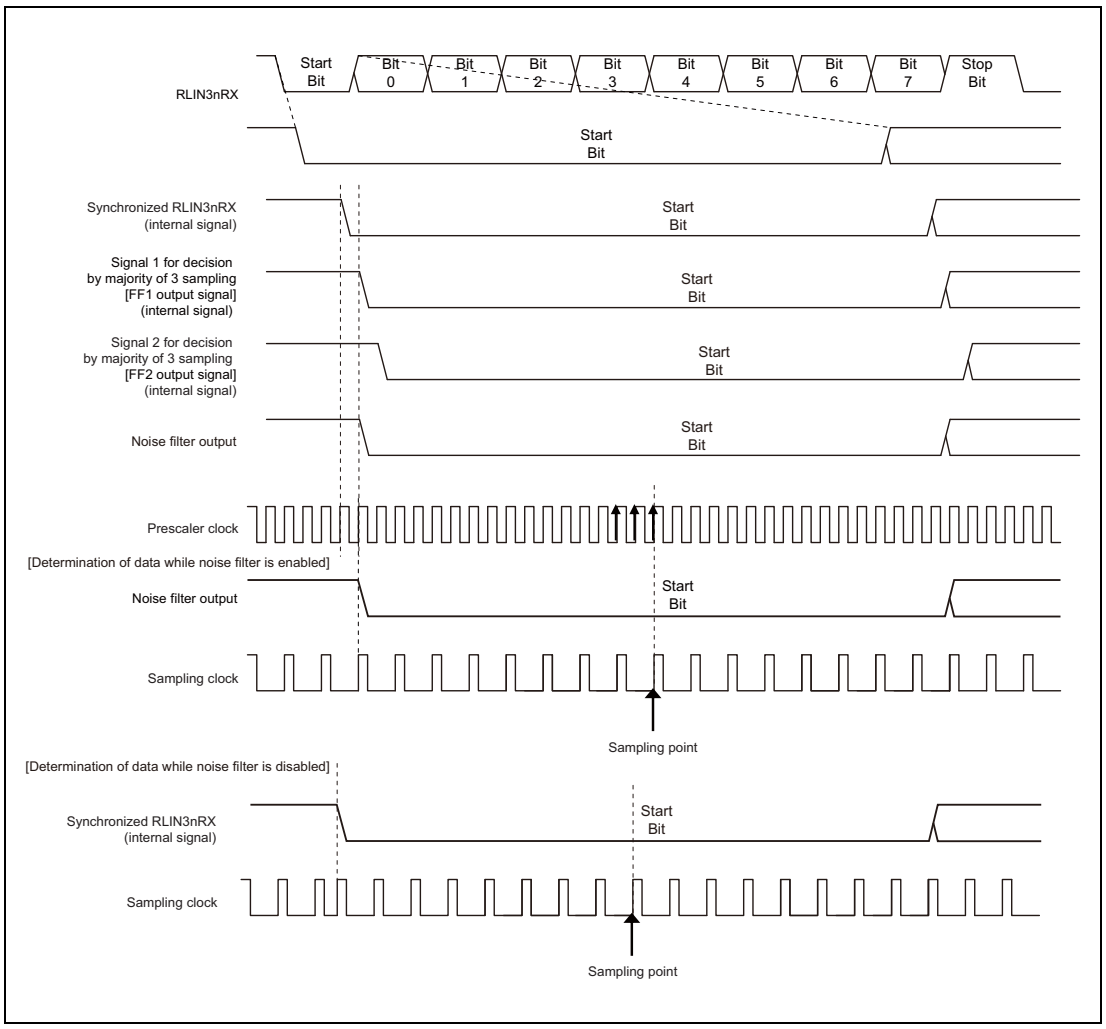


Figure 18.40 Determination of Received Data when Noise Filter is Used

Section 19 I2C Bus Interface (RIIC)

This section contains a generic description of the I2C Bus Interface (RIIC).

The first part of this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RIIC.

19.1 Features of RH850/F1M RIIC

19.1.1 Number of Units and Channels

This microcontroller has the following number of RIIC units.

Each RIIC unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 19.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	1		
Name	RIICn (n = 0)		

Table 19.2 Index

Index	Description
n	Throughout this section, the individual RIIC units are identified by the index “n” (n = 0): for example, RIICnCR1 is the I2C bus control register 1.

19.1.2 Register Base Address

RIIC base address is listed in the following table.

RIIC register addresses are given as offsets from the base address.

Table 19.3 Register Base Address

Base Address Name	Base Address
<RIIC0_base>	FFCA 0000_H

19.1.3 Clock Supply

The RIIC clock supply is shown in the following table.

Table 19.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RIICn	PCLK*1	CKSCLK_IIIC
	Register access clock	CKSCLK_IIIC

Note 1. Set PCLK to a value that is less than 1/2 the SCL clock (high level width).

19.1.4 Interrupt Requests

RIIC interrupt requests are listed in the following table.

Table 19.5 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
RIIC0			
INTIICnEE	RIIC communication error / event generation interrupt	79	—
INTIICnRI	RIIC receive end interrupt	78	20
INTIICnTI	RIIC transmit data empty interrupt	76	19
INTIICnTEI	RIIC transmit end interrupt	77	—

19.1.5 Reset Sources

RIIC reset sources are listed in the following table. RIIC is initialized by these reset sources.

Table 19.6 Reset Sources

Unit Name	Reset Source
RIIC0	All reset sources (ISORES)

19.1.6 External Input/Output Signals

External input/output signals of RIIC are listed below.

Table 19.7 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
RIIC0		
RIICnSCL	Serial clock I/O pin	RIIC0SCL
RIICnSDA	Serial data I/O pin	RIIC0SDA

When using these ports, the PBDCn register for the corresponding port and the corresponding bit in the PODCn register must be set to 1.

19.2 Overview

19.2.1 Functional Overview

Communications format

- I2C bus format
- Master mode or slave mode selectable
- Automatic securing of the various set-up times, hold times, and bus-free times for the transfer rate

Transfer rate

Up to 400 kbps

SCL clock

- For master operation, the duty cycle of the SCL clock is selectable in the following range:
  - 0% < Duty < 100%

Issuing and detecting conditions

Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.

Slave address

- Up to three slave-address settings can be specified.
- Seven- and ten-bit address formats are supported (along with the use of both at once).
- General call addresses and device ID addresses are detectable.

Acknowledgement

- For transmission, the acknowledge bit is automatically loaded
  - Transfer of the next data for transmission can be automatically suspended when not-acknowledge is received.
- For reception, the acknowledge bit is automatically transmitted
  - If a wait between the eighth and ninth clock cycles has been selected, acknowledge bit response can be controlled by software based on the reception data contents.

Wait function

- In reception, the following periods of waiting can be obtained by holding the clock signal (SCL) at the low level:
  - Waiting between the eighth and ninth clock cycles
  - Waiting between the ninth clock cycle and the first clock cycle of the next transfer (WAIT function)

SDA output delay function

Timing of the output of transmitted data, including the acknowledge bit, can be delayed.

Arbitration

- For multi-master operation
  - Operation to synchronize the SCL (clock) signal in cases of conflict with the SCL signal from another master is possible.
  - When issuing the start condition would create a conflict, loss of arbitration can be detected if the level of the internal SDA output doesn't match the level on the SDA line.
  - In master operation, loss of arbitration can be detected if the level of the internal SDA output doesn't match the level on the SDA line.
- Loss of arbitration due to detection of the start condition while the bus is busy can be detected (to prevent double issue of start conditions).
- Loss of arbitration in transfer of a not-acknowledge bit can be detected if the level of the internal SDA output doesn't match the level on the SDA line.
- Loss of arbitration can be detected if the level of the internal SDA output doesn't match the level on the SDA line in slave transmission.

Timeout function

The internal timeout detection function is capable of detecting long-interval stop of the SCL (clock signal).

Noise elimination

The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise elimination by the filters is adjustable by software.

Interrupt sources

- Four sources:
  - Communication error or event occurrence (detection of arbitration-loss, NACK, timeout, a start condition including a restart condition, or a stop condition)
  - Receive complete (including matching with a slave address)
  - Transmit-data-empty (including matching with a slave address)
  - Transmission complete

19.2.2 Block Diagram

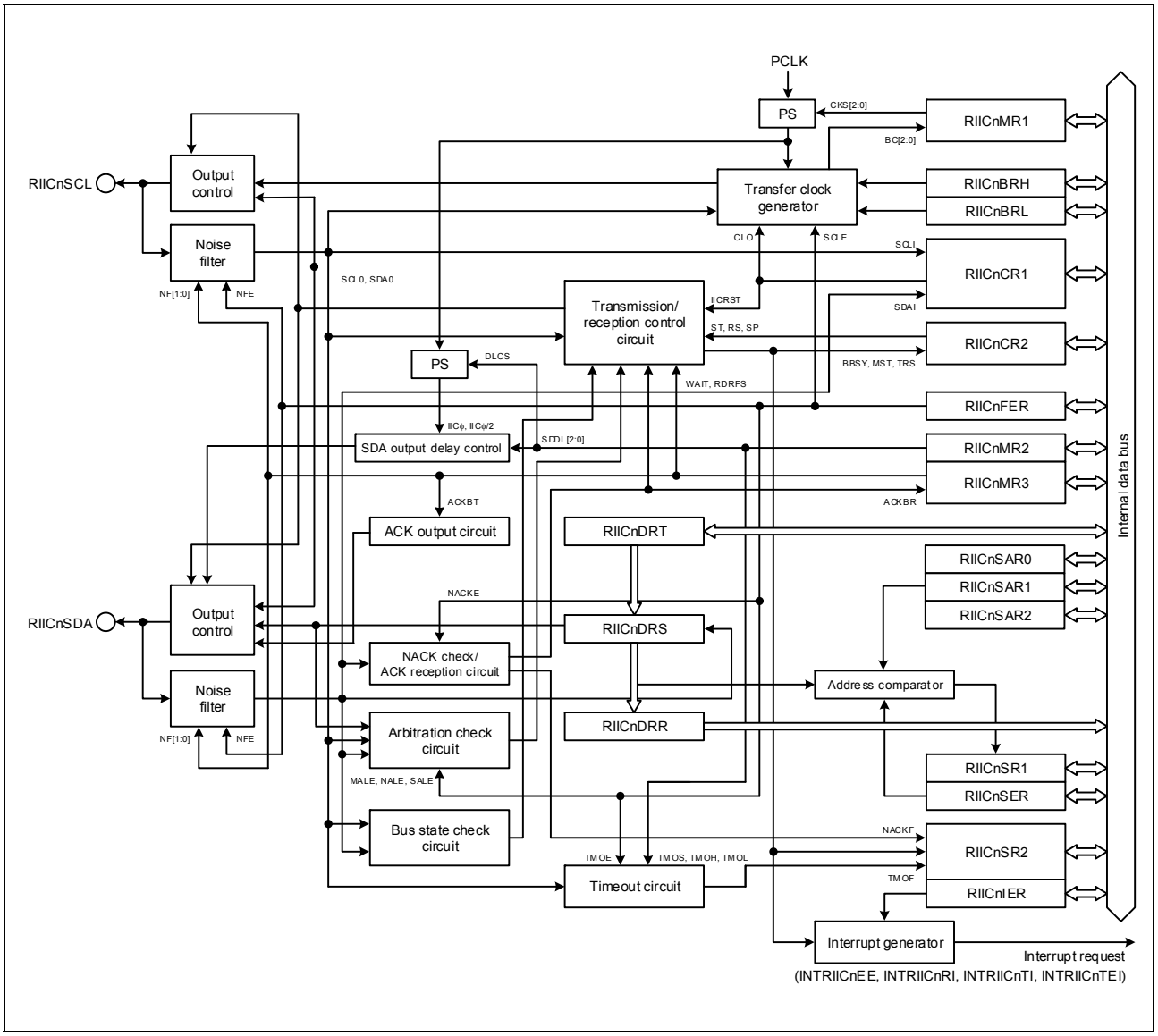


Figure 19.1 Block Diagram of RIIC

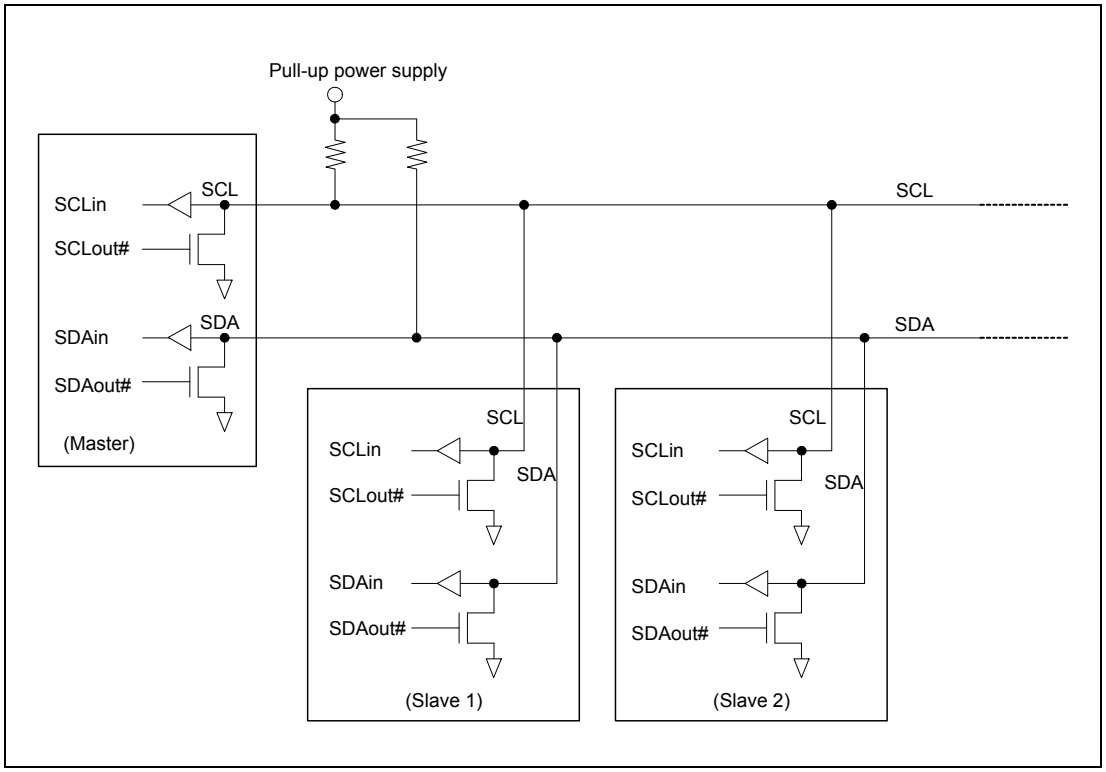


Figure 19.2 Example of Connections to the External Circuit by the I/O Pins (I2C Bus Configuration Example)

19.3 Registers

19.3.1 List of Registers

RIIC registers are listed in the table below.

For details about <RIICn_base>, see **Section 19.1.2, Register Base Address**.

Table 19.8 List of Registers

Unit Name	Register Name	Symbol	Address
RIICn	I2C Bus Control Register 1	RIICnCR1	<RIICn_base> + 0000_H
RIICn	I2C Bus Control Register 2	RIICnCR2	<RIICn_base> + 0004_H
RIICn	I2C Bus Mode Register 1	RIICnMR1	<RIICn_base> + 0008_H
RIICn	I2C Bus Mode Register 2	RIICnMR2	<RIICn_base> + $000C_H$
RIICn	I2C Bus Mode Register 3	RIICnMR3	<RIICn_base> + 0010_H
RIICn	I2C Bus Function Enable Register	RIICnFER	<RIICn_base> + 0014_H
RIICn	I2C Bus Status Enable Register	RIICnSER	<RIICn_base> + 0018_H
RIICn	I2C Bus Interrupt Enable Register	RIICnIER	<RIICn_base> + $001C_H$
RIICn	I2C Bus Status Register 1	RIICnSR1	<RIICn_base> + 0020_H
RIICn	I2C Bus Status Register 2	RIICnSR2	<RIICn_base> + 0024_H
RIICn	I2C Slave Address Register 0	RIICnSAR0	<RIICn_base> + 0028_H
RIICn	I2C Slave Address Register 1	RIICnSAR1	<RIICn_base> + $002C_H$
RIICn	I2C Slave Address Register 2	RIICnSAR2	<RIICn_base> + 0030_H
RIICn	I2C Bus Bit Rate Low-Level Register	RIICnBRL	<RIICn_base> + 0034_H
RIICn	I2C Bus Bit Rate High-Level Register	RIICnBRH	<RIICn_base> + 0038_H
RIICn	I2C Bus Transmit Data Register	RIICnDRT	<RIICn_base> + $003C_H$
RIICn	I2C Bus Receive Data Register	RIICnDRR	<RIICn_base> + 0040_H
RIICn	I2C Bus Shift Register	RIICnDRS	—

19.3.2 RIICnCR1 — I2C Bus Control Register 1

Access: RIICnCR1 register can be read or written in 32-bit units.
RIICnCR1L and RIICnCR1H registers can be read or written in 16-bit units.
RIICnCR1LL, RIICnCR1LH, RIICnCR1HL, and RIICnCR1HH registers can be read or written in 8-bit units.

Address: RIICnCR1: <RIICn_base> + 0000_H

RIICnCR1L: <RIICn_base> + 0000_H, RIICnCR1H: <RIICn_base> + 0002_H

RIICnCR1LL: <RIICn_base> + 0000_H, RIICnCR1LH: <RIICn_base> + 0001_H,
RIICnCR1HL: <RIICn_base> + 0002_H, RIICnCR1HH: <RIICn_base> + 0003_H

Value after reset: 0000 $001F_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	W	R/W	R/W	R	R

Table 19.9 RIICnCR1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ICE	I2C Bus Interface Enable 0: Disabled (the RIICnSCL and RIICnSDA pins are not driven). 1: Enabled (the RIICnSCL and RIICnSDA pins are driven). (This bit selects an RIIC reset or internal reset in combination with the IICRST bit.)
6	IICRST	I2C Bus Interface Internal Reset 0: Clears the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL/SDA output latch)
5	CLO	Extra SCL Clock Cycle Output 0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle. (This bit is cleared automatically after one clock cycle is output.)
4	SOWP	SCLO/SDAO Write Protect 0: SCLO and SDAO bits can be written. 1: SCLO and SDAO bits are protected. (When read, 1 is returned.)
3	SCLO	SCL Output Control/Monitor • Read: 0: The RIIC has driven the RIICnSCL pin low. 1: The RIICnSCL pin has been released. • Write: 0: The RIIC drives the RIICnSCL pin low. 1: Releases the RIICnSCL pin.

Table 19.9 RIICnCR1 Register Contents (2/2)

Bit Position	Bit Name	Function
2	SDAO	SDA Output Control/Monitor • Read: 0: The RIIC has driven the RIICnSDA pin low. 1: The RIICnSDA pin has been released. • Write: 0: The RIIC drives the RIICnSDA pin low. 1: Releases the RIICnSDA pin.
1	SCLI	SCL Line Monitor 0: RIICnSCL line is low. 1: RIICnSCL line is high.
0	SDAI	SDA Line Monitor 0: RIICnSDA line is low. 1: RIICnSDA line is high.

SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)

These bits are used to directly control the RIICnSDA and RIICnSCL signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a start condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, or during transmission or reception. Operation after rewriting in the periods mentioned above is not guaranteed.

When reading these bits, the state of signals output from the RIIC is returned.

CLO Bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.

Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.

For details on this function, see **Section 19.13.2, Extra SCL Clock Cycle Output Function**.

IICRST Bit (I2C Bus Internal Reset)

This bit is used to reset the internal states of the RIIC.

Setting this bit to 1 initiates an RIIC reset or internal reset.

Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. **Table 19.10** lists the types of RIIC reset.

The RIIC reset resets all registers (except ICE and IICRST) including the RIICnCR2.BBSY flag and internal states of the RIIC, and the internal reset resets the bit counter (RIICnMR1.BC[2:0] bits), the I2C bus shift register (RIICnDRS), and the I2C bus status registers (RIICnSR1 and RIICnSR2) as well as the internal states of the RIIC. For the reset conditions for each register, see **Section 19.14, Reset Function of RIIC**.

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 RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error, etc.

If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the RIICnSCL pin and RIICnSDA pin at a high impedance.

CAUTION

If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but instead initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCL line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 19.10 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers (except ICE and IICRST) and internal states of the RIIC.
	1	Internal reset	Reset the RIICnMR1.BC[2:0] bits, and the RIICnSR1, RIICnSR2, RIICnDRS registers and the internal states of the RIIC.

ICE Bit (I2C Bus Interface Enable)

The ICE bit selects driving or non-driving of the RIICnSCL and RIICnSDA pins. Moreover, this bit can perform two types of reset in combination with the IICRST bit. For the types of reset, see **Table 19.10, RIIC Resets**.

Set the ICE bit to 1 when using RIIC. Setting the ICE bit to 1 selects driving of the RIICnSCL and RIICnSDA pins.

Set the ICE bit to 0 when RIIC is not to be used. Clearing the ICE bit to 0 stops driving of the RIICnSCL and RIICnSDA pins.

CAUTION

Though the output from RIICnSDA or RIICnSCL is disabled while the ICE bit is 0, the input to RIICnSDA or RIICnSCL is enabled. The RIICnSCL and RIICnSDA pin functions should not be assigned to the RIIC. If assigned, it causes the slave addresses to be compared.

19.3.3 RIICnCR2 — I2C Bus Control Register 2

Access: RIICnCR2 register can be read or written in 32-bit units.
RIICnCR2L and RIICnCR2H registers can be read or written in 16-bit units.
RIICnCR2LL, RIICnCR2LH, RIICnCR2HL, and RIICnCR2HH registers can be read or written in 8-bit units.

Address: RIICnCR2: <RIICn_base> + 0004_H

RIICnCR2L: <RIICn_base> + 0004_H, RIICnCR2H: <RIICn_base> + 0006_H

RIICnCR2LL: <RIICn_base> + 0004_H, RIICnCR2LH: <RIICn_base> + 0005_H,
RIICnCR2HL: <RIICn_base> + 0006_H, RIICnCR2HH: <RIICn_base> + 0007_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R

Table 19.11 RIICnCR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	BBSY	Bus Busy Detection Flag 0: The I2C bus is released (the bus is free). 1: The I2C bus is occupied (the bus is busy).
6	MST*1	Master/Slave Mode 0: Slave mode 1: Master mode
5	TRS*1	Transmit/Receive Mode 0: Receive mode 1: Transmit mode
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SP	Stop Condition Issuance Request 0: Does not request to issue a stop condition. 1: Requests to issue a stop condition.
2	RS	Restart Condition Issuance Request 0: Does not request to issue a restart condition. 1: Requests to issue a restart condition.
1	ST	Start Condition Issuance Request 0: Does not request to issue a start condition. 1: Requests to issue a start condition.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. When the RIICnMR1.MTWP bit is set to 1, the MST and TRS bits can be written to.

ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free).

For details on the start condition issuance, see **Section 19.12, Start Condition/Restart Condition/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 has been issued
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free).
Note that arbitration may be lost as the start condition issuance error if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy).

RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.

When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the restart condition issuance, see **Section 19.12, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the RS bit with the RIICnCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued or a start condition is detected
- When a stop condition is detected
- When the RIICnCR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTIONS

1. Do not set the RS bit to 1 while issuing a stop condition.
2. It is commended to issue a restart condition in master transmit mode. If the RS bit is set to 1 (restart condition issuance request) in mode other than master mode, the restart condition is not issued in this mode but the RS bit remains set. If the operating mode changes to master mode with the bit not being cleared, the restart condition may be issued.

SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.

When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the stop condition issuance, see **Section 19.12, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the SP bit with both the RIICnCR2.BBSY flag and the RIICnCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued or a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTIONS

1. Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free).
2. Do not set the SP bit to 1 while a restart condition is being issued.

TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.

The value of the TRS bit is automatically changed to the value for transmission mode or reception mode (1 or 0) by detection or issuing of a start condition, setting or clearing of the R/W# bit, etc. Although writing to the TRS bit is possible when the RIICnMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When the address received in slave mode matches the address enabled in RIICnSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the RIICnMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The RIICnSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave transmit mode, a restart condition is detected (a restart condition is detected with RIICnCR2.BBSY = 1 and RIICnCR2.MST = 0)

- When 0 is written to the TRS bit with the RIICnMR1.MTWP bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.

The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.

The value of the MST bit is automatically changed to the value for master mode or slave mode (1 or 0) by detection or issuing of a start condition, etc. Although writing to the MST bit is possible when the RIICnMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the RIICnMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 0 is written to the MST bit with the RIICnMR1.MTWP bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection)

The BBSY flag indicates whether the I2C bus is occupied (bus busy) or released (bus free).

This bit is set to 1 when the SDA line changes from high to low under the condition of SCL = high, assuming that a start condition has been issued.

When the SDA line changes from low to high under the condition of SCL = high, this bit is cleared to 0 after the bus free time (specified in RIICnBRL) start condition is not detected, assuming that a stop condition has been issued.

[Setting condition]

When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in RIICnBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the RIICnCR1.IICRST bit with the RIICnCR1.ICE bit set to 0 (RIIC reset)

CAUTIONS

- When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.
- When an internal reset is applied while the bus is not free, the BBSY flag is not cleared.

19.3.4 RIICnMR1 — I2C Bus Mode Register 1

Access: RIICnMR1 register can be read or written in 32-bit units.
RIICnMR1L and RIICnMR1H registers can be read or written in 16-bit units.
RIICnMR1LL, RIICnMR1LH, RIICnMR1HL, and RIICnMR1HH registers can be read or written in 8-bit units.

Address: RIICnMR1: <RIICn_base> + 0008_H

RIICnMR1L: <RIICn_base> + 0008_H, RIICnMR1H: <RIICn_base> + $000A_H$

RIICnMR1LL: <RIICn_base> + 0008_H, RIICnMR1LH: <RIICn_base> + 0009_H,
RIICnMR1HL: <RIICn_base> + $000A_H$, RIICnMR1HH: <RIICn_base> + $000B_H$

Value after reset: 0000 0008_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W

Table 19.12 RIICnMR1 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	MTWP	MST/TRS Write Protect 0: Disables writing to the RIICnCR2.MST and TRS bits. 1: Enables writing to the RIICnCR2.MST and TRS bits.
6 to 4	CKS[2:0]	Internal Reference Clock Selection (IICϕ) b6 b4 0 0 0: PCLK/1 clock 0 0 1: PCLK/2 clock 0 1 0: PCLK/4 clock 0 1 1: PCLK/8 clock 1 0 0: PCLK/16 clock 1 0 1: PCLK/32 clock 1 1 0: PCLK/64 clock 1 1 1: PCLK/128 clock
3	BCWP*1	BC Write Protect 0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.) 1: Protects the BC[2:0] bits.
2 to 0	BC[2:0]	Bit Counter b2 b0 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

Note 1. When rewriting the BC[2:0] bits, write 0 to the BCWP bit simultaneously.

BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred frames. When setting any value other than 000_B, set the value while the SCL line is at a low level.

[Clearing conditions]

- When 1 is written to the RIICnCR1.IICRST bit to apply a RIIC reset or an internal reset is initiated.
- Data transfer including the acknowledge bit being completed.
- A start condition including a restart condition being detected.

19.3.5 RIICnMR2 — I2C Bus Mode Register 2

Access: RIICnMR2 register can be read or written in 32-bit units.
RIICnMR2L and RIICnMR2H registers can be read or written in 16-bit units.
RIICnMR2LL, RIICnMR2LH, RIICnMR2HL, and RIICnMR2HH registers can be read or written in 8-bit units.

Address: RIICnMR2: <RIICn_base> + $000C_H$

RIICnMR2L: <RIICn_base> + $000C_H$, RIICnMR2H: <RIICn_base> + $000E_H$

RIICnMR2LL: <RIICn_base> + $000C_H$, RIICnMR2LH: <RIICn_base> + $000D_H$,
RIICnMR2HL: <RIICn_base> + $000E_H$, RIICnMR2HH: <RIICn_base> + $000F_H$

Value after reset: $0000\ 0006_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 19.13 RIICnMR2 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	DLCS	SDA Output Delay Clock Source Selection 0: The internal reference clock (IICϕ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICϕ/2) is selected as the clock source of the SDA output delay counter.*1
6 to 4	SDDL[2:0]	SDA Output Delay Counter • When RIICnMR2.DLCS = 0 (IICϕ) b6 b4 0 0 0: No output delay 0 0 1: 1 IICϕ cycle 0 1 0: 2 IICϕ cycles 0 1 1: 3 IICϕ cycles 1 0 0: 4 IICϕ cycles 1 0 1: 5 IICϕ cycles 1 1 0: 6 IICϕ cycles 1 1 1: 7 IICϕ cycles • When RIICnMR2.DLCS = 1 (IICϕ/2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICϕ cycles 0 1 0: 3 or 4 IICϕ cycles 0 1 1: 5 or 6 IICϕ cycles 1 0 0: 7 or 8 IICϕ cycles 1 0 1: 9 or 10 IICϕ cycles 1 1 0: 11 or 12 IICϕ cycles 1 1 1: 13 or 14 IICϕ cycles
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TMOH	Timeout H Count Control 0: Disables counting while the SCL line is at a high level. 1: Enables counting while the SCL line is at a high level.

Table 19.13 RIICnMR2 Register Contents (2/2)

Bit Position	Bit Name	Function
1	TMOL	Timeout L Count Control 0: Disables counting while the SCL line is at a low level. 1: Enables counting while the SCL line is at a low level.
0	TMOS	Timeout Detection Time Selection 0: Long mode is selected. 1: Short mode is selected.

Note 1. The setting DLCS = 1 (IICϕ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting DLCS = 1 becomes invalid and the clock source becomes the internal reference clock (IICϕ).

TMOS Bit (Timeout Detection Time Selection)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (RIICnFER.TMOE bit = 1). When this bit is set to 0, long mode is selected. When this bit 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 SCL line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICϕ) as a count source.

For details on the timeout function, see **Section 19.13.1, Timeout Function**.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held low when the timeout function is enabled (RIICnFER.TMOE bit = 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held high when the timeout function is enabled (RIICnFER.TMOE bit = 1).

SDDL[2:0] Bits (SDA Output Delay Setup Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.

For details on this function, see **Section 19.7, Facility for Delaying SDA Output**.

CAUTION

Set the SDA output delay time to meet the I2C bus standard (within the data enable time/acknowledge enable time*1). Note that, if a value outside the standard is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

Note 1. Data enable time/acknowledge enable time
3,450 ns (up to 100 kbps: standard mode [Sm])
900 ns (up to 400 kbps: fast mode [Fm])

19.3.6 RIICnMR3 — I2C Bus Mode Register 3

Access: RIICnMR3 register can be read or written in 32-bit units.
RIICnMR3L and RIICnMR3H registers can be read or written in 16-bit units.
RIICnMR3LL, RIICnMR3LH, RIICnMR3HL, and RIICnMR3HH registers can be read or written in 8-bit units.

Address: RIICnMR3: <RIICn_base> + 0010_H

RIICnMR3L: <RIICn_base> + 0010_H, RIICnMR3H: <RIICn_base> + 0012_H

RIICnMR3LL: <RIICn_base> + 0010_H, RIICnMR3LH: <RIICn_base> + 0011_H,
RIICnMR3HL: <RIICn_base> + 0012_H, RIICnMR3HH: <RIICn_base> + 0013_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Table 19.14 RIICnMR3 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	WAIT*2	WAIT 0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading RIICnDRR.
5	RDRFS*2	RDRF Flag Set Timing Selection 0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCL line is not held low at the falling edge of the eighth clock cycle.) 1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle. (The SCL line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKBT bit.
4	ACKWP*1	ACKBT Write Protect 0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.
3	ACKBT*1	Transmit Acknowledge 0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 1 is sent as the acknowledge bit (NACK transmission).
2	ACKBR	Receive Acknowledge 0: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).
1, 0	NF[1:0]	Digital noise Filter Stage Selection b1 b0 0 0: Noise of up to one IICϕ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICϕ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICϕ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICϕ cycles is filtered out (4-stage filter).

Note 1. If it is attempted to write 1 to both ACKWP and ACKBT bits, the ACKBT bit cannot be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

NF[1:0] Bits (Digital Noise Filter Stage Selection)

These bits are used to select the number of stages of the digital noise filter.

CAUTION

Set the noise range to be filtered out by the noise filter within a range less than the SCL line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or low-level period, whichever is shorter) - [1.5 internal reference clock synchronized (IICϕ) cycles] or more, the SCL clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

ACKBR Bit (Receive Acknowledge)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

When 1 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledge timing 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 after ACKWP reading while the ACKWP bit is set to 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

CAUTION

The ACKBT bit must be written to while the ACKWP bit is 1. If the ACKBT bit is written to with the ACKWP bit cleared to 0, writing to the ACKBT bit is disabled.

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Selection)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL line low at the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCL line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCL line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL line is released by writing a value to the ACKBT bit.

After data is received with this setting, the SCL line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (RIICnDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. 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 SCL line is held low from the falling edge of the ninth clock cycle until the RIICnDRR value is read each time single-byte data is received. This enables receive operation in byte units.

CAUTION

When the value of the WAIT bit is cleared to 0, be sure to read the RIICnDRR beforehand.

19.3.7 RIICnFER — I2C Bus Function Enable Register

Access: RIICnFER register can be read or written in 32-bit units.
RIICnFERL and RIICnFERH registers can be read or written in 16-bit units.
RIICnFERLL, RIICnFERLH, RIICnFERHL, and RIICnFERHH registers can be read or written in 8-bit units.

Address: RIICnFER: <RIICn_base> + 0014_H

RIICnFERL: <RIICn_base> + 0014_H, RIICnFERH: <RIICn_base> + 0016_H

RIICnFERLL: <RIICn_base> + 0014_H, RIICnFERLH: <RIICn_base> + 0015_H,
RIICnFERHL: <RIICn_base> + 0016_H, RIICnFERHH: <RIICn_base> + 0017_H

Value after reset: 0000 0072_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.15 RIICnFER Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	SCLE	SCL Synchronous Circuit Enable 0: No SCL synchronous circuit is used. 1: An SCL synchronous circuit is used.
5	NFE	Digital Noise Filter Circuit Enable 0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.
4	NACKE	NACK Reception Transfer Suspension Enable 0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).
3	SALE	Slave Arbitration-Lost Detection Enable 0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.
1	MALE	Master Arbitration-Lost Detection Enable 0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the RIICnCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the RIICnCR2.MST and TRS bits automatically when arbitration is lost.) If 1 is written to the ST bit while the BBSY flag is 1 in slave transmit mode, the TRS bit is not cleared.
0	TMOE	Timeout Function Enable 0: The timeout function is disabled. 1: The timeout function is enabled.

TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, see **Section 19.13.1, Timeout Function**.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to select enabling or disabling of the arbitration-lost detection function. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether the detection of ACK during transmission of NACK in reception (such as when slaves with the same address are present on the bus and each is transmitting different data, or when two or more masters select the same slave device simultaneously with different numbers of bytes for reception) is judged to represent a loss in arbitration.

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify 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 (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify 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 is continued regardless of the received acknowledge content.

SCLE Bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with a rising or falling edge on the SCL line. Normally, set this bit to 1.

When the SCLE bit is cleared to 0 (SCL synchronous circuit not used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in RIICnBRH and RIICnBRL regardless of the SCL line state. For this reason, if the bus load of the I^2C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.

This bit must not be cleared to 0 except for checking the output of the transfer rate.

19.3.8 RIICnSER — I2C Bus Status Enable Register

Access: RIICnSER register can be read or written in 32-bit units.
RIICnSERL and RIICnSERH registers can be read or written in 16-bit units.
RIICnSERLL, RIICnSERLH, RIICnSERHL, and RIICnSERHH registers can be read or written in 8-bit units.

Address: RIICnSER: <RIICn_base> + 0018_H

RIICnSERL: <RIICn_base> + 0018_H, RIICnSERH: <RIICn_base> + $001A_H$

RIICnSERLL: <RIICn_base> + 0018_H, RIICnSERLH: <RIICn_base> + 0019_H,
RIICnSERHL: <RIICn_base> + $001A_H$, RIICnSERHH: <RIICn_base> + $001B_H$

Value after reset: $0000\ 0009_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	DIDE	—	GCE	SAR2	SAR1	SAR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 19.16 RIICnSER Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	DIDE	Device-ID Address Detection Enable 0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	GCE	General Call Address Enable 0: General call address detection is disabled. 1: General call address detection is enabled.
2	SAR2	Slave Address Register 2 Enable 0: Slave address in RIICnSAR2 is disabled. 1: Slave address in RIICnSAR2 is enabled.
1	SAR1	Slave Address Register 1 Enable 0: Slave address in RIICnSAR1 is disabled. 1: Slave address in RIICnSAR1 is enabled.
0	SAR0	Slave Address Register 0 Enable 0: Slave address in RIICnSAR0 is disabled. 1: Slave address in RIICnSAR0 is enabled.

SARy Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in RIICnSARy.

When this bit is set to 1, the slave address set in RIICnSARy is enabled and is compared with the received slave address.

When this bit is cleared to 0, the slave address set in RIICnSARy is disabled and is ignored even if it matches the received slave address.

GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address ($0000\ 000_B$ + 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 RIIC recognizes the received slave address as the general call address independently of the slave addresses set in RIICnSARy (y = 0 to 2) and performs data receive operation.

When this bit is cleared to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID ($1111\ 100_B$) is received in the first frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is cleared to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, see **Section 19.9.3, Device-ID Address Detection**.

19.3.9 RIICnIER — I2C Bus Interrupt Enable Register

Access: RIICnIER register can be read or written in 32-bit units.
RIICnIERL and RIICnIERH registers can be read or written in 16-bit units.
RIICnIERLL, RIICnIERLH, RIICnIERHL, and RIICnIERHH registers can be read or written in 8-bit units.

Address: RIICnIER: <RIICn_base> + $001C_H$

RIICnIERL: <RIICn_base> + $001C_H$, RIICnIERH: <RIICn_base> + $001E_H$

RIICnIERLL: <RIICn_base> + $001C_H$, RIICnIERLH: <RIICn_base> + $001D_H$,
RIICnIERHL: <RIICn_base> + $001E_H$, RIICnIERHH: <RIICn_base> + $001F_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.17 RIICnIER Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	TIE	Transmit Data Empty Interrupt Enable 0: Transmit data empty interrupt request (INTIICnTI) is disabled. 1: Transmit data empty interrupt request (INTIICnTI) is enabled.
6	TEIE	Transmit End Interrupt Enable 0: Transmit end interrupt request (INTIICnTEI) is disabled. 1: Transmit end interrupt request (INTIICnTEI) is enabled.
5	RIE	Receive Complete Interrupt Enable 0: Receive complete interrupt request (INTIICnRI) is disabled. 1: Receive complete interrupt request (INTIICnRI) is enabled.
4	NAKIE	NACK Reception Interrupt Enable 0: NACK reception interrupt request (NAKI) is disabled. 1: NACK reception interrupt request (NAKI) is enabled.
3	SPIE	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (SPI) is disabled. 1: Stop condition detection interrupt request (SPI) is enabled.
2	STIE	Start Condition Detection Interrupt Enable 0: Start condition detection interrupt request (STI) is disabled. 1: Start condition detection interrupt request (STI) is enabled.
1	ALIE	Arbitration-Lost Interrupt Enable 0: Arbitration-lost interrupt request (ALI) is disabled. 1: Arbitration-lost interrupt request (ALI) is enabled.
0	TMOIE	Timeout Interrupt Enable 0: Timeout interrupt request (TMOI) is disabled. 1: Timeout interrupt request (TMOI) is enabled.

TMOIE Bit (Timeout Interrupt Enable)

This bit is used to enable or disable timeout interrupt requests (TMOI) when the RIICnSR2.TMOF flag is set to 1. A TMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Enable)

This bit is used to enable or disable arbitration-lost interrupt requests (ALI) when the RIICnSR2.AL flag is set to 1. An ALI interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Enable)

This bit is used to enable or disable start condition detection interrupt requests (STI) when the RIICnSR2.START flag is set to 1. An STI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Enable)

This bit is used to enable or disable stop condition detection interrupt requests (SPI) when the RIICnSR2.STOP flag is set to 1. An SPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Enable)

This bit is used to enable or disable NACK reception interrupt requests (NAKI) when the RIICnSR2.NACKF flag is set to 1. An NAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Complete Interrupt Enable)

This bit is used to enable or disable receive complete interrupt requests (INTIICnRI) when the RIICnSR2.RDRF flag is set to 1.

TEIE Bit (Transmit End Interrupt Enable)

This bit is used to enable or disable transmit end interrupts (INTIICnTEI) when the RIICnSR2.TEND flag is set to 1. An INTIICnTEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit is used to enable or disable transmit data empty interrupts (INTIICnTI) when the RIICnSR2.TDRE flag is set to 1.

19.3.10 RIICnSR1 — I2C Bus Status Register 1

Access: RIICnSR1 register can be read or written in 32-bit units.
RIICnSR1L and RIICnSR1H registers can be read or written in 16-bit units.
RIICnSR1LL, RIICnSR1LH, RIICnSR1HL, and RIICnSR1HH registers can be read or written in 8-bit units.

Address: RIICnSR1: <RIICn_base> + 0020_H

RIICnSR1L: <RIICn_base> + 0020_H, RIICnSR1H: <RIICn_base> + 0022_H

RIICnSR1LL: <RIICn_base> + 0020_H, RIICnSR1LH: <RIICn_base> + 0021_H,
RIICnSR1HL: <RIICn_base> + 0022_H, RIICnSR1HH: <RIICn_base> + 0023_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R(/W) *1	R	R(/W) *1	R(/W) *1	R(/W) *1	R(/W) *1

Note 1. Only 0 can be written to this bit.

Table 19.18 RIICnSR1 Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	DID	Device-ID Address Detection Flag 0: Device-ID address is not detected. 1: Device-ID address is detected.
4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	GCA	General Call Address Detection Flag 0: General call address is not detected. 1: General call address is detected.
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 is not detected. 1: Slave address 2 is detected.
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 is not detected. 1: Slave address 1 is detected.
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 is not detected. 1: Slave address 0 is detected.

AASy Flag (Slave Address y Detection) (y = 0 to 2)

[Setting conditions]

<For 7-bit address format: RIICnSARy.FSy = 0>

When the received slave address matches the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

When the received slave address matches a value of (1111 0_B + RIICnSARy.SVA[9:8]) and the following address matches the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

<For 7-bit address format: RIICnSARy.FSy = 0>

- When the received slave address does not match the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

- When the received slave address does not match a value of (1111 0_B + RIICnSARy.SVA[9:8]) with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (1111 0_B + RIICnSARy.SVA[9:8]) and the following address does not match the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

GCA Flag (General Call Address Detection)

[Setting condition]

- When the received slave address matches the general call address (0000 000_B + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000_B + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection)

[Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID ($1111\ 100_B$) + 0 [W]) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled). This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.
- When a restart condition is detected after a match with the device ID address and the device ID address ($1111\ 100_B$) plus 1[R] has matched while the setting of the RIICnSER.DIDE bit is 1 (device ID address detection enabled). This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID ($1111\ 100_B$)) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID ($1111\ 100_B$) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

19.3.11 RIICnSR2 — I2C Bus Status Register 2

Access: RIICnSR2 register can be read or written in 32-bit units.
RIICnSR2L and RIICnSR2H registers can be read or written in 16-bit units.
RIICnSR2LL, RIICnSR2LH, RIICnSR2HL, and RIICnSR2HH registers can be read or written in 8-bit units.

Address: RIICnSR2: <RIICn_base> + 0024_H

RIICnSR2L: <RIICn_base> + 0024_H, RIICnSR2H: <RIICn_base> + 0026_H

RIICnSR2LL: <RIICn_base> + 0024_H, RIICnSR2LH: <RIICn_base> + 0025_H,
RIICnSR2HL: <RIICn_base> + 0026_H, RIICnSR2HH: <RIICn_base> + 0027_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1

Note 1. Only 0 can be written to this bit.

Table 19.19 RIICnSR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	TDRE	Transmit Data Empty Flag 0: RIICnDRT contains transmit data. 1: RIICnDRT contains no transmit data.
6	TEND	Transmit End Flag 0: Data is being transmitted. 1: Data has been transmitted.
5	RDRF	Receive Complete Flag 0: RIICnDRR contains no receive data. 1: RIICnDRR contains receive data.
4	NACKF	NACK Detection Flag 0: NACK is not detected. 1: NACK is detected.
3	STOP	Stop Condition Detection Flag 0: Stop condition is not detected. 1: Stop condition is detected.
2	START	Start Condition Detection Flag 0: Start condition is not detected. 1: Start condition is detected.
1	AL	Arbitration-Lost Flag 0: Arbitration is not lost. 1: Arbitration is lost.
0	TMOF	Timeout Detection Flag 0: Timeout is not detected. 1: Timeout is detected.

TMOF Flag (Timeout Detection)

This flag is set to 1 when the RIIC recognizes timeout after the SCL line state remains unchanged for a certain period.

[Setting condition]

The timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is stuck low or high during the following conditions:

- The bus is busy (the RIICnCR2.BBSY flag is 1) in master mode (the RIICnCR2.MST bit is 1).
- The RIIC's own slave address matches (the RIICnSR1 register is not 00_H) and the bus is busy (the RIICnCR2.BBSY bit is 1) in slave mode (the RIICnCR2.MST bit is 0).
- The bus is free (the RIICnCR2.BBSY flag is 0) while generation of a start condition is requested (the RIICnCR2.ST bit is 1).

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The RIIC monitors the level on the SDA line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL bit to 1 to indicate that the bus is occupied by another device.

The RIIC can also set the flag to indicate the detection of loss of arbitration during NACK transmission in receive mode or during data transmission in slave mode.

[Setting conditions]

<When master arbitration-lost detection is enabled: RIICnFER.MALE = 1>

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA line is driven low while the internal SDA output is at a high level (the SDA pin is in the high-impedance state))
- When a start condition is detected while the RIICnCR2.ST bit is 1 (start condition issuance request) or the internal SDA output state does not match the SDA line level
- When the RIICnCR2.ST bit is set to 1 (start condition issuance request) with the RIICnCR2.BBSY flag set to 1.

<When NACK arbitration-lost detection is enabled: RIICnFER.NALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode

<When slave arbitration-lost detection is enabled: RIICnFER.SALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL bit after reading AL = 1

- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 19.20 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions

RIICnFER			RIICnSR2		
MALE	NALE	SALE	AL	Error	Arbitration-Lost Generation Source
1	—	—	1	Start condition issuance error	When internal SDA output state does not match SDA line level when a start condition is detected while the RIICnCR2.ST bit is 1
					When RIICnCR2.ST is set to 1 with RIICnCR2.BBSY set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
—	1	—	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
—	—	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

—: Don't care

START Flag (Start Condition Detection)

[Setting condition]

When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection)

[Setting condition]

When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection)

[Setting condition]

When acknowledge is not received (NACK is received) from the receive device in transmit mode with the RIICnFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to RIICnDRT in transmit mode or reading from RIICnDRR in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, clear the NACKF flag to 0.

RDRF Flag (Receive Complete)

[Setting conditions]

- When receive data has been transferred from RIICnDRS to RIICnDRR
  At the rising edge of the eighth or ninth SCL clock cycle (selected by the RIICnMR3.RDRFS bit)
- When the received slave address matches the address enabled in RIICnSER after a start condition (or a restart condition) is detected with the RIICnCR2.TRS bit cleared to 0
- In master mode, transition to master reception while the R/W# bit appended to the slave address is set to 1

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from RIICnDRR
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

TEND Flag (Transmit End)

[Setting condition]

At the rising edge of the ninth SCL clock cycle while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to RIICnDRT
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty)

[Setting conditions]

- When data has been transferred from RIICnDRT to RIICnDRS and RIICnDRT becomes empty
- When the RIICnCR2.TRS bit is set to 1
  - When the RIICnCR2.MST bit is set to 1 after a start condition is detected
  - When 1 is written to the RIICnCR2.TRS bit while the RIICnMR1.MTWP bit is 1
- When the received slave address matches the address enabled in RIICnSER after a start condition including a restart condition is detected with the RIICnCR2.TRS bit set to 1

[Clearing conditions]

- When data is written to RIICnDRT
- When the RIICnCR2.TRS bit is cleared to 0
  - When a stop condition is detected
  - When the RIIC enters receive mode from transmit mode
  - When 0 is written to the RIICnCR2.TRS bit while the RIICnMR1.MTWP bit is 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

When the NACKF flag is set to 1 while the RIICnFER.NACKE bit is 1, the RIIC suspends data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the RIICnDRS register and the RIICnDRT register becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

19.3.12 RIICnSARy — I2C Slave Address Register y (y = 0 to 2)

Access: RIICnSARy register can be read or written in 32-bit units.
RIICnSARyL and RIICnSARyH registers can be read or written in 16-bit units.
RIICnSARyLL, RIICnSARyLH, RIICnSARyHL, and RIICnSARyHH registers can be read or written in 8-bit units.

Address: RIICnSAR0: <RIICn_base> + 0028_H

RIICnSAR0L: <RIICn_base> + 0028_H, RIICnSAR0H: <RIICn_base> + $002A_H$

RIICnSAR0LL: <RIICn_base> + 0028_H, RIICnSAR0LH: <RIICn_base> + 0029_H,
RIICnSAR0HL: <RIICn_base> + $002A_H$, RIICnSAR0HH: <RIICn_base> + $002B_H$

RIICnSAR1: <RIICn_base> + $002C_H$

RIICnSAR1L: <RIICn_base> + $002C_H$, RIICnSAR1H: <RIICn_base> + $002E_H$

RIICnSAR1LL: <RIICn_base> + $002C_H$, RIICnSAR1LH: <RIICn_base> + $002D_H$,
RIICnSAR1HL: <RIICn_base> + $002E_H$, RIICnSAR1HH: <RIICn_base> + $002F_H$

RIICnSAR2: <RIICn_base> + 0030_H

RIICnSAR2L: <RIICn_base> + 0030_H, RIICnSAR2H: <RIICn_base> + 0032_H

RIICnSAR2LL: <RIICn_base> + 0030_H, RIICnSAR2LH: <RIICn_base> + 0031_H,
RIICnSAR2HL: <RIICn_base> + 0032_H, RIICnSAR2HH: <RIICn_base> + 0033_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FSy	—	—	—	—	—	SVA[9:1]									SVA0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.21 RIICnSARy Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	FSy	7-Bit/10-Bit Address Format Selection 0: The 7-bit address format is selected. 1: The 10-bit address format is selected.
14 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 1	SVA[9:1]	7-Bit Address/10-Bit Address Upper Bits A slave address is set. • When the FSy bit is 0 (7-bit address format), the SVA[7:1] bits are valid and form a 7-bit slave address. • When the FSy bit is 1 (10-bit address format), SVA[9:1] bits form a 10-bit slave address (combined with the SVA0 bit).
0	SVA0	10-Bit Address LSB The least significant bit (LSB) of a 10-bit slave address is set. • When the FSy bit is 0 (7-bit address format), this bit is invalid. • When the FSy bit is 1 (10-bit address format), this bit is a 10-bit slave address (combined with the SVA[9:1] bits).

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (RIICnSARy.FSy = 1), this bit functions as the LSB of a 10-bit address and forms a 10-bit address in combination with the SVA[9:1] bits.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, this bit is valid. While the RIICnSARy.FSy bit or SARy bit is 0, the setting of this bit is ignored.

SVA[9:1] Bits (7-Bit Address/10-Bit Address Upper Bits)

When the 7-bit address format is selected (RIICnSARy.FSy = 0), these bits function as a 7-bit address. When the 10-bit address format is selected (RIICnSARy.FSy = 1), these bits function as a 10-bit address in combination with the SVA0 bit.

While the RIICnSER.SARy bit is 0, the setting of these bits is ignored.

FSy Bit (7-Bit/10-Bit Address Format Selection)

This bit is used to select 7-bit address or 10-bit address for slave address y (in RIICnSARy).

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 0, the 7-bit address format is selected for slave address y, the RIICnSARy.SVA[7:1] setting is valid, and the settings of the SVA[9:8] bits and the RIICnSARy.SVA0 bit are ignored.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[9:1] bits and the SVA0 bit are valid.

While the RIICnSER.SARy bit is 0 (RIICnSARy disabled), the setting of the RIICnSARy.FSy bit is invalid.

19.3.13 RIICnBRL — I2C Bus Bit Rate Low-Level Register

Access: RIICnBRL register can be read or written in 32-bit units.
RIICnBRLL and RIICnBRLH registers can be read or written in 16-bit units.
RIICnBRLLL, RIICnBRLLH, RIICnBRLHL, and RIICnBRLHH registers can be read or written in 8-bit units.

Address: RIICnBRL: <RIICn_base> + 0034_H

RIICnBRLL: <RIICn_base> + 0034_H, RIICnBRLH: <RIICn_base> + 0036_H

RIICnBRLLL: <RIICn_base> + 0034_H, RIICnBRLLH: <RIICn_base> + 0035_H,
RIICnBRLHL: <RIICn_base> + 0036_H, RIICnBRLHH: <RIICn_base> + 0037_H

Value after reset: 0000 $00FF_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	BRL[4:0]				
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 19.22 RIICnBRL Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	BRL[4:0]	Bit Rate Low-Level Period Low-level period of SCL clock

The RIICnBRL register is a 5-bit register that is used to set the width at low level for the SCL clock.

It also works to generate the data setup time for automatic SCL low-hold operation (see **Section 19.10, Automatic Low-Hold Function for SCL**); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*1.

RIICnBRL counts the low-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: standard mode [Sm])
100 ns (up to 400 kbps: fast mode [Fm])

19.3.14 RIICnBRH — I2C Bus Bit Rate High-Level Register

Access: RIICnBRH register can be read or written in 32-bit units.
RIICnBRHL and RIICnBRHH registers can be read or written in 16-bit units.
RIICnBRHLL, RIICnBRHLH, RIICnBRHHL, and RIICnBRHHH registers can be read or written in 8-bit units.

Address: RIICnBRH: <RIICn_base> + 0038_H

RIICnBRHL: <RIICn_base> + 0038_H, RIICnBRHH: <RIICn_base> + $003A_H$

RIICnBRHLL: <RIICn_base> + 0038_H, RIICnBRHLH: <RIICn_base> + 0039_H,
RIICnBRHHL: <RIICn_base> + $003A_H$, RIICnBRHHH:<RIICn_base> + $003B_H$

Value after reset: 0000 $00FF_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	BRH[4:0]				
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 19.23 RIICnBRH Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	BRH[4:0]	Bit Rate High-Level Period High-level period of SCL clock

RIICnBRH is a 5-bit register to set the high-level period of SCL clock. RIICnICBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high-level period.

RIICnBRH counts the high-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits.

The I2C transfer rate and the SCL clock duty are calculated using the following expression.

(1) When RIICnFER.SCLE = 0
Transfer rate = 1 / {[(RIICnBRH+1) + (RIICnBRL+1)] / IICφ*1 + tr + tf}
Duty cycle = {tr+[(RIICnBRH+1)/ IICφ]}/{ tr + tf + [(RIICnBRH+1) + (RIICnBRL+1)]/IICφ}

(2) When RIICnFER.SCLE=1, RIICnFER.NFE=0, IICφ = PCLK
Transfer rate = 1 / {[(RIICnBRH+3) + (RIICnBRL+3)] / IICφ*1 + tr + tf}
Duty cycle = {tr +[(RIICnBRH+3)/ IICφ]}/{ tr + tf + [(RIICnBRH+3) + (RIICnBRL+3)]/IICφ}

(3) When RIICnFER.SCLE=1, RIICnFER.NFE=1, IICφ = PCLK
Transfer rate =1 / {[(RIICnBRH+3+nf) + (RIICnBRL+3+nf)] / IICφ*1 + tr + tf}
Duty cycle = {tr +[(RIICnBRH+3+nf)/ IICφ]}/{ tr + tf + [(RIICnBRH+3+nf) + (RIICnBRL+3+nf)]/ IICφ}

(4) RIICnFER.SCLE=1, RIICnFER.NFE=0, IICφ < PCLK
Transfer rate =1 / {[(RIICnBRH+2) + (RIICnBRL+2)] / IICφ*1+ tr + tf}
Duty cycle = {tr +[(RIICnBRH+2)/ IICφ]}/{ tr + tf + [(RIICnBRH+2) + (RIICnBRL+2)]/IICφ}

(5) When RIICnFER.SCLE=1, RIICnFER.NFE=1, IICφ < PCLK
Transfer rate =1 / {[(RIICnBRH+2+nf) + (RIICnBRL+2+nf)] / IICφ*1+ tr + tf}
Duty cycle = {tr +[(RIICnBRH+2+nf)/ IICφ]}/{ tr + tf + [(RIICnBRH+2+nf) + (RIICnBRL+2+nf)]/ IICφ}

tf: SCL line falling time [ns]*2
tr: SCL line rising time [ns]*2
nf: Digital noise filter stage
Duty cycle: 0% < Duty < 100%

Note 1. As for IICφ, see CKS[2:0] in **Section 19.3.4, RIICnMR1 — I2C Bus Mode Register 1**.

Note 2. The SCL line rising time [tr] and SCL line falling 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.

Table 19.24 lists examples of the RIICnBRH and RIICnBRL register settings when the SCL synchronization circuit is not used.

Table 19.24 Examples of RIICnBRH/RIICnBRLSettings for Transfer Rate

Transfer Rate (kbps)	PCLK Frequency (MHz)														
	8					10					12.5				
		RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL	
	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	100_B	22	$F6_H$	25	$F9_H$	101_B	13	ED_H	15	EF_H	101_B	16	$F0_H$	20	$F4_H$
50	010_B	16	$F0_H$	19	$F3_H$	010_B	21	$F5_H$	24	$F8_H$	011_B	12	EC_H	15	EF_H
100	001_B	15	EF_H	18	$F2_H$	001_B	19	$F3_H$	23	$F7_H$	001_B	24	$F8_H$	29	FD_H
400	000_B	4	$E4_H$	10	EA_H	000_B	5	$E5_H$	12	EC_H	000_B	7	$E7_H$	16	$F0_H$

Transfer Rate (kbps)	PCLK Frequency (MHz)														
	16					20					25				
		RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL	
	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	101_B	22	$F6_H$	25	$F9_H$	110_B	13	ED_H	15	EF_H	110_B	16	$F0_H$	20	$F4_H$
50	011_B	16	$F0_H$	19	$F3_H$	011_B	21	$F5_H$	24	$F8_H$	100_B	12	EC_H	15	EF_H
100	010_B	15	EF_H	18	$F2_H$	010_B	19	$F3_H$	23	$F7_H$	010_B	24	$F8_H$	29	FD_H
400	000_B	9	$E9_H$	20	$F4_H$	000_B	11	EB_H	25	$F9_H$	001_B	7	$E7_H$	16	$F0_H$

Transfer Rate (kbps)	PCLK Frequency (MHz)									
	30					33				
		RIICnBRH		RIICnBRL			RIICnBRH		RIICnBRL	
	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]	CKS [2:0]	SCL Clock Width (high level) [IICφ]	Setting Value [write value]	SCL Clock Width (low level) [IICφ]	Setting Value [write value]
10	110_B	20	$F4_H$	24	$F8_H$	110_B	22	$F6_H$	26	FA_H
50	100_B	15	EF_H	18	$F2_H$	100_B	17	$F1_H$	20	$F4_H$
100	011_B	14	EE_H	17	$F1_H$	011_B	16	$F0_H$	19	$F3_H$
400	001_B	8	$E8_H$	19	$F3_H$	001_B	9	$E9_H$	21	$F5_H$

CAUTION

CBRH/ICBRL settings in these tables are calculated using the following values:
SCL line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns
SCL line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns
For the specified values of SCL line rising time (tr) and SCL line falling time (tf), see the I2C bus standard from NXP Semiconductors.

19.3.15 RIICnDRT — I2C Bus Transmit Data Register

Access: RIICnDRT register can be read or written in 32-bit units.
RIICnDRTL and RIICnDRTH registers can be read or written in 16-bit units.
RIICnDRTLL, RIICnDRTLH, RIICnDRTHL, and RIICnDRTHH registers can be read or written in 8-bit units.

Address: RIICnDRT: <RIICn_base> + $003C_H$

RIICnDRTL: <RIICn_base> + $003C_H$, RIICnDRTH: <RIICn_base> + $003E_H$

RIICnDRTLL: <RIICn_base> + $003C_H$, RIICnDRTLH: <RIICn_base> + $003D_H$,
RIICnDRTHL: <RIICn_base> + $003E_H$, RIICnDRTHH: <RIICn_base> + $003F_H$

Value after reset: 0000 $00FF_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When RIICnDRT detects a space in the I2C bus shift register (RIICnDRS), it transfers the transmit data that has been written to RIICnDRT to RIICnDRS and starts transmitting data in transmit mode.

The double-buffer structure of RIICnDRT and RIICnDRS allows continuous transmit operation if the next transmit data has been written to RIICnDRT while the RIICnDRS data is being transmitted.

RIICnDRT can always be read and written. Write transmit data to RIICnDRT once when a transmit data empty interrupt (INTIICnTI) request is generated. When writing to bit 31 to 8, write the value after reset.

19.3.16 RIICnDRR — I2C Bus Receive Data Register

Access: RIICnDRR register is a read-only register that can be read in 32-bit units.
RIICnDRRL and RIICnDRRH registers are the read-only registers that can be read in 16-bit units.
RIICnDRRLL, RIICnDRRLH, RIICnDRRHL, and RIICnDRRHH registers are the read-only registers that can be read in 8-bit units.

Address: RIICnDRR: <RIICn_base> + 0040_H

RIICnDRRL: <RIICn_base> + 0040_H, RIICnDRRH: <RIICn_base> + 0042_H

RIICnDRRLL: <RIICn_base> + 0040_H, RIICnDRRLH: <RIICn_base> + 0041_H,
RIICnDRRHL: <RIICn_base> + 0042_H, RIICnDRRHH: <RIICn_base> + 0043_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When 1 byte of data has been received, the received data is transferred from the I2C bus shift register (RIICnDRS) to RIICnDRR to enable the next data to be received.

The double-buffer structure of RIICnDRS and RIICnDRR allows continuous receive operation if the received data has been read from RIICnDRR while RIICnDRS is receiving data.

RIICnDRR cannot be written. Read data from RIICnDRR once when a receive complete interrupt (INTIICnRI) request is generated.

If RIIC receives the next receive data before the current data is read from RIICnDRR (while the RIICnSR2.RDRF flag is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

19.3.17 RIICnDRS — I2C Bus Shift Register

Access: This register is not accessible.

Address: —

Value after reset: 0000 00FF$_H$

Bit	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
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRS[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

RIICnDRS is a shift register to transmit and receive data.

During transmission, transmit data is transferred from RIICnDRT to RIICnDRS and is sent from the SDA pin. During reception, data is transferred from RIICnDRS to RIICnDRR after 1 byte of data has been received.

RIICnDRS cannot be accessed directly.

19.4 Interrupt Sources

The RIIC issues four types of interrupt request: transfer error or event generation (arbitration-lost detection, NACK detection, timeout detection, start condition detection, and stop condition detection), receive end, transmit data empty, and transmit end.

Table 19.25 lists details of the several interrupt requests. The receive complete and transmit data empty interrupt request are both capable of launching data transfer by the DMAC.

Table 19.25 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DMACA Launching	Interrupt Condition
INTIICnTI	Transmit Data Empty	TDRE	Possible	TDRE = 1 and TIE = 1
INTIICnTEI	Transmit End	TEND	Not possible	TEND = 1 and TEIE = 1
INTIICnRI	Receive End	RDRF	Possible	RDRF = 1 and RIE = 1
INTIICnEE	Transfer Error/ Event Generation	AL	Not possible	AL = 1 and ALIE = 1
		NACKF		NACKF = 1 and NAKIE = 1
		TMOF		TMOF = 1 and TMOIE = 1
		START		START = 1 and STIE = 1
		STOP		STOP = 1 and SPIE = 1

Clear or mask the each flag during interrupt handling.

CAUTIONS

1. **There is a latency (delay) between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt processing. Returning from interrupt processing without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.**
2. **Since INTIICnTI is an edge-detected interrupt, it does not require clearing. Furthermore, the RIICnSR2.TDRE flag (a condition for INTIICnTI) is automatically cleared to 0 when data for transmission are written to RIICnDRT or a stop condition is detected (RIICnSR2.STOP flag = 1).**
3. **Since INTIICnRI is an edge-detected interrupt, it does not require clearing. Furthermore, the RIICnSR2.RDRF flag (a condition for INTIICnRI) is automatically cleared to 0 when data are read from RIICnDRR.**
4. **When using the INTIICnTEI interrupt, clear the RIICnSR2.TEND flag in the INTIICnTEI interrupt processing.**
   Note that the RIICnSR2.TEND flag is automatically cleared to 0 when data for transmission are written to RIICnDRT or a stop condition is detected (RIICnSR2.STOP flag = 1).

19.5 Operation

19.5.1 Communication Data Format

The I2C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start condition or restart condition is an address frame used to specify 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 19.3 shows the I2C bus format, and **Figure 19.4** shows the I2C bus timing.

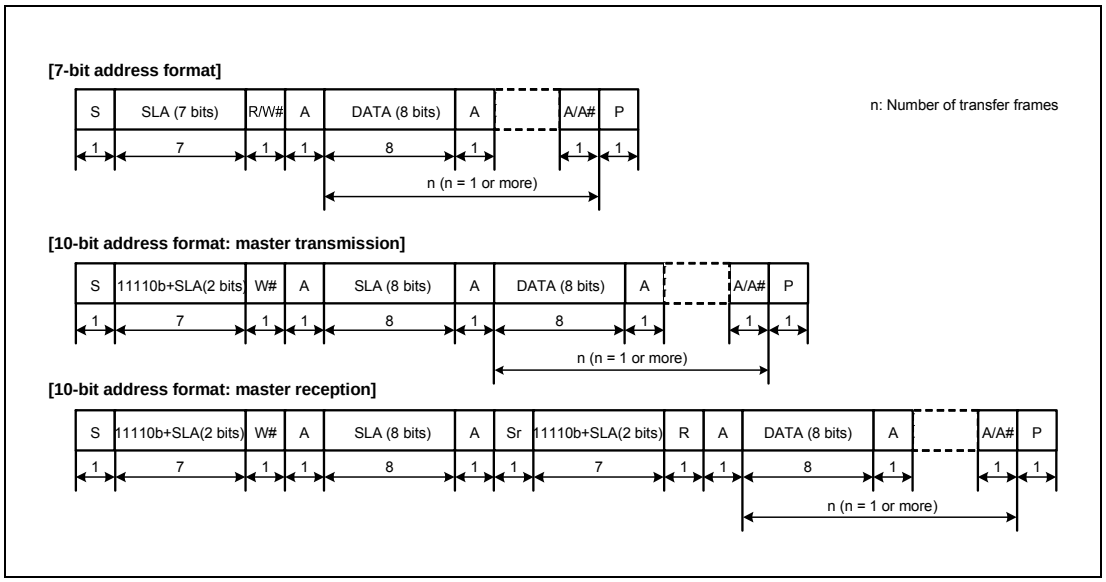


Figure 19.3 I2C Bus Format

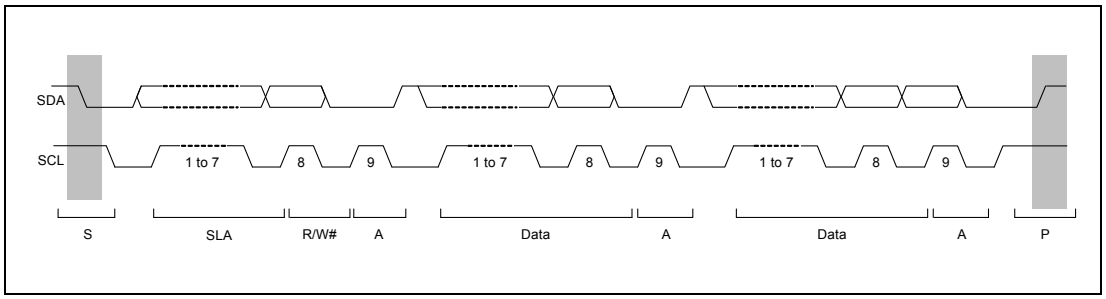


Figure 19.4 I2C Bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDA line low from high level while the SCL line is at a high level.

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 SDA line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not-acknowledge. The receiving device has not returned a response or is not present so the SDA line has remained at the high level.

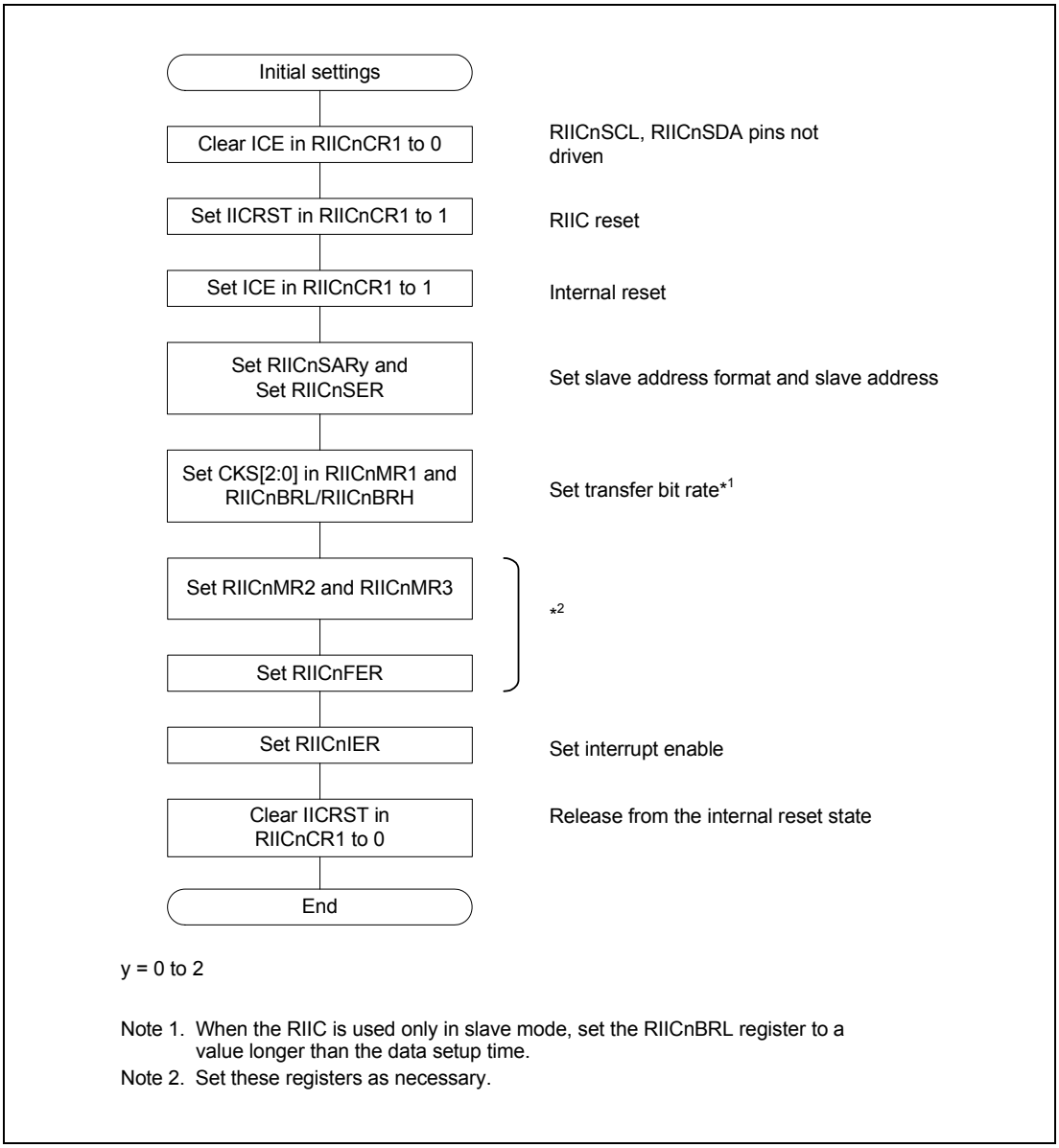
Sr: Restart condition. The master device drives the SDA line low from the high level after the setup time has elapsed with the SCL line at the high level.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA line high from low level while the SCL line is at a high level.

19.5.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in **Figure 19.5**. Make initial settings for the RIIC once when starting the RIIC.



Note 1. When the RIIC is used only in slave mode, set the RIICnBRL register to a value longer than the data setup time.

Note 2. Set these registers as necessary.

Figure 19.5 Example of RIIC Initialization Flowchart

19.5.3 Master Transmit Operation

In master transmit operation, the RIIC outputs the SCL (clock) and transmitted data signals as the master device, and the slave device returns acknowledgements. **Figure 19.6** shows an example of usage of master transmission and **Figure 19.7** to **Figure 19.9** show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

(1) Set the RIICnCR1.IICRST bit 1 to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 19.5**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared 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 SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS and MST bits to 1.

(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the slave address including the R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Since the RIICnSR2.NACKF flag being 1 at this time indicates that the slave address has not been recognized or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.
For data transmission with an address in the 10-bit format, start by writing $1111\ 0_B$, the two higher-order bits of the slave address, and W# to RIICnDRT as the first address transmission. Then, as the second address transmission, write the eight lower-order bits of the slave address to RIICnDRT.

(4) After confirming that the RIICnSR2.TDRE flag is 1, write the data for transmission to the RIICnDRT register. The RIIC automatically holds the SCL line low until the data for transmission are ready or a stop condition is issued.

(5) After all bytes of data for transmission have been written to the RIICnDRT register, wait until the value of the RIICnSR2.TEND flag returns to 1, and then set the RIICnCR2.SP bit to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.

(6) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00_B and enters slave receive mode. Furthermore, it automatically clears the RIICnSR2.TDRE and TEND flags to 0, and sets the RIICnSR2.STOP flag in to 1.

(7) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

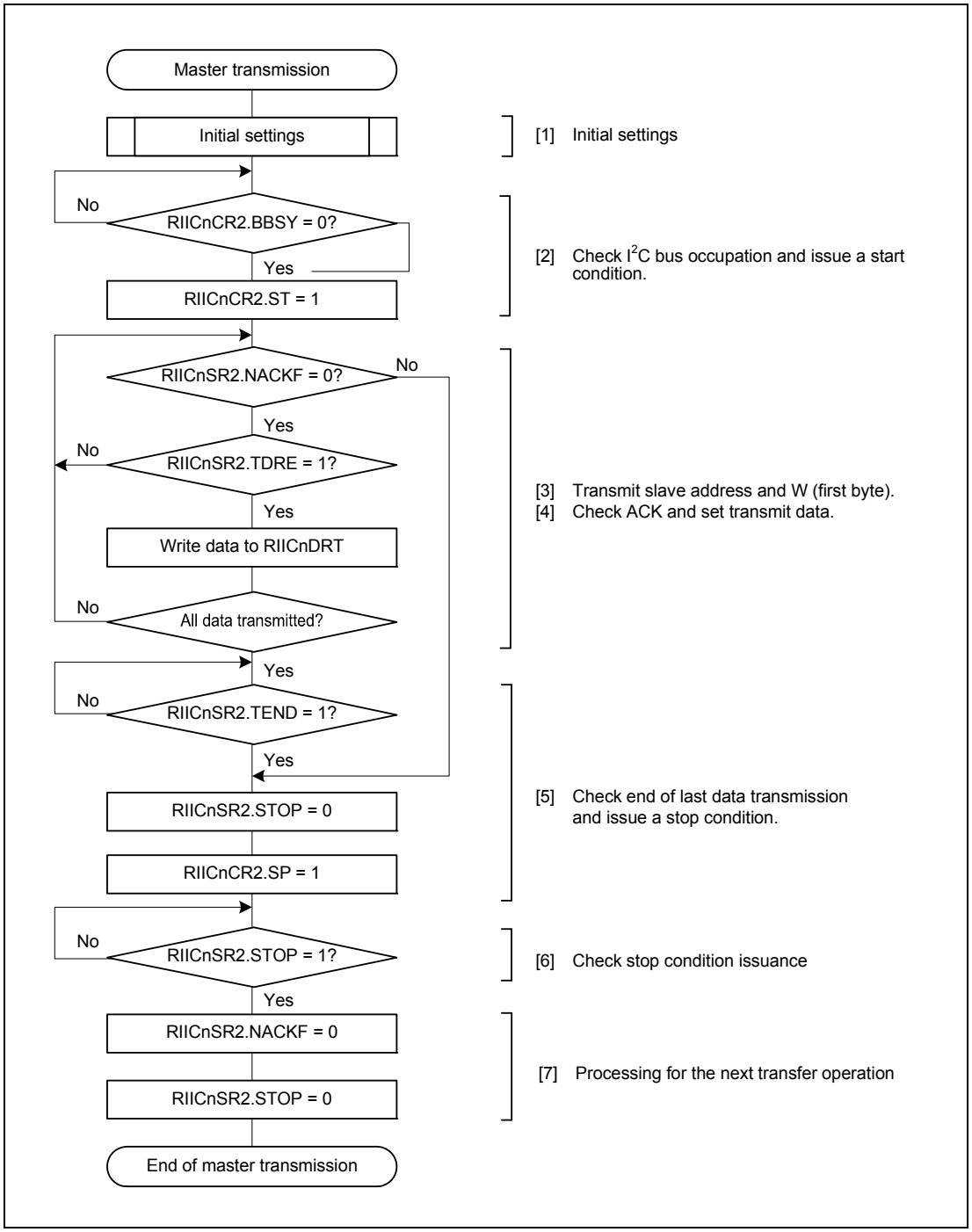


Figure 19.6 Example of Master Transmission Flowchart

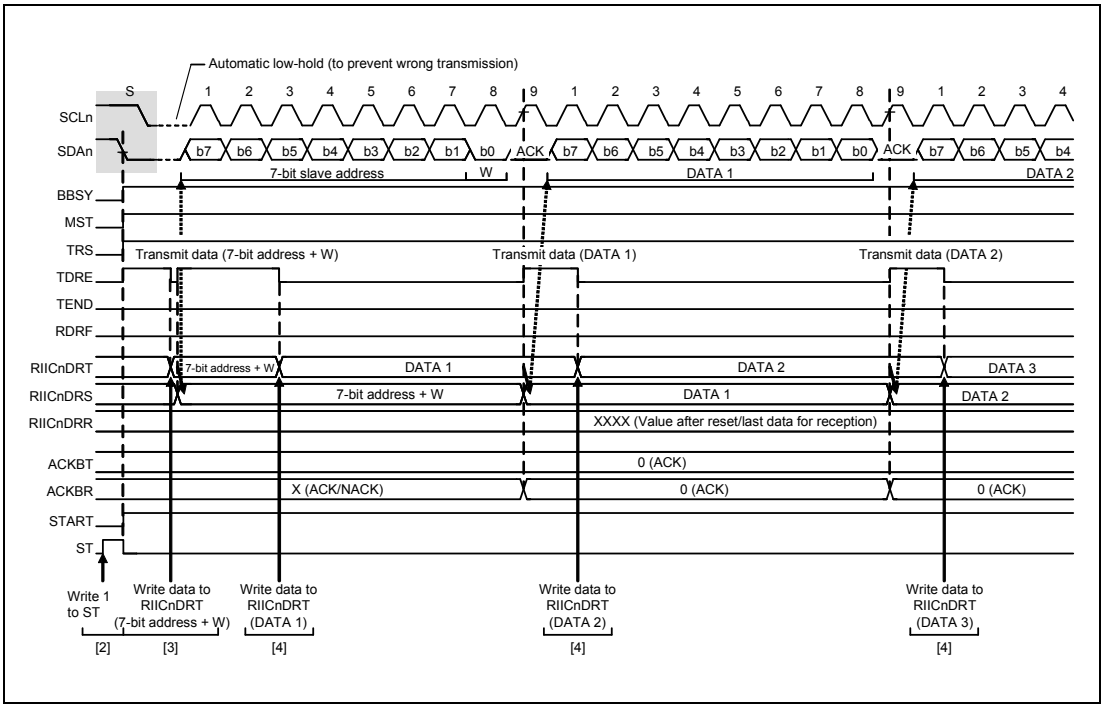


Figure 19.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

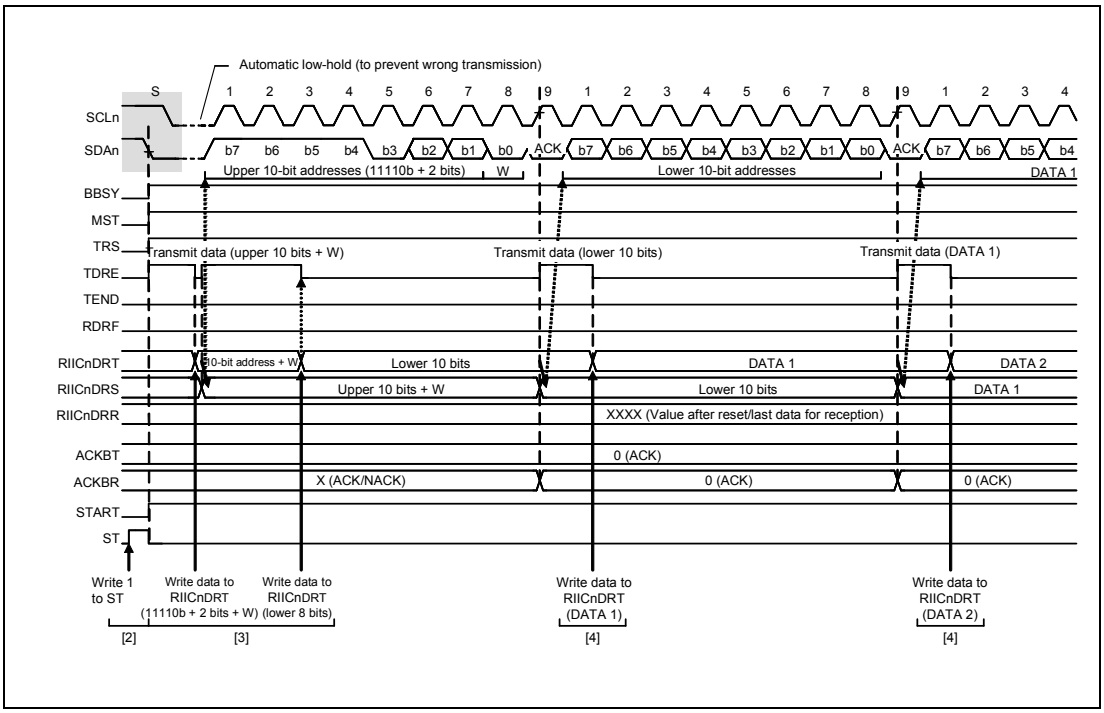


Figure 19.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

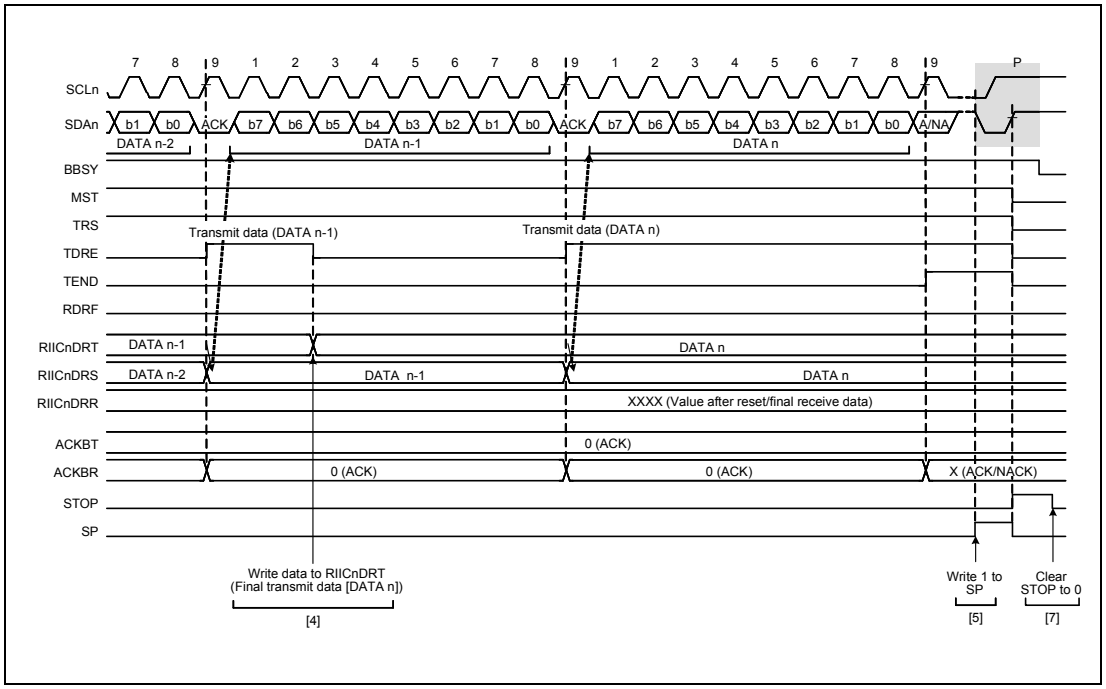


Figure 19.9 Master Transmit Operation Timing (3)

19.5.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL (clock) signal, receives data from the slave device, and returns acknowledgements. Since the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 19.10 shows an example of master reception flowchart (7-bit address format, 1 or 2 bytes), **Figure 19.11** shows an example of master reception flowchart (7-bit address format, 3 bytes or more), and **Figure 19.12** to **Figure 19.14** show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

(1) Set the RIICnCR1.IICRST bit to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 19.5**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and value of the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the slave address including the R/W# bit has been transmitted, the value of the RIICnCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the RIICnCR2.TRS bit is cleared to 0 on the rising edge of the ninth cycle of SCL (the clock signal), placing the RIIC in master receive mode. At this time, the TDRE flag is automatically cleared to 0 and the RIICnSR2.RDRF flag is automatically set to 1.
Since the RIICnSR2.NACKF flag being 1 at this time indicates that the slave address has not been recognized or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to transmit the two higher-order bits of the slave address and then the eight lower-order bits of the slave address, and issue a restart condition following generation of the transmission end interrupt (or after TEND = 1) (For details about the operation timing, see **Figure 19.13**). After that, transmitting 1111 0_B plus the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read RIICnDRR after confirming that the RIICnSR2.RDRF flag is 1; this makes the RIIC start output of the SCL (clock) signal and start data reception.

(5) After 1 byte of data has been received, the RIICnSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RIICnMR3.RDRFS bit. Reading out RIICnDRR at this time will produce the received data, and the RDRF flag is automatically cleared to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the RIICnMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the RIICnMR3.WAIT bit to 1 (for wait insertion) before reading the RIICnDRR (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the RIICnMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a stop condition is possible.

(6) When the RIICnMR3.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 (final) byte, set the RIICnMR3.ACKBT bit to 1 (NACK).

(7) After reading out the byte before last from the RIICnDRR register, if the value of the RIICnSR2.RDRF flag is confirmed to be 1, write 1 to the RIICnCR2.SP bit (stop condition issuance request) and then read the last byte from RIICnDRR. When RIICnDRR is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCL line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00_B and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the RIICnSR2.STOP flag to 1.

(9) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

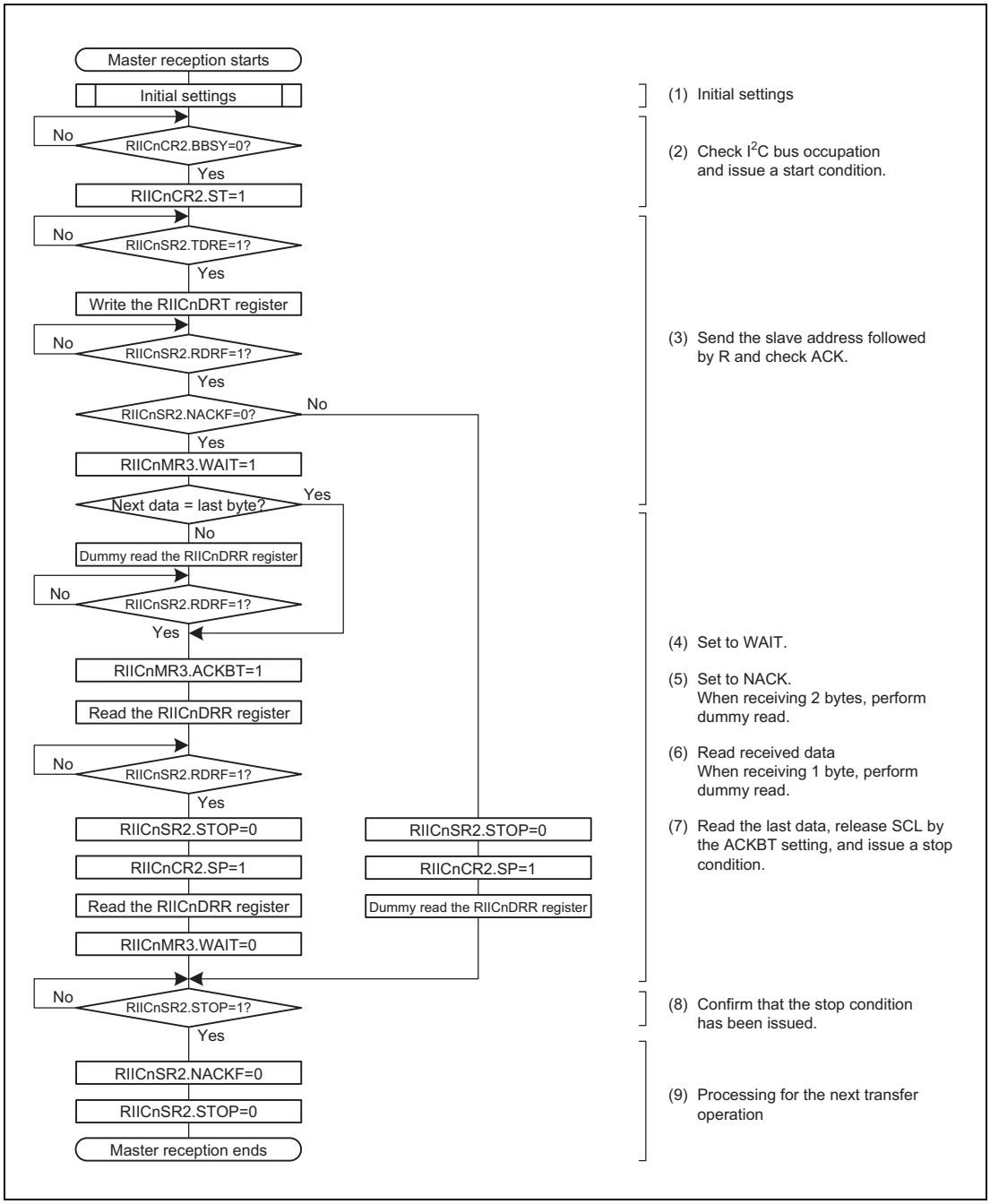


Figure 19.10 Example of Master Reception Flowchart (7-Bit Address Format, 1 or 2 Bytes)

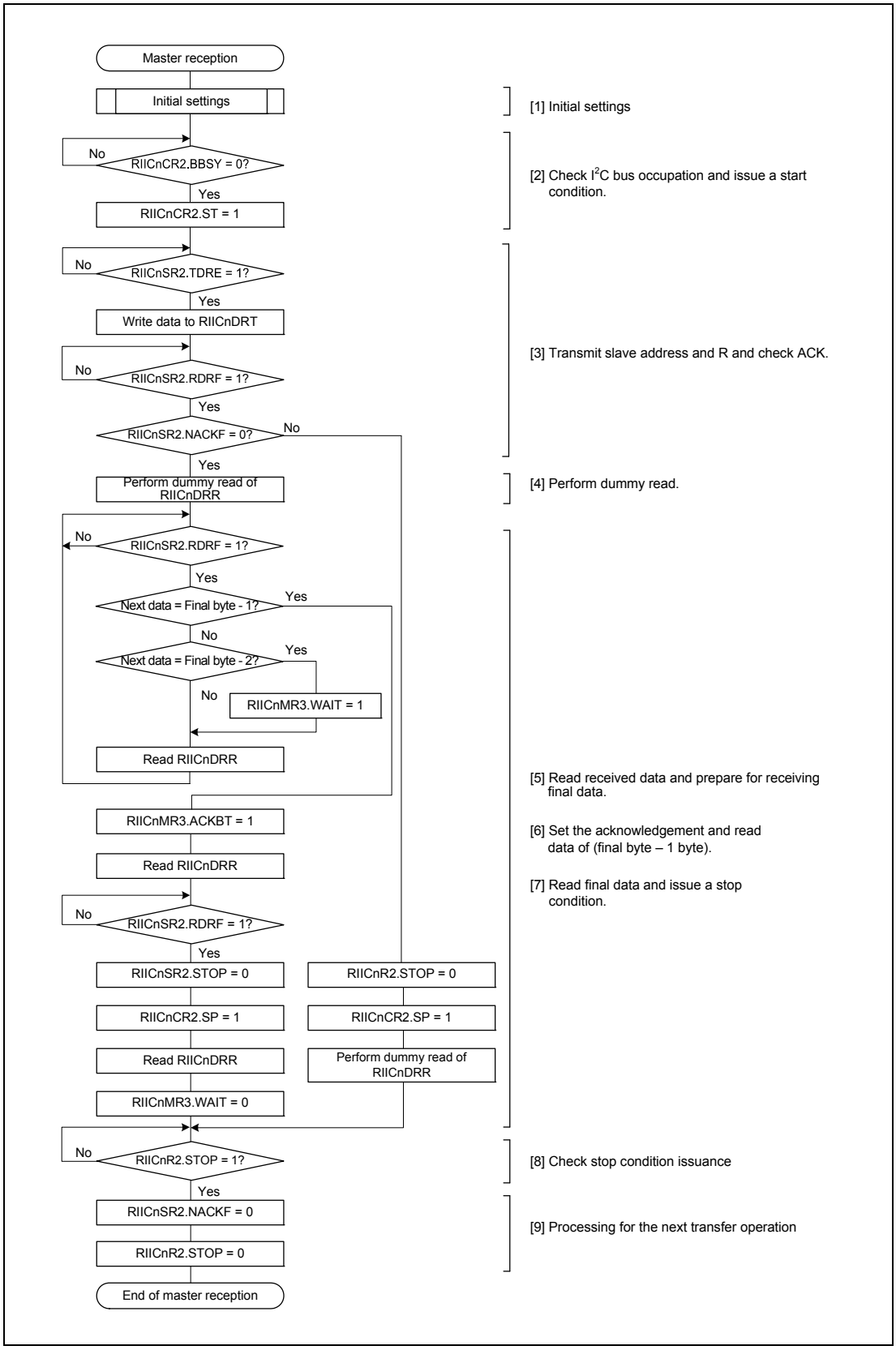


Figure 19.11 Example of Master Reception Flowchart (7-Bit Address Format, 3 Bytes or More)

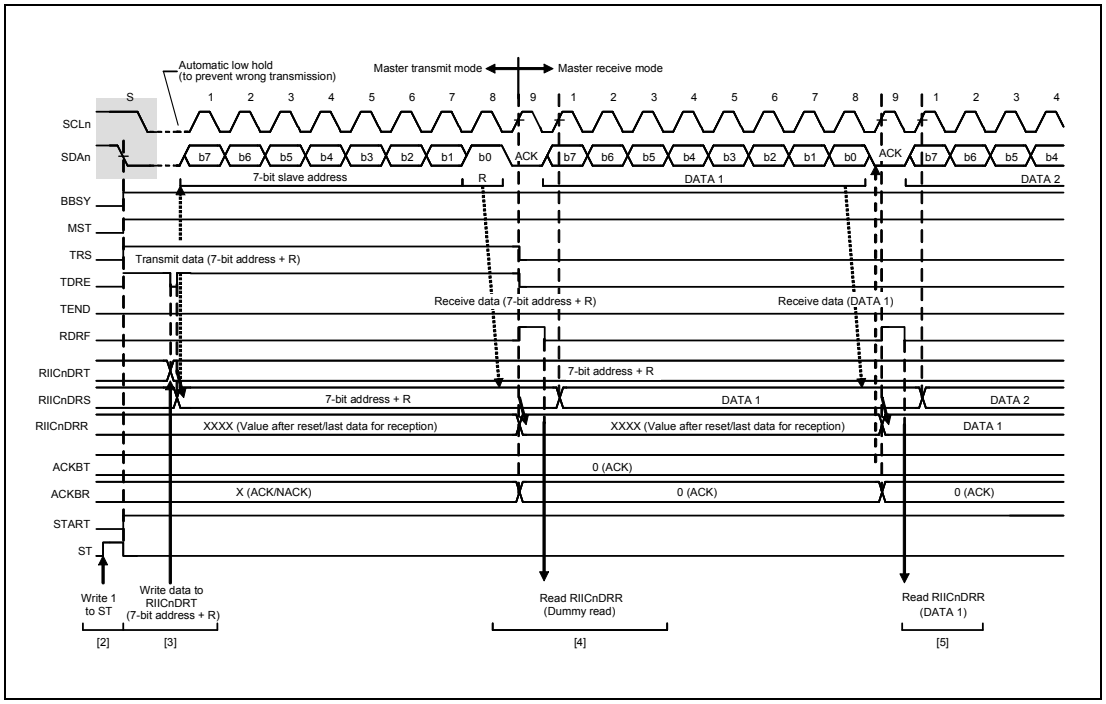


Figure 19.12 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

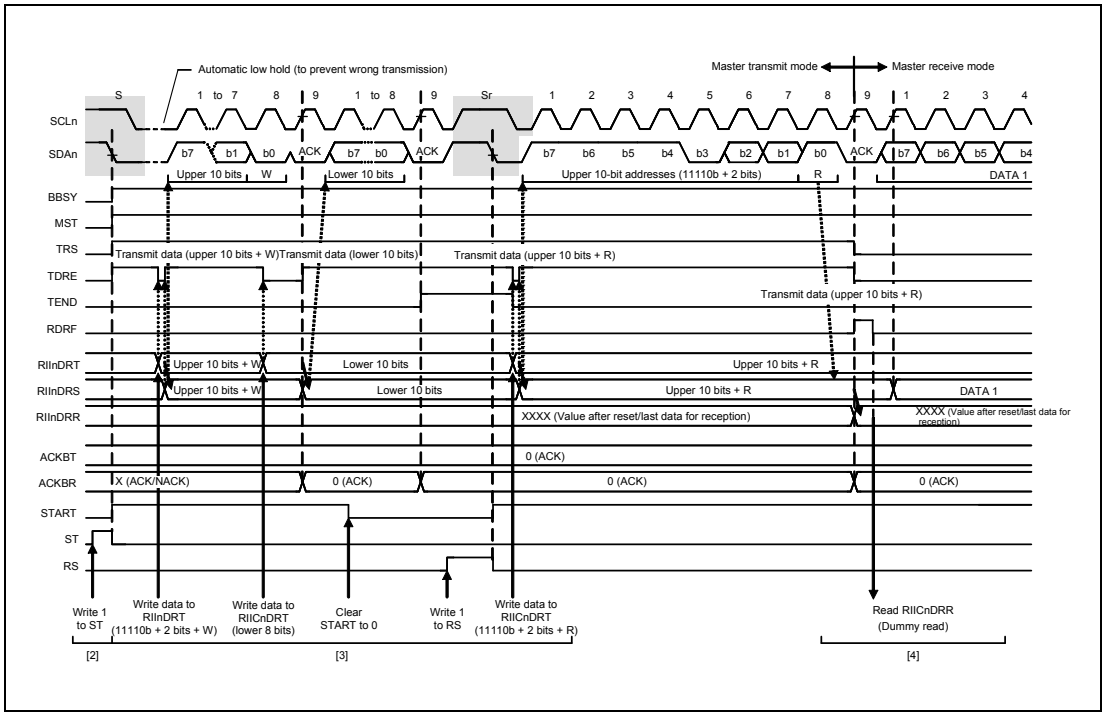


Figure 19.13 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)

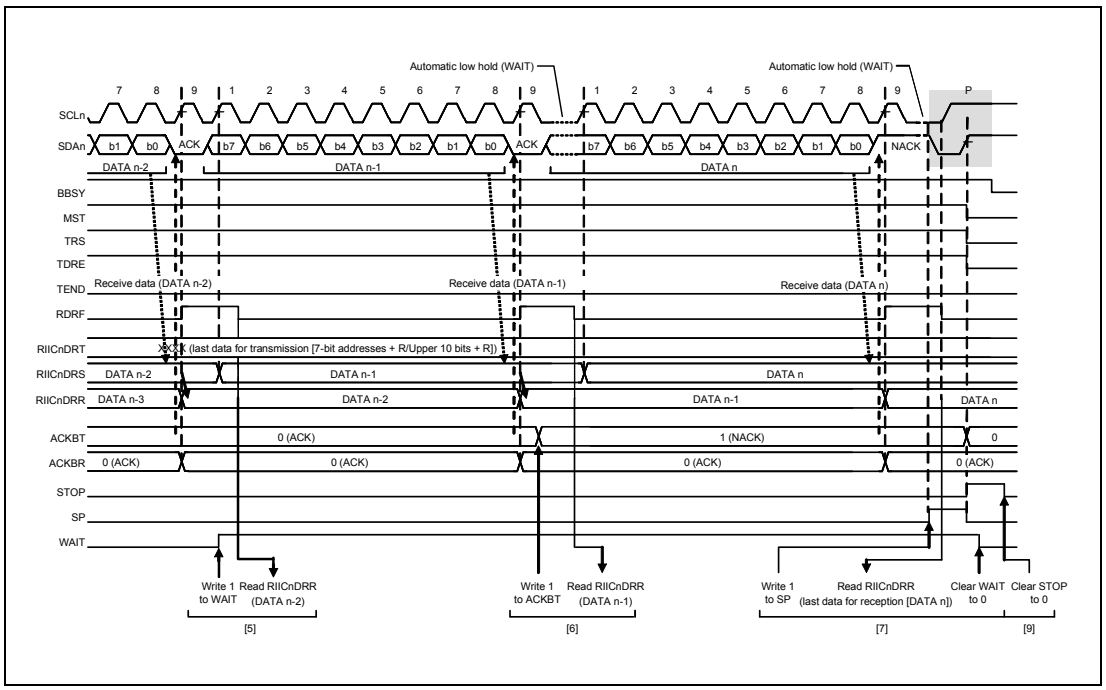


Figure 19.14 Master Receive Operation Timing (3) (when RDRFS = 0)

19.5.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL (clock) signal, the RIIC transmits data as a slave device, and the master device returns acknowledgements.

Figure 19.5 shows an example of usage of slave transmission and **Figure 19.16** and **Figure 19.17** show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Follow the procedure in **Figure 19.5** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC automatically sets the BBSY and RIICnSR2.START flags to 1 and automatically clears the ST bit to 0 on detection of a start condition.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the RIICnCR2.TRS bit and the RIICnSR2.TDRE flag to 1.

(3) After the RIICnSR2.TDRE flag is confirmed to be 1, write the data for transmission to the RIICnDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives an NACK signal) while the RIICnFER.NACKE bit is 1, the RIIC suspends transfer of the next data.

(4) Wait until the RIICnSR2.TEND flag is set to 1 while the RIICnSR2.TDRE flag is 1, after the RIICnSR2.NACKF flag is set to 1 or the last byte for transmission is written to the RIICnDRT register. When the RIICnSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL line low on the ninth falling edge of SCL clock.

(5) When the RIICnSR2.NACKF flag or the RIICnSR2.TEND flag is 1, dummy read RIICnDRR to complete the processing. This releases the SCL line.

(6) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.GCA, and AASy (y = 0 to 2), flags RIICnSR2.TDRE and TEND, and the RIICnCR2.TRS bit to 0, and enters slave receive mode. Furthermore, it automatically sets the RIICnSR2.STOP flag to 1.

(7) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

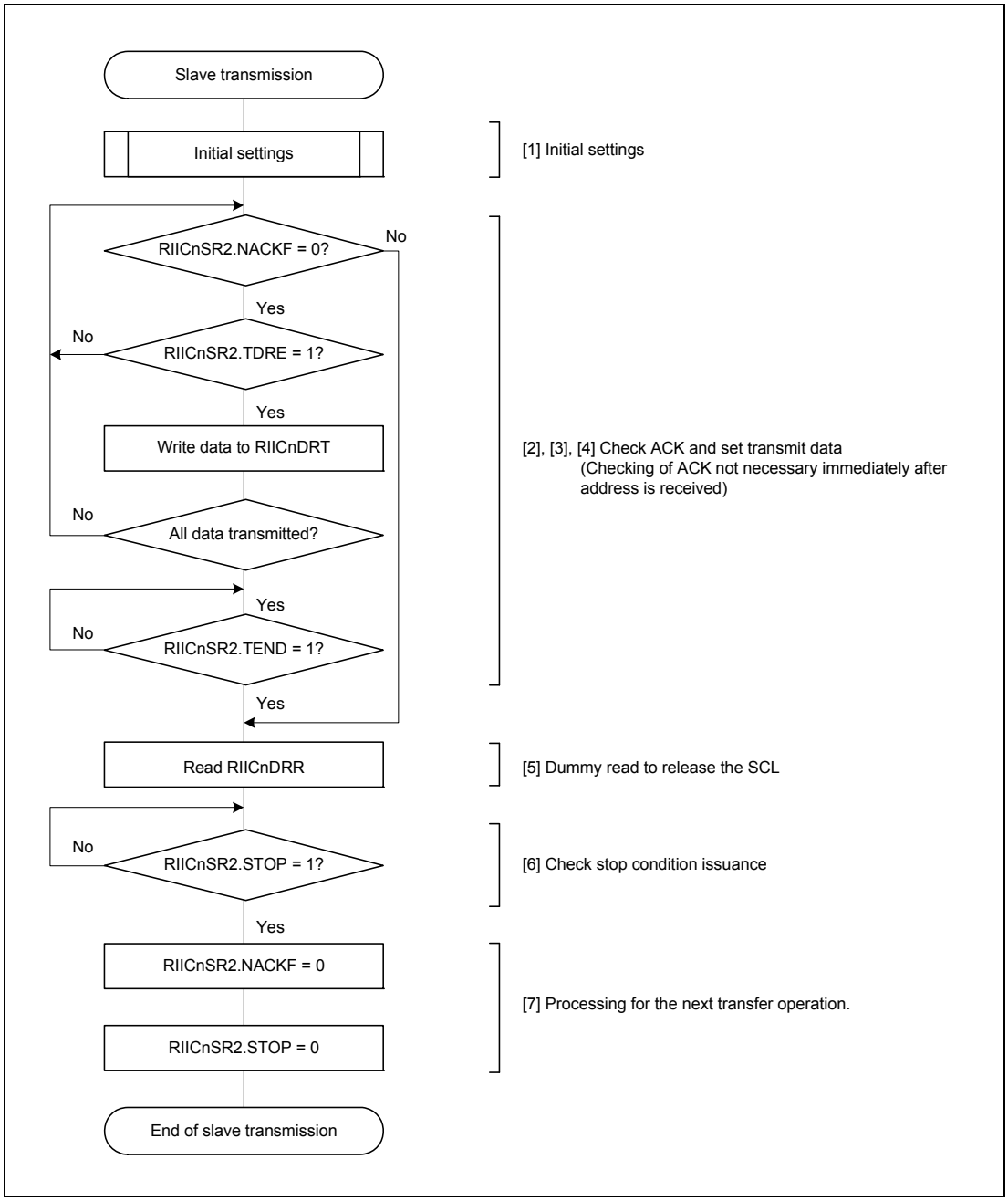


Figure 19.15 Example of Slave Transmission Flowchart

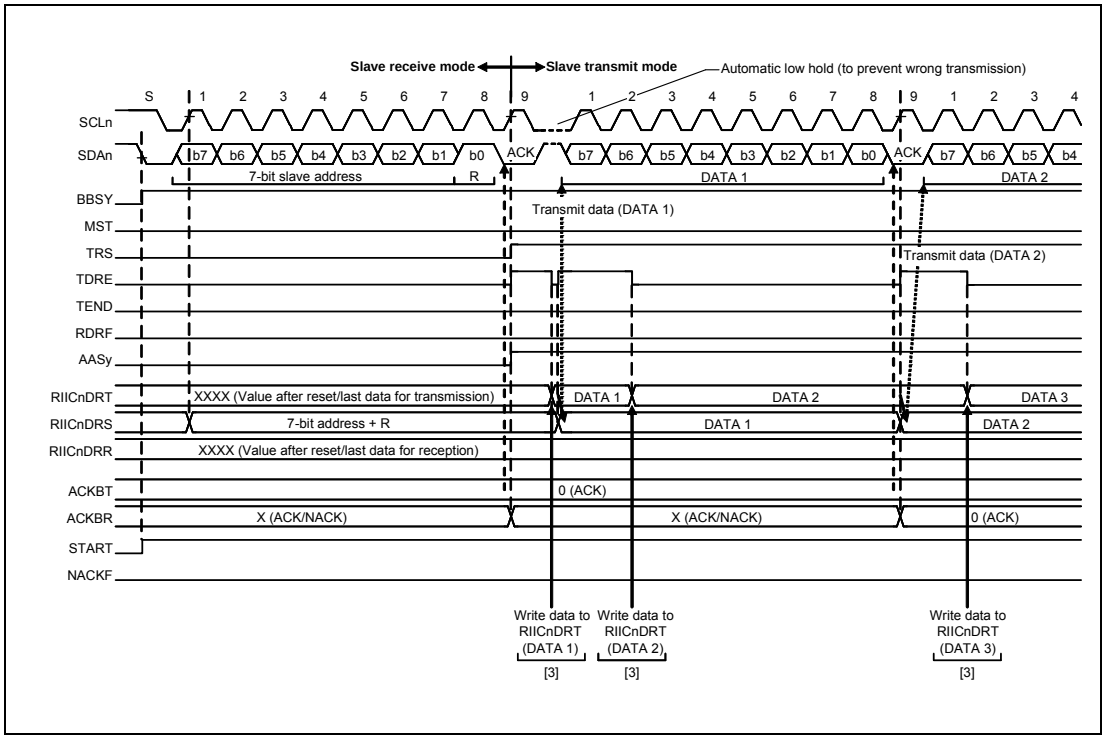


Figure 19.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

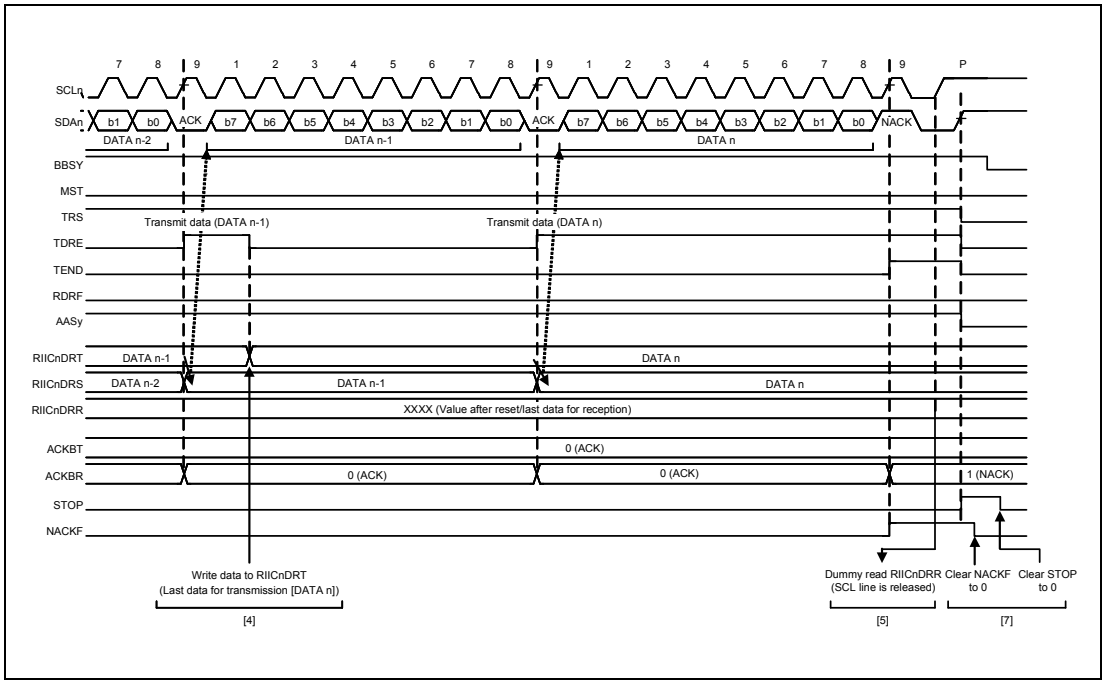


Figure 19.17 Slave Transmit Operation Timing (2)

19.5.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgements as a slave device.

Figure 19.18 shows an example of usage of slave reception and **Figure 19.19** and **Figure 19.20** show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Follow the procedure in **Figure 19.5** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC automatically sets the BBSY and RIICnSR2.START flags to 1 and automatically clears the ST bit to 0 on detection of a start condition.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1. GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the RIICnSR2.RDRF flag to 1.

(3) After the RIICnSR2.STOP flag is confirmed to be 0 and the RIICnSR2.RDRF flag to be 1, dummy read RIICnDRR as the first read operation (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower eight bits when the 10-bit address format is selected).

(4) When RIICnDRR is read, the RIIC automatically clears the RIICnSR2.RDRF flag to 0. If reading of RIICnDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading RIICnDRR releases the SCL line from being held at the low level.
When the RIICnSR2.STOP flag is 1 and the RIICnSR2.RDRF flag is also 1, read RIICnDRR until all the data is completely received.

(5) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.GCA, and AASy (y = 0 to 2) to 0. Furthermore, it automatically sets the RIICnSR2.STOP flag to 1.

(6) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.STOP flag to 0 for the next transfer operation.

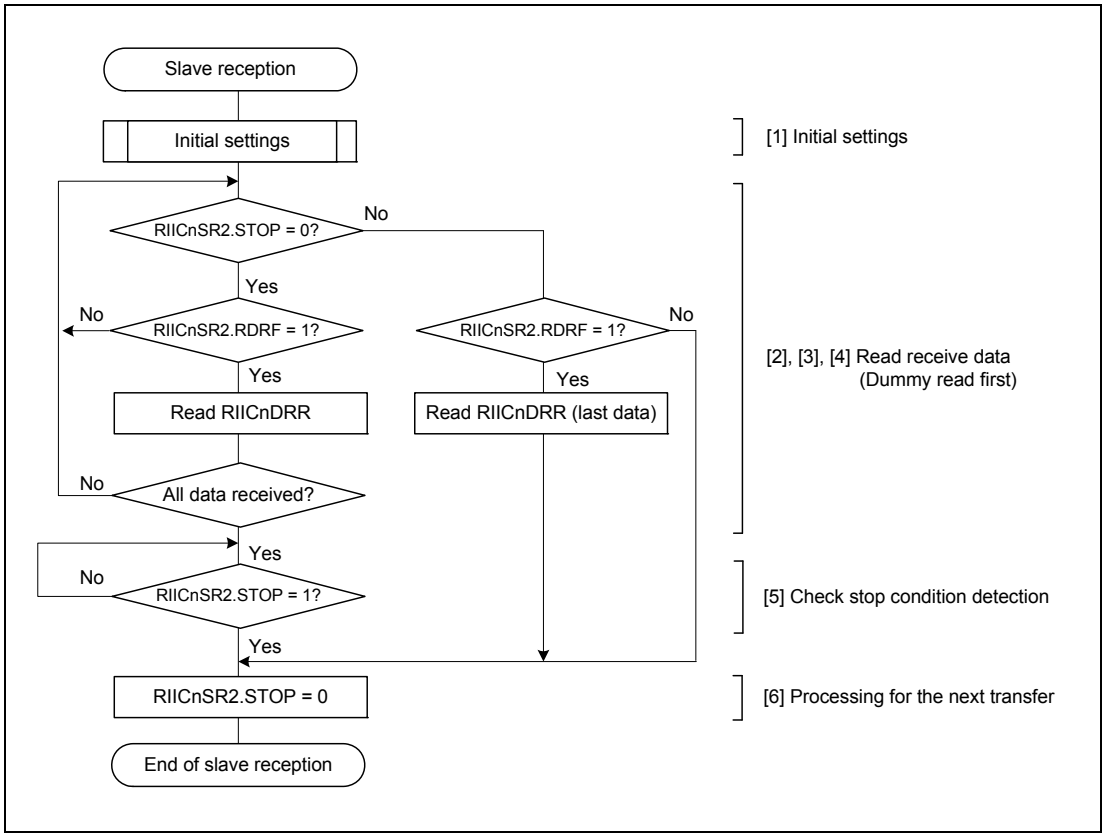


Figure 19.18 Example of Slave Reception Flowchart

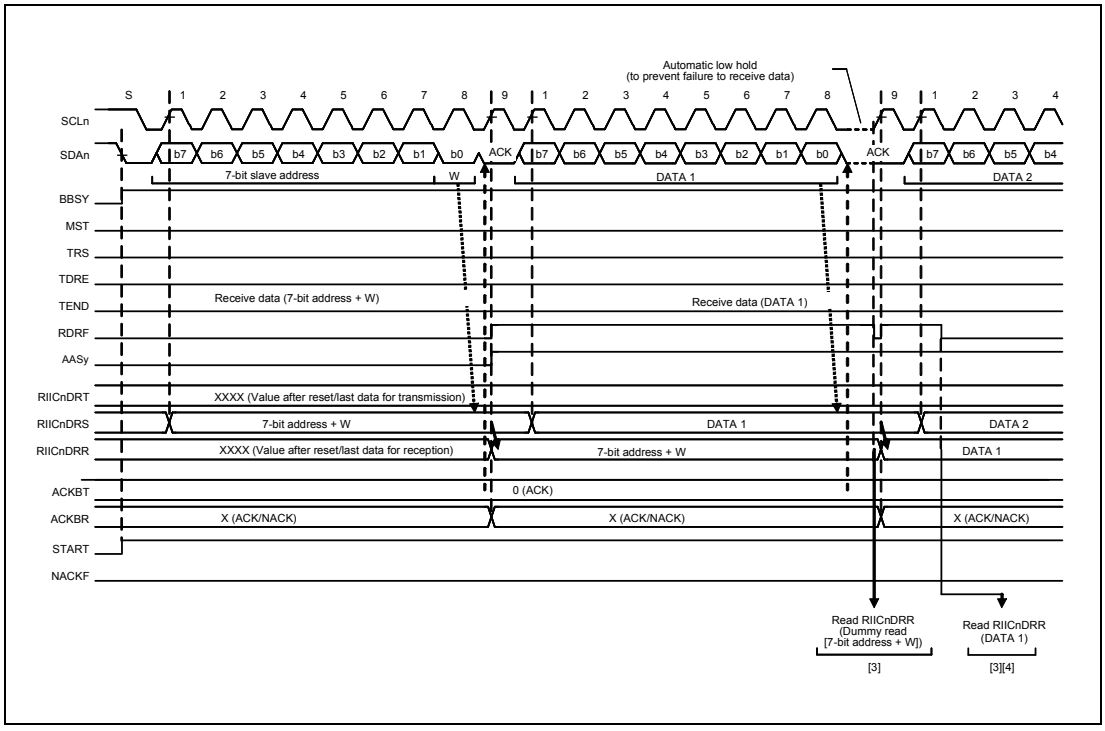


Figure 19.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

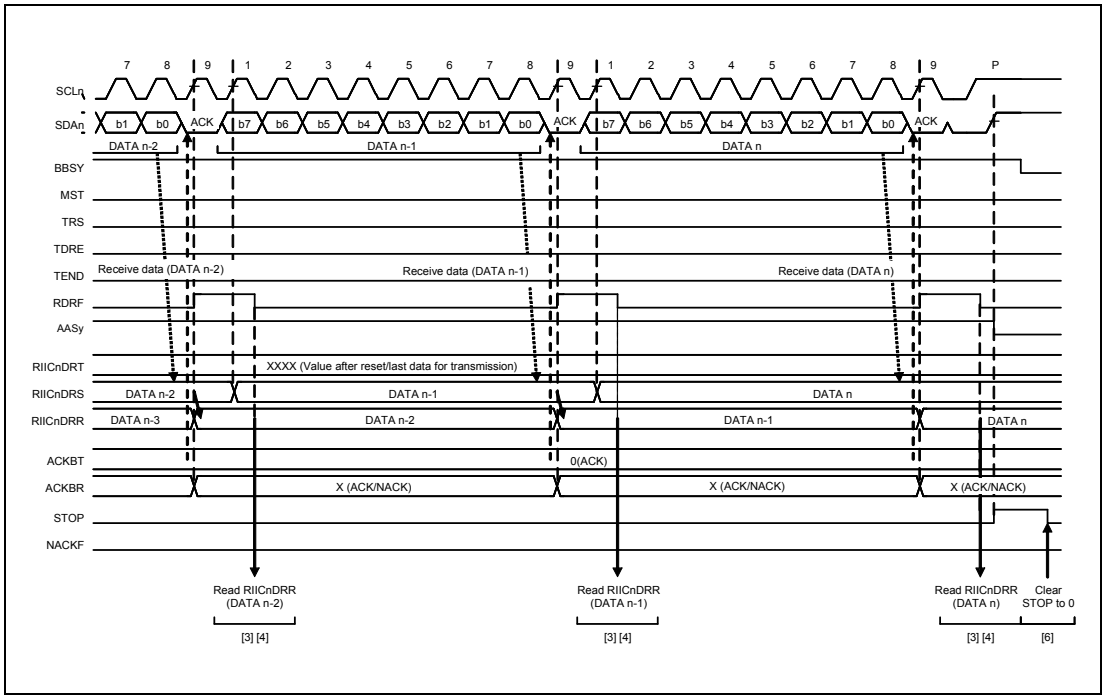


Figure 19.20 Slave Receive Operation Timing (2) (when RDRFS = 0)

19.6 SCL Synchronization Circuit

In generation of the SCL (clock) signal, the RIIC starts counting out the value for width at high level specified in RIICnBRH when it detects a rising edge on the SCL line and drives the SCL line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL line, it starts counting out the width at low level period specified in RIICnBRL, and then stops driving the SCL line (releases the line) once counting of the width at low level is complete. The SCL (clock) signal is thus generated.

If multiple master devices are connected to the I2C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCL line during communication.

When the RIIC has detected a rising edge on the SCL line and thus started counting out the width at high level specified in RIICnBRH, and the level on the SCL line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL line low, and starts counting out the width at low level specified in RIICnBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCL line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL line has been released. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the RIICnFER.SCLE bit is set to 1.

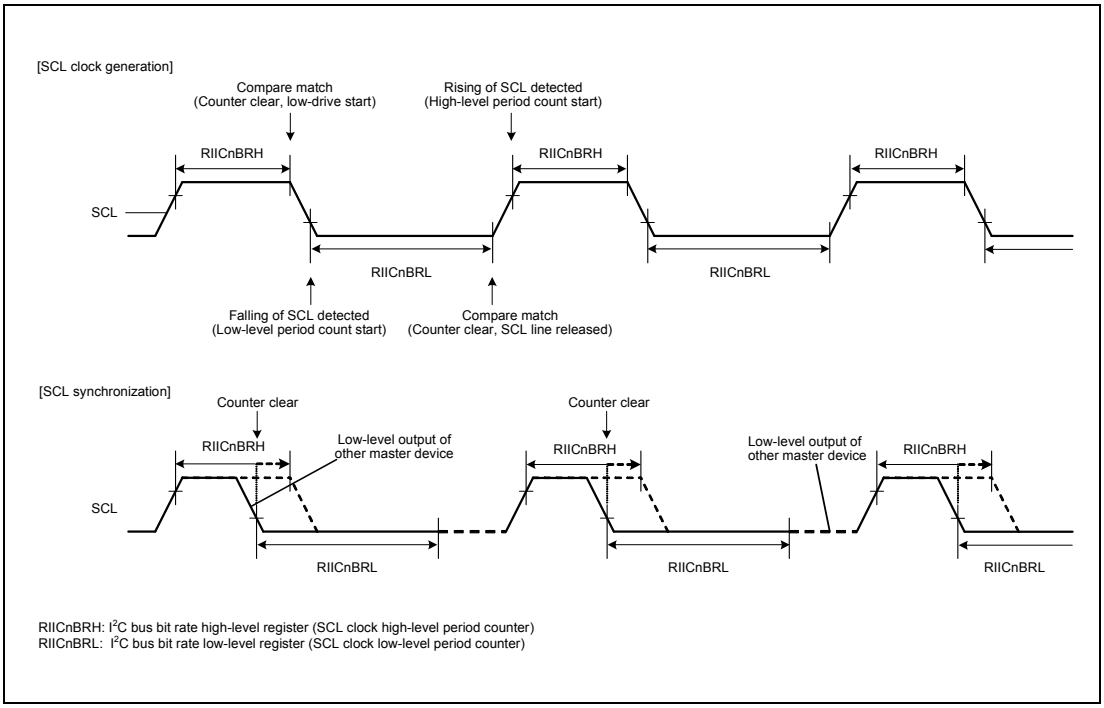


Figure 19.21 Generation and Synchronization of the SCL Signal from the RIIC

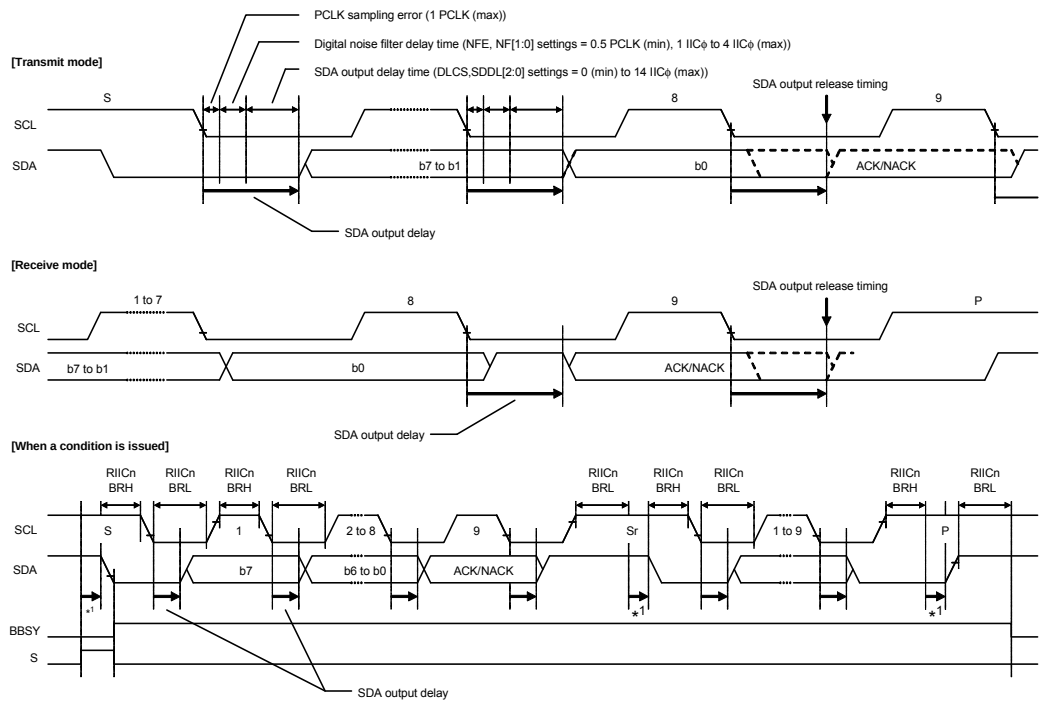
19.7 Facility for Delaying SDA Output

The RIIC module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL (clock) signal is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices.

The output delay facility is enabled by setting the RIICnMR2.SDDL[2:0] bits to any value other than 000_B, and disabled by setting the same bits to 000_B.

While the SDA output delay facility is enabled (i.e. while the SDDL[2:0] bits are set to any value other than 000_B), the RIICnMR2.DLCS bit selects the clock source for counting by the SDA output delay counter as the internal base clock (IICϕ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IICϕ/2). The counter counts the number of cycles set in the SDDL[2:0] bits. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.



Note 1. The output delay facility is set by the SDDL[2:0] bits when a start (S), restart (Sr), or stop (P) condition is issued.

Figure 19.22 SDA Output Delay Facility

19.8 Digital Noise-Filter Circuits

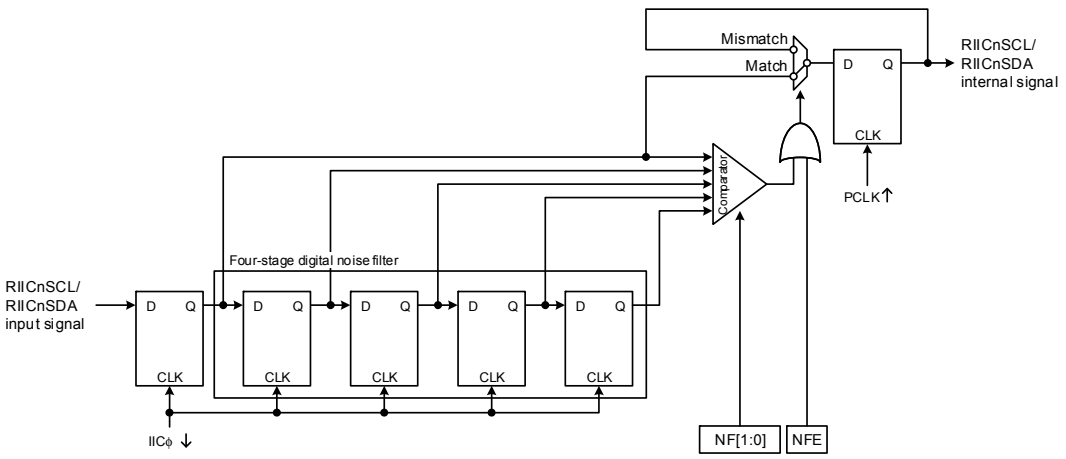
The states of the RIICnSCL and RIICnSDA pins are conveyed to the internal circuitry through the digital noise-filter circuit. **Figure 19.23** is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the RIIC consists of four flip-flop circuit stages connected in series and a match-detection circuit.

The number of effective stages in the digital noise filter is selected by the RIICnMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four IICϕ cycles.

The input signal to the RIICnSCL pin (or RIICnSDA pin) is sampled on falling edges of the IICϕ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the RIICnMR3.NF[1:0] bits, the signal level is conveyed as an internal signal. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is small, the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, it is possible to disable the digital noise-filter circuit (by clearing the RIICnFER.NFE bit to 0).



Note: NFE: Digital noise filter circuit enable bit
NF[1:0]: Digital noise filter stage selection bit

Figure 19.23 Block Diagram of Digital Noise Filter Circuit

19.9 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and device ID address, and also can set 7-bit or 10-bit slave addresses.

19.9.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the RIICnSER.SARy bit (y = 0 to 2) is set to 1, the slave addresses set in RIICnSARy (y = 0 to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding RIICnSR1.AASy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. The RIICnSR2.RDRF flag or the RIICnSR2.TDRE flag is set to 1 by the following R/W# bit. This causes a receive complete interrupt (INTIICnRI) or transmit data empty interrupt (INTIICnTI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 19.24 to **Figure 19.26** show the AASy flag set timing in three cases.

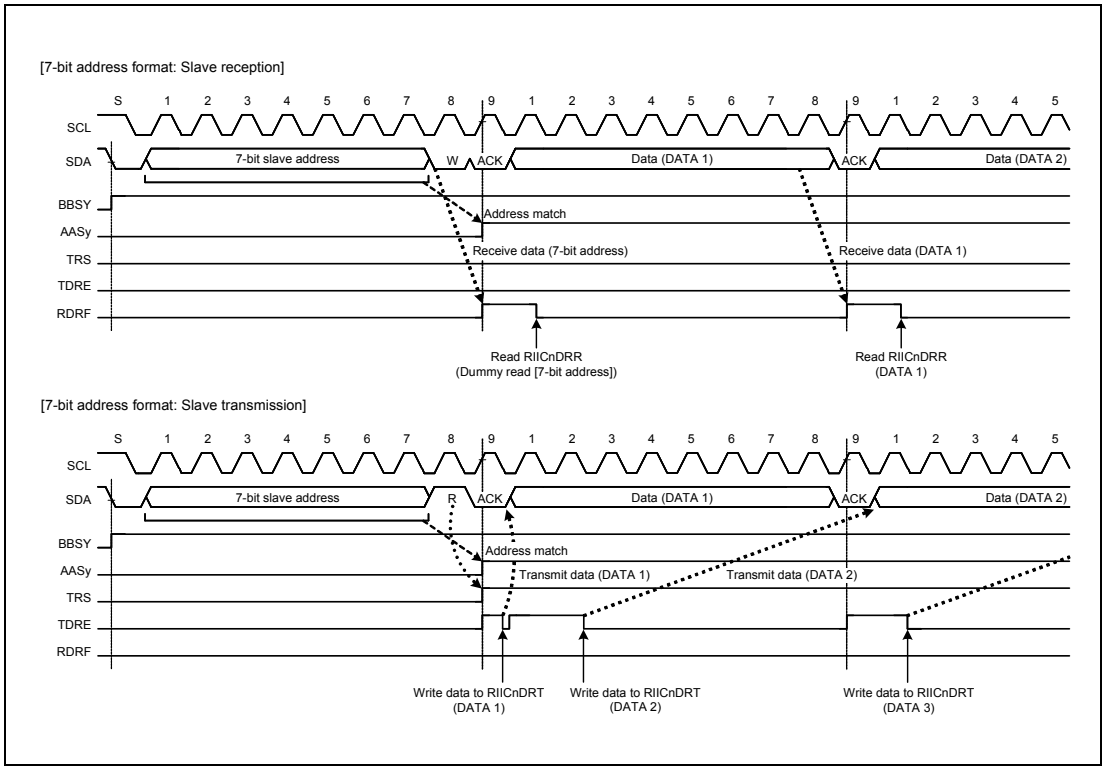


Figure 19.24 AASy Flag Set Timing with 7-Bit Address Format Selected

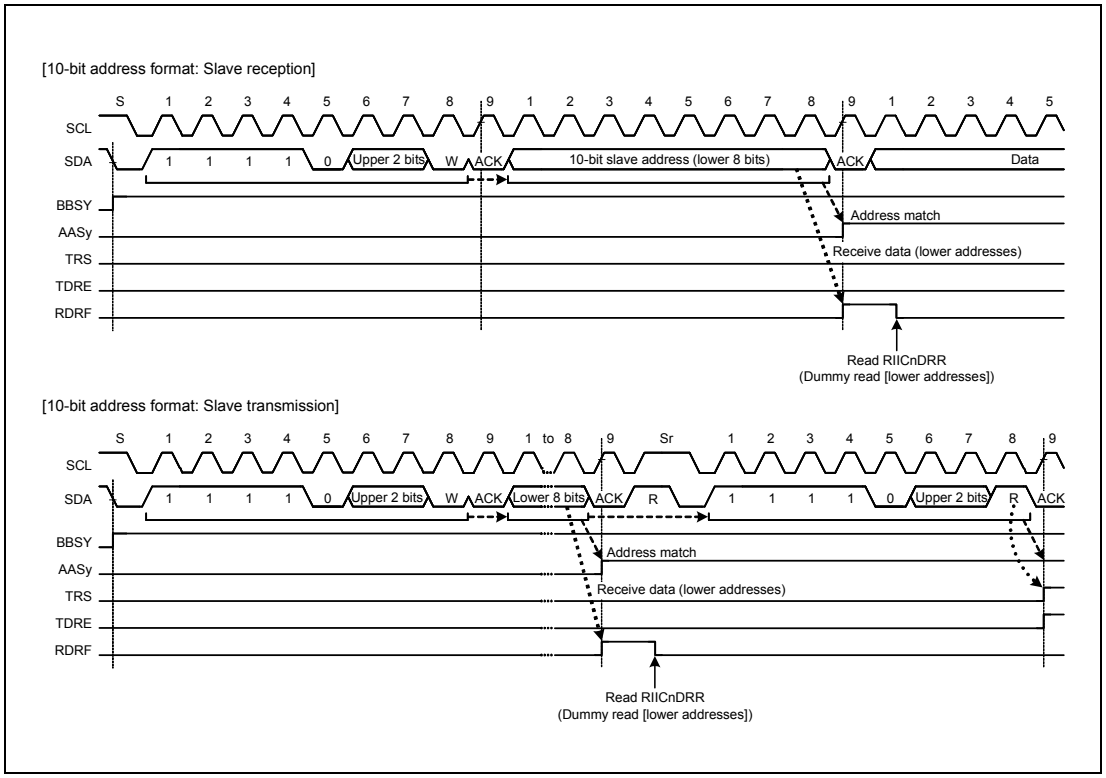


Figure 19.25 AASy Flag Set Timing with 10-Bit Address Format Selected

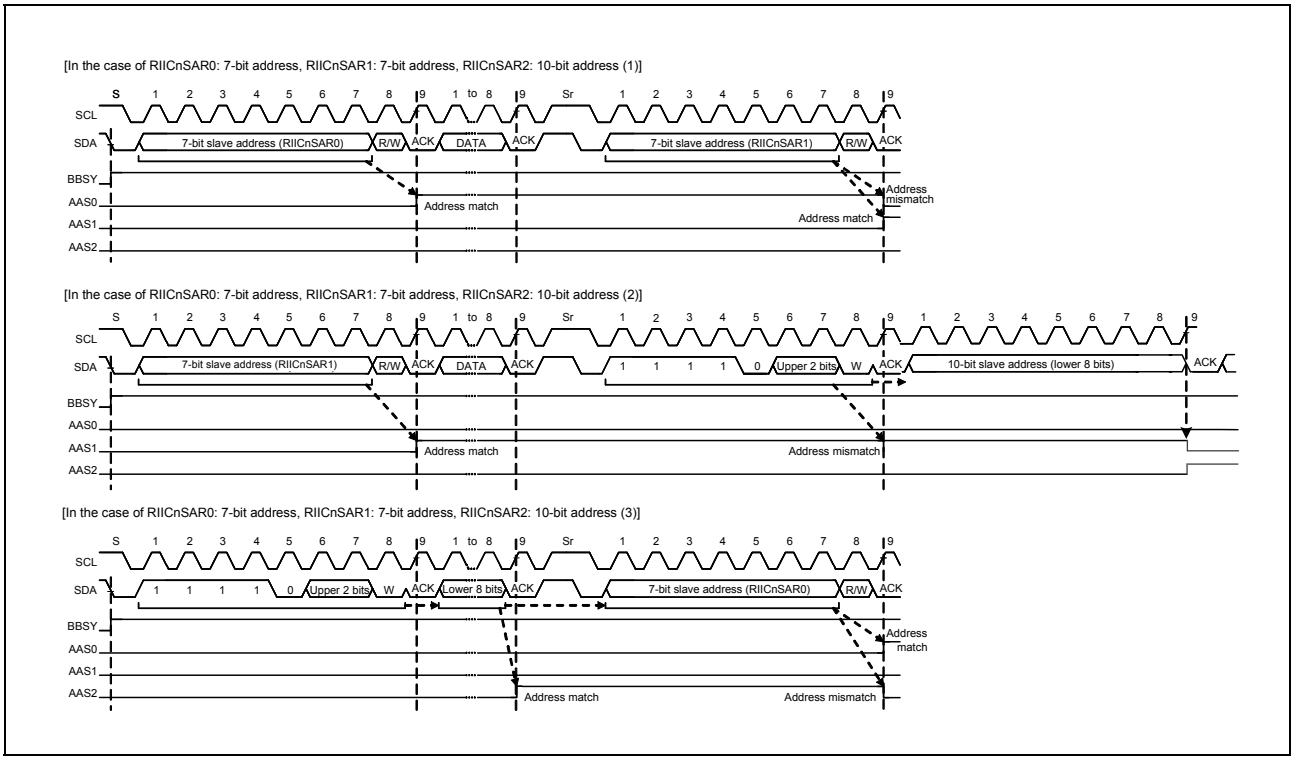


Figure 19.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

19.9.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address (0000 000_B + 0 [W]). This is enabled by setting the RIICnSER.GCAE bit to 1.

If the address received after a start or restart condition is issued is 0000 000_B + 1[R] (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the RIICnSR1.GCA flag and the RIICnSR2.RDRF flag are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive complete interrupt (INTIICnRI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

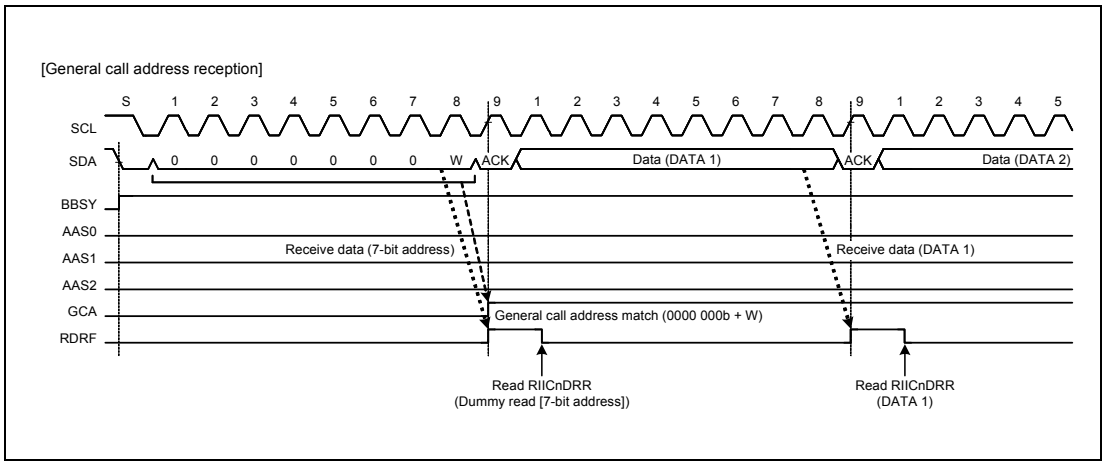


Figure 19.27 Timing of GCA Flag Setting during Reception of General Call Address

19.9.3 Device-ID Address Detection

The RIIC module has a facility for detecting device-ID addresses conforming with the I2C bus specification (Rev. 03). When the RIIC receives 1111 100_B as the first byte after a start condition or restart condition was issued with the RIICnSER.DIDE bit set to 1, the RIIC recognizes the address as a device ID, sets the RIICnSR1.DID flag to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding RIICnSR1.AASy flag (y = 0 to 2) to 1.

After that, when the first byte received after a start or restart condition is issued matches the device ID address (1111 100_B) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the RIICnSR2.TDRE flag to 1.

In the device-ID address detection function, the RIIC clears the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100_B) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC 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.

Furthermore, prepare the device-ID fields (three 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 data for transmission. For details, see I2C Bus Standard from NXP Semiconductors.

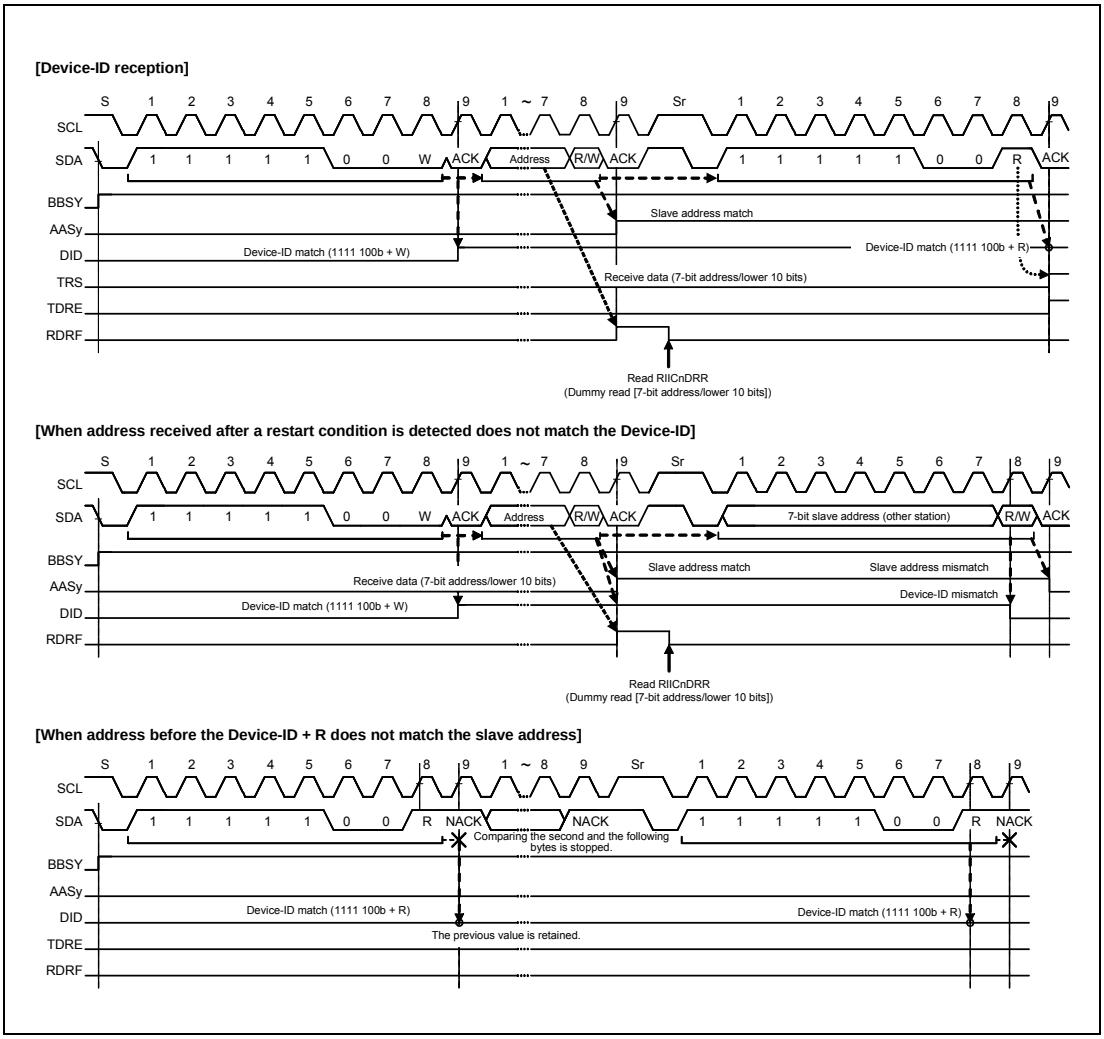


Figure 19.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

19.10 Automatic Low-Hold Function for SCL

19.10.1 Function to Prevent Wrong Transmission of Transmit Data

To prevent the unintended transmission of erroneous data, this low-hold period is extended until data for transmission have been written. In addition, the RIIC holds the SCL line low over the period until a stop condition is issued and also over the period until the RIICnDRR register is dummy read.

<Master transmit mode>

- Low-level interval after a start condition or restart condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next
- Low-level interval from the ninth clock cycle until a stop condition is issued

<Slave transmit mode>

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next
- Low-level interval from the ninth clock cycle and the RIICnDRR register is dummy read

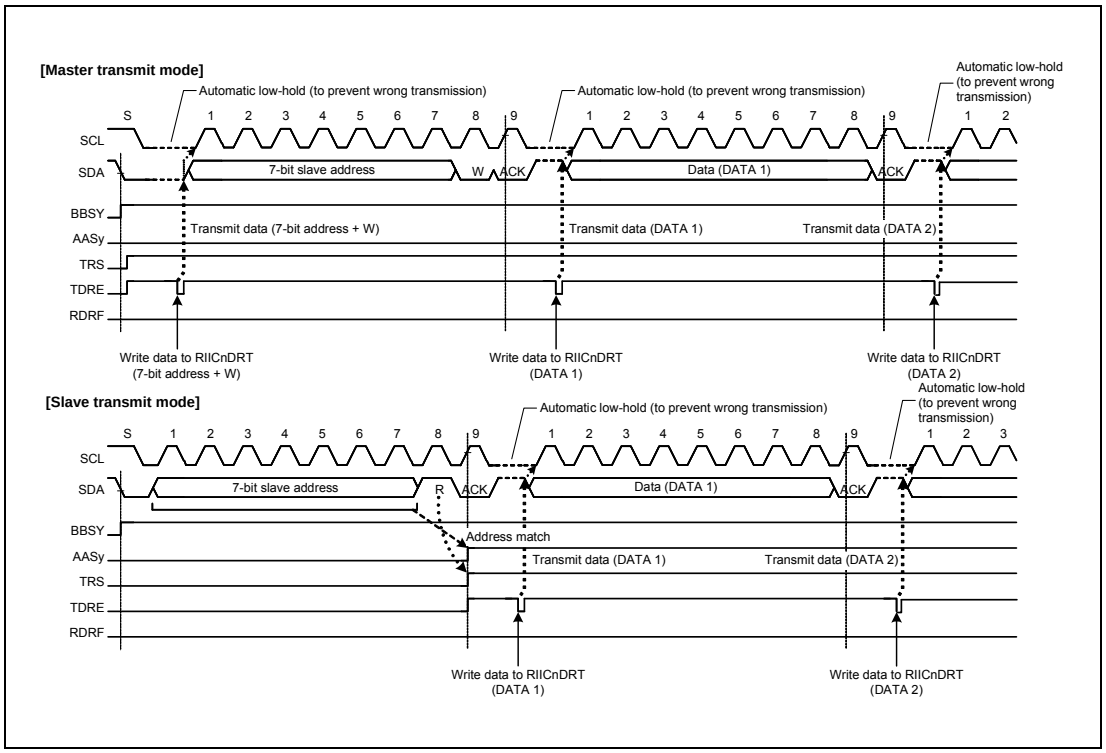


Figure 19.29 Automatic Low-Hold Operation in Transmit Mode

19.10.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (RIICnCR2.TRS bit = 1). This function is enabled when the RIICnFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (RIICnSR2.TDRE flag = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDA 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 (RIICnSR2.NACKF flag = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0, issue a restart or stop condition, and then issue a start condition again.

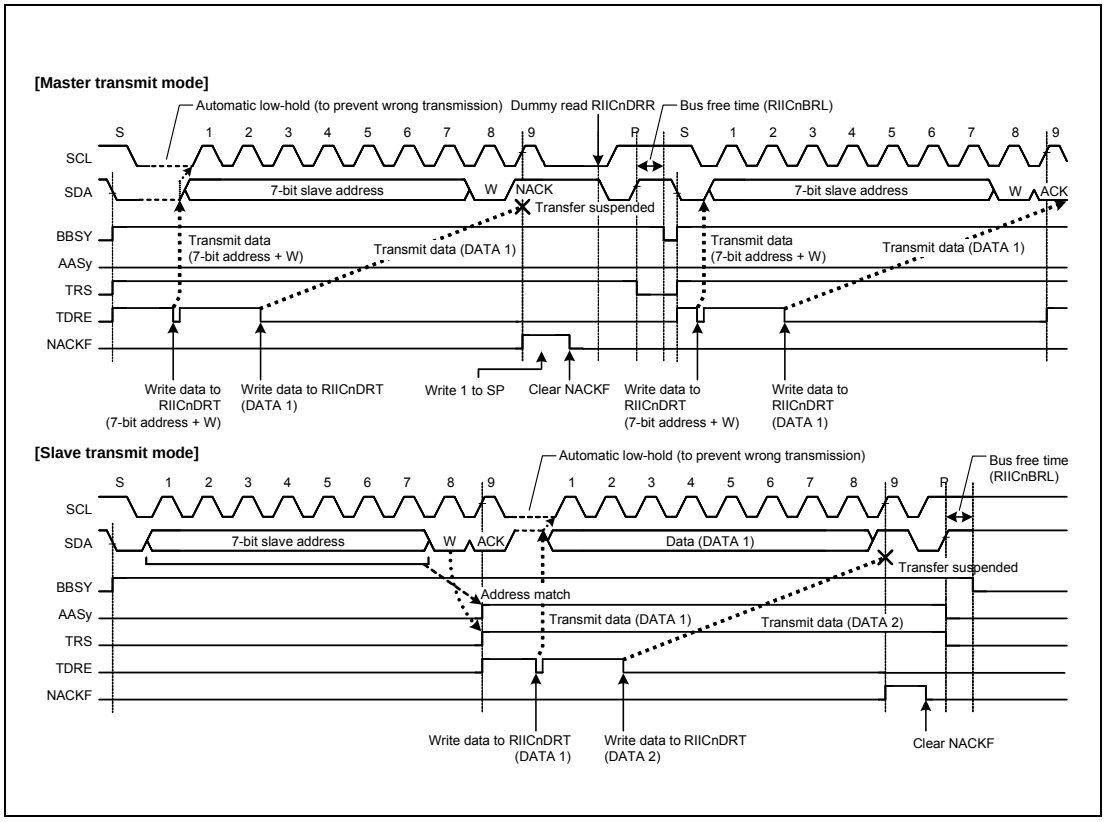


Figure 19.30 Suspension of Data Transfer when NACK is Received (NACKE = 1)

19.10.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (RIICnDRR) read is delayed for a period of one transfer frame or more with receive complete (RIICnSR2.RDRF flag = 1) in receive mode (RIICnCR2.TRS = 0), the RIIC holds the SCL line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCL line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCL line is held low can be selected with a combination of the RIICnMR3.WAIT and RDRFS bits.

(1) One-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the RIICnMR3.WAIT bit is set to 1, the RIIC performs one-byte receive operation using the WAIT bit function.
Furthermore, when the RIICnMR3.RDRFS bit is 0, the RIIC automatically sends the RIICnMR3.ACKBT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCL line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from RIICnDRR, which enables bytewise receive operation.

The WAIT bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and device ID address) is obtained in master receive mode or slave receive mode.

(2) One-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RIICnMR3.RDRFS bit is set to 1, the RIIC performs one-byte receive operation using the RDRFS bit function.
When the RIICnSR2.RDRFS bit is set to 1, the RDRF flag (receive complete) in RIICnSR2 is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCL line is automatically held low at the falling edge of the eighth SCL clock cycle. This lowhold is released by writing a value to the RIICnMR3.ACKBT bit, but cannot be released by reading data from RIICnDRR, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and device ID address) is obtained in master receive mode or slave receive mode.

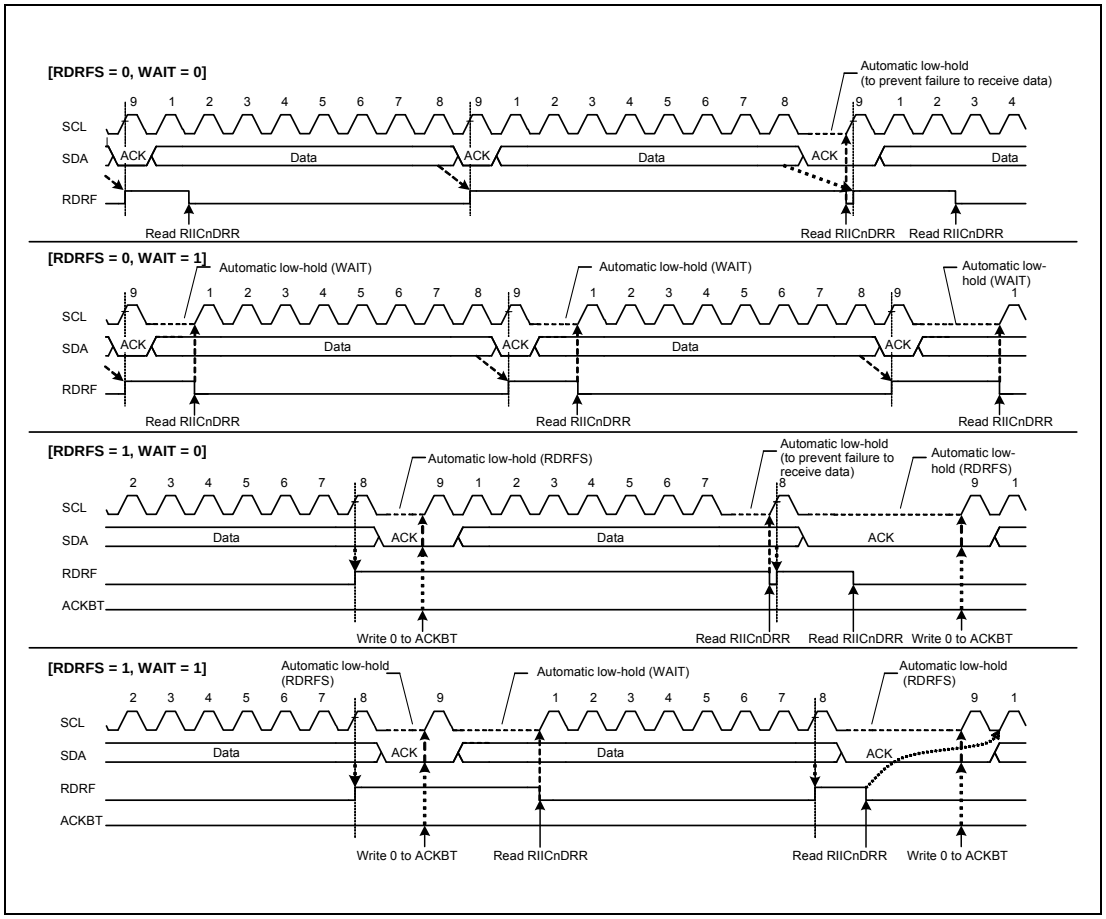


Figure 19.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

19.11 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C bus standard, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

19.11.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA line low to issue a start condition. However, if the SDA line has already been driven low by another master device issuing a start condition, the RIIC considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if the RIICnCR2.ST bit is set to 1 while the bus is busy (RIICnCR2.BBSY flag = 1), the RIIC considers itself to have lost in arbitration, so priority is given to transfer by the other master device and no start condition is generated.

When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state, and the low level is detected on the SDA line), the RIIC loses in arbitration.

After a loss in arbitration of mastership, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the RIICnFER.MALE bit is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDA line after a start condition was issued by setting the RIICnCR2.ST bit to 1 while the RIICnCR2.BBSY flag was cleared to 0 (erroneous issuing of a start condition)
- Setting of the RIICnCR2.ST bit to 1 (start condition double-issue error) while the RIICnCR2.BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA line in master transmit mode (RIICnCR2.MST and TRS bits = 11_B)

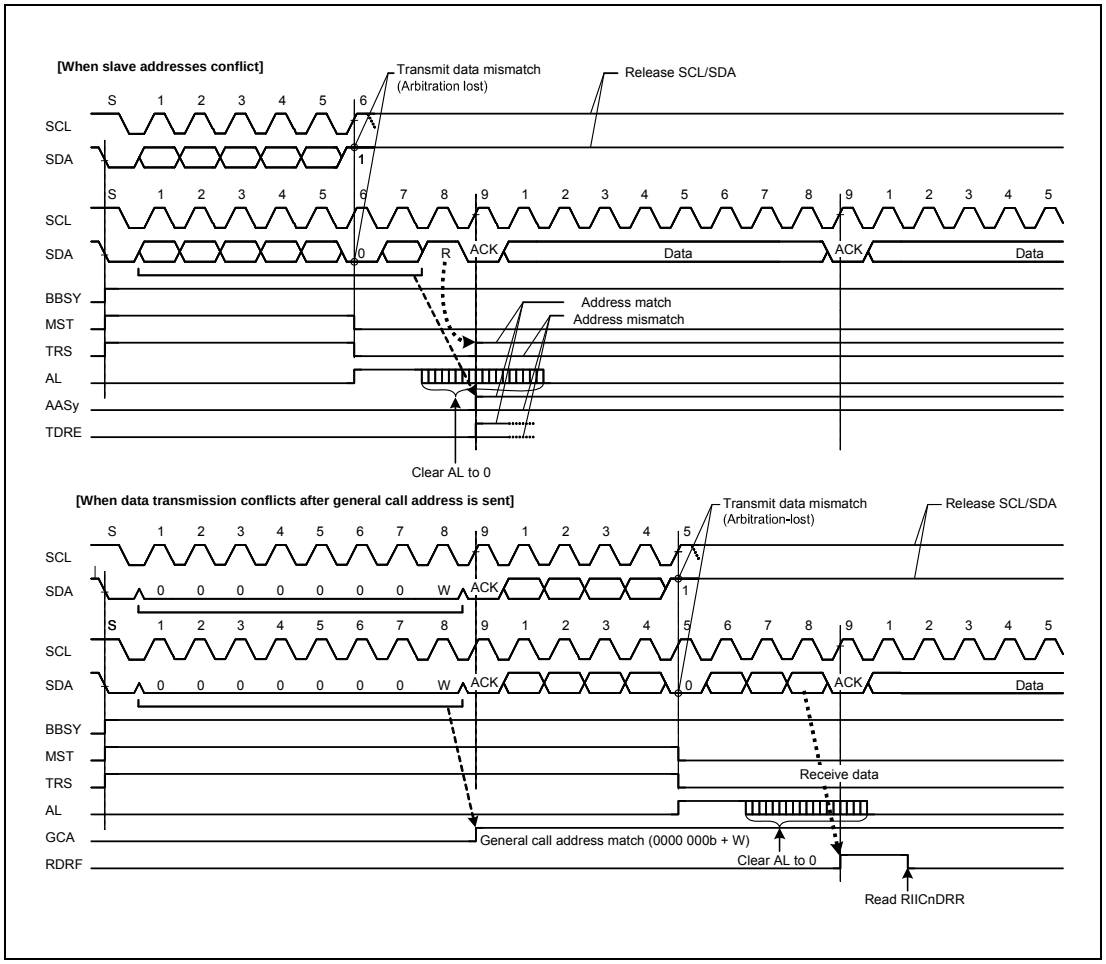


Figure 19.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

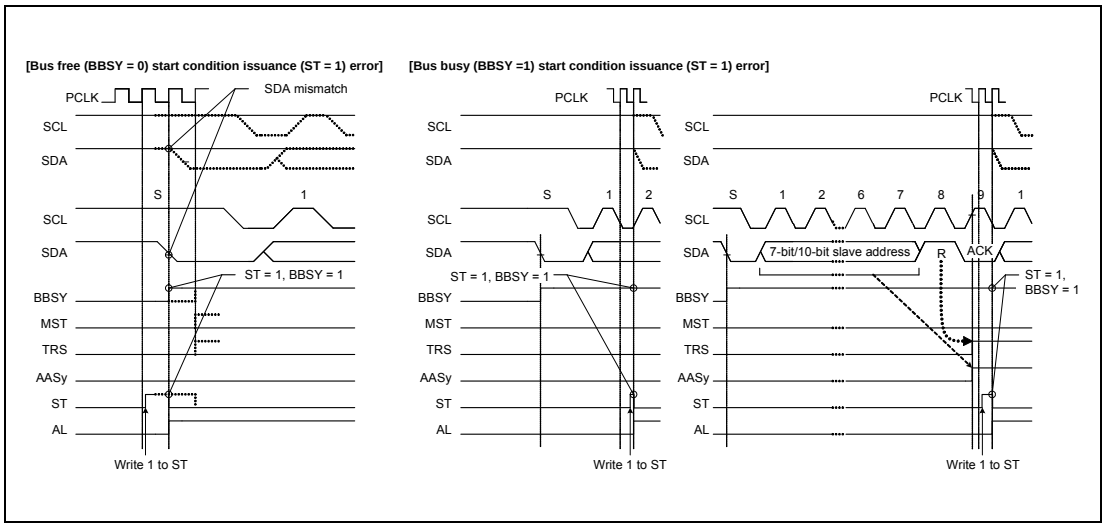


Figure 19.33 Arbitration-Lost when a Start Condition is Issued (MALE = 1)

The TRS bit is not cleared if 1 is written to the ST bit while the BBSY flag is 1 in slave transmit mode.

19.11.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA line (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state, and the low level is detected on the SDA line) during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. **Figure 19.34** shows an example of arbitration-lost detection during transmission of NACK.

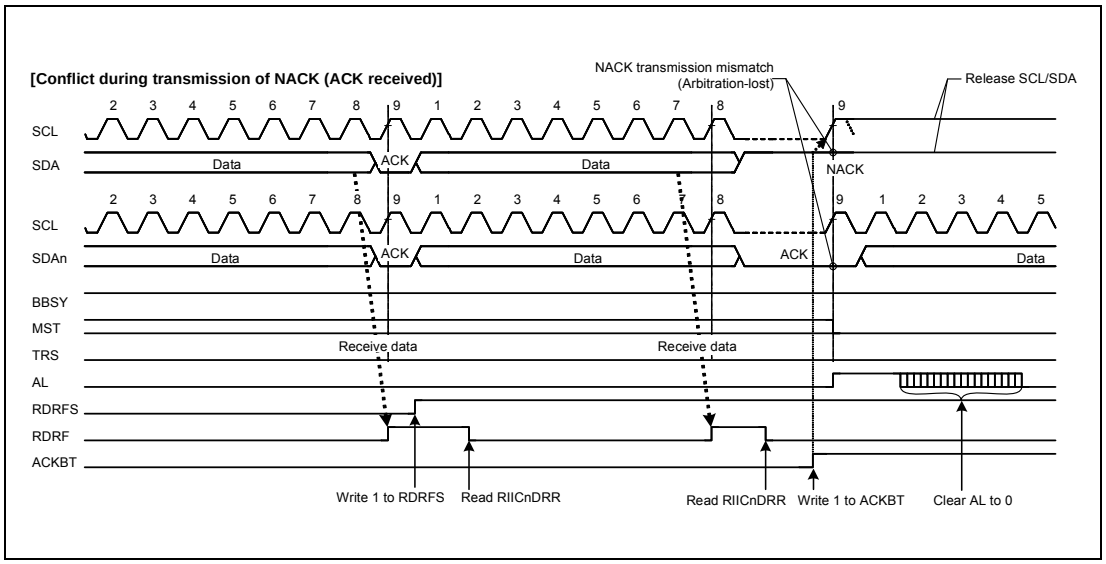


Figure 19.34 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives two bytes of data from the slave device, and master B receives four 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 both master A and 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. Here, master A sends NACK when it has received two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four 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 ACK transmitted by master B and issues a stop condition.

Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When the RIIC 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 RIIC enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the RIICnFER.NALE bit 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 line (ACK is received) during transmission of NACK (RIICnMR3.ACKBT bit = 1)

19.11.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the highimpedance state, and the low level is detected on the SDA line) in slave transmit mode.

When it loses slave arbitration, the RIIC enters slave receive mode.

The RIIC detects slave arbitration-lost when the following condition is met with the RIICnFER.SALE bit 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 line in slave transmit mode (RIICnCR2.MST and TRS bits = 01_B)

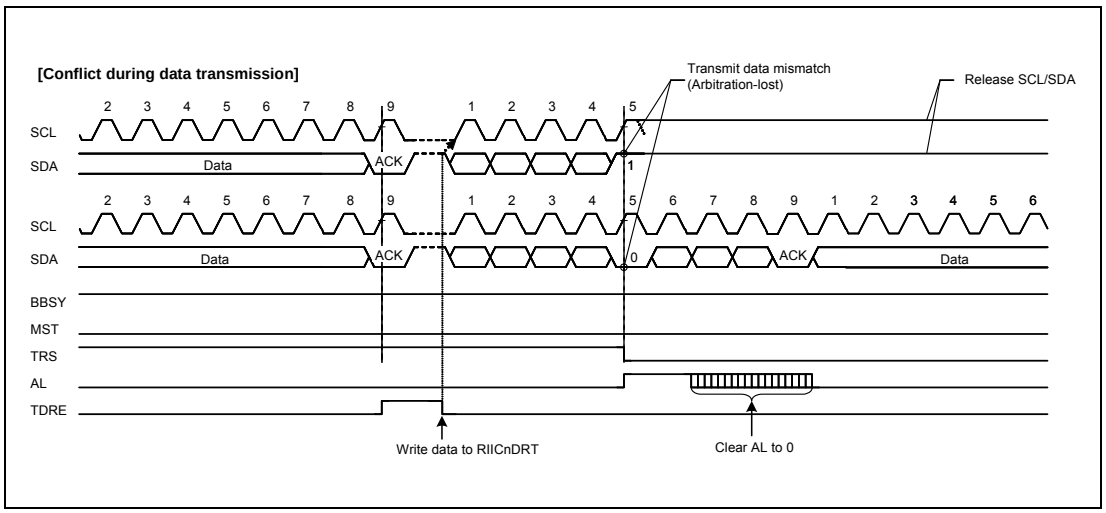


Figure 19.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

19.12 Start Condition/Restart Condition/Stop Condition Issuing Function

19.12.1 Issuing a Start Condition

The RIIC issues a start condition when the RIICnCR2.ST bit is set to 1.

When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the RIICnCR2.BBSY flag is 0 (bus free). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.

A start condition is issued in the following sequence.

[Start condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the start condition hold time.
- Drive the SCL line low (high level to low level).
- Detect low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

19.12.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RIICnCR2.RS bit is set to 1.

When the RS bit is set to 1, a restart condition issuance request is made even during communication and the RIIC issues a restart condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

(To detect the issuance of a restart condition, clear the RIICnSR2.START flag before a restart condition is issued.)

A restart condition is issued in the following sequence.

[Restart condition issuance]

- Release the SDA line.
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRL and the restart condition setup time.
- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the restart condition hold time.
- Drive the SCL line low (high level to low level).
- Detect a low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

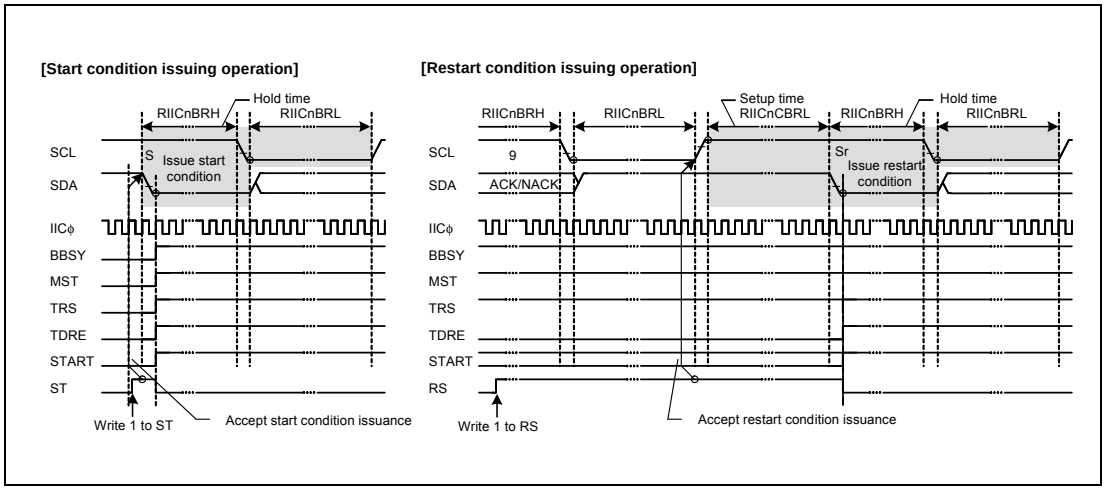


Figure 19.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

19.12.3 Issuing a Stop Condition

The RIIC issues a stop condition when the RIICnCR2.SP bit is set to 1.

When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRH and the stop condition setup time.
- Release the SDA line (low level to high level).
- Ensure the time set in RIICnBRL and the bus free time.
- Clear the BBSY flag to 0 (to release the bus mastership).

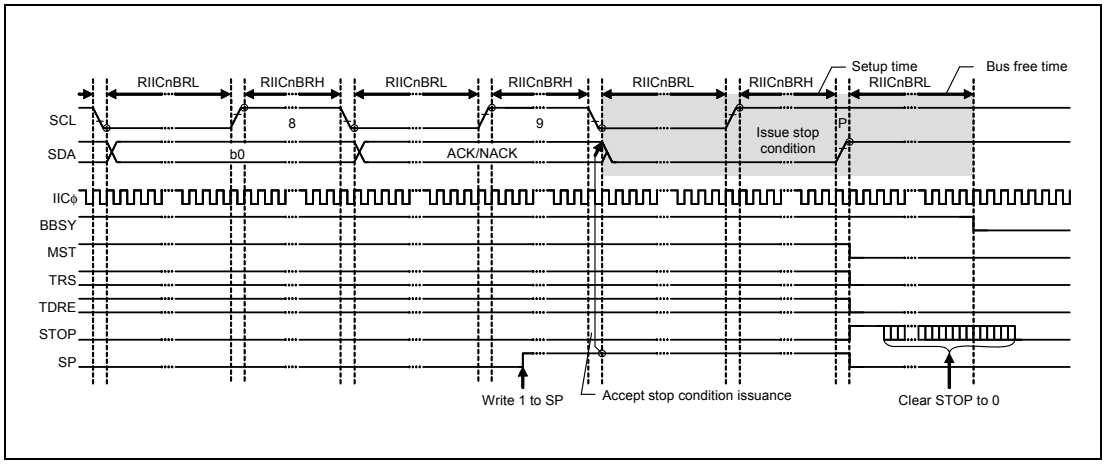


Figure 19.37 Stop Condition Issue Timing (SP Bit)

19.13 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I2C bus might hang with a fixed level on the SCL line and/or SDA line.

As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, and the RIIC/internal reset function.

By checking the RIICnCR1.SCLO, SDAO, SCLI, and SDAI bits, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL or SDA lines.

19.13.1 Timeout Function

The RIIC has the timeout function to detect an abnormality that the SCL line is held for a certain period of time. The RIIC can detect an abnormal bus state by monitoring that the SCL line is held low or high for a predetermined time.

The timeout function monitors the SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL line changes (rising or falling), but continues to count unless the SCL line changes. If the internal counter overflows due to no SCL line change, the RIIC can detect the timeout and report the bus abnormality.

The internal counter is cleared when one of the conditions is met.

(1) When RIICnMR2.TMOH=0, and RIICnMR2.TMOL=1:
The internal counter is cleared by SCL rising

(2) When RIICnMR2.TMOH=1, and RIICnMR2.TMOL=0:
The internal counter is cleared by SCL falling

(3) When RIICnMR2.TMOH=RIICnMR2.TMOL=1:
The internal counter is cleared by SCL rising or falling

This timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is stuck low or high during the following conditions:

- The bus is busy (RIICnCR2.BBSY flag is 1) in master mode (RIICnCR2.MST bit is 1).
- The RIIC's own slave address matches (RIICnSR1 register is not 00_H) and the bus is busy (RIICnCR2.BBSY flag is 1) in slave mode (RIICnCR2.MST bit is 0).
- The bus is free (RIICnCR2.BBSY flag is 0) while generation of a START condition is requested (RIICnCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IICϕ) set by the RIICnMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (RIICnMR2.TMOS bit = 0) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCL line level (low/high or both levels) during which this counter is activated can be selected by the setting of the RIICnMR2.TMOH and TMOL bits. If both TMOL and TMOH bits are cleared to 0, the internal counter does not work.

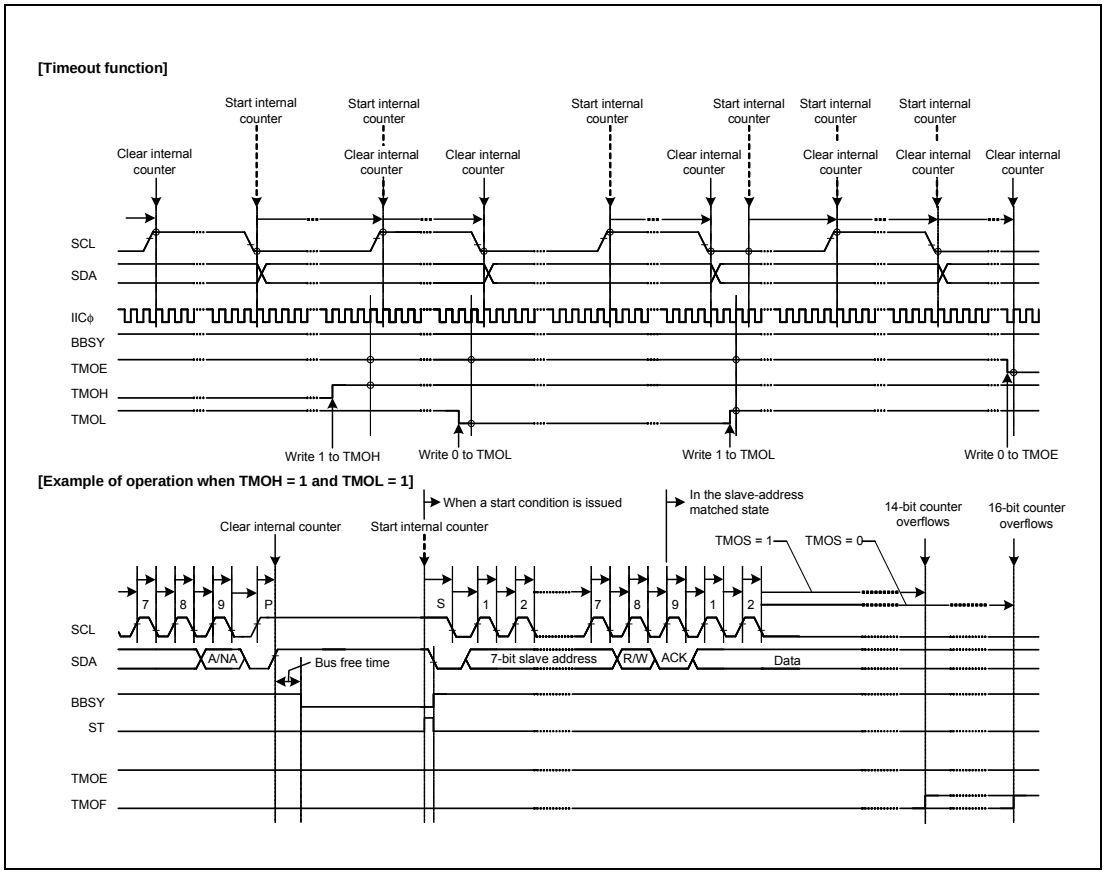


Figure 19.38 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)

19.13.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL (clock) cycles to release the SDA line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the SDA line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL (clock) signal as the unit in the case of a bus error where the RIIC cannot issue a stop condition because the slave device is holding the SDA line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the RIICnCR1.CLO bit is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the RIICnMR1.CKS[2:0] bits, and of the RIICnBRH and RIICnBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically cleared to 0. Therefore, further extra clock cycles can be output consecutively by the software program writing 1 to the CLO bit after having read CLO = 0.

When the RIIC module is in master mode and the slave device is holding the SDA line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The facility for output of an extra cycle of the SCL (clock) signal can be used to output extra cycles of SCL one by one to make the slave device release the SDA line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA line by the slave device can be monitored by reading the RIICnCR1.SDAI bit. After confirming release of the SDA line by the slave device, complete communications by reissuing the stop condition.

Use this facility with the RIICnFER.MALE bit (master arbitration-lost detection disabled) cleared to 0. If the MALE bit is set to 1 (master arbitration-lost detection enabled), arbitration is lost when the value of the RIICnCR1.SDAO bit does not match the state of the SDA line, so take care on this point.

[Additional output conditions for the SCL clock]

- In master mode and when the bus is free
- In master mode and the SCL line is not held low (the bus is busy)

Figure 19.39 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

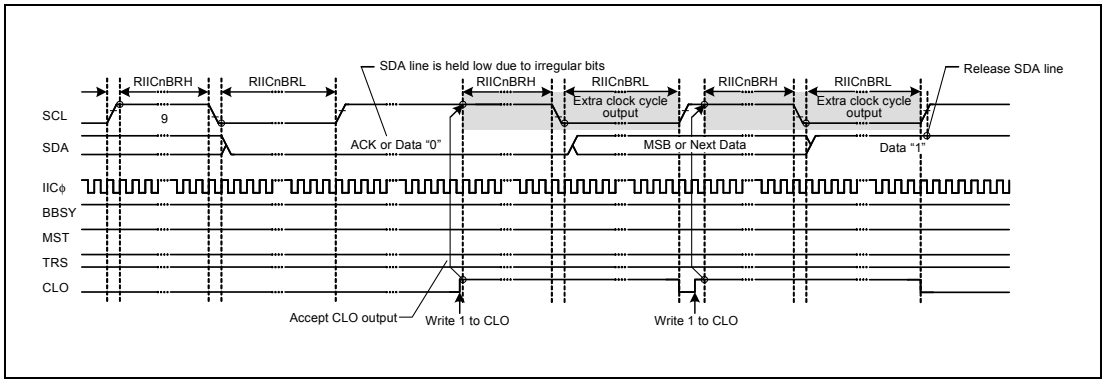


Figure 19.39 Extra SCL Clock Cycle Output Function (CLO Bit)

19.13.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the RIICnCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings.

After issuing a reset, be sure to clear the RIICnCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCL and SDA pins to the high impedance state.

Issuing a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (RIICnCR1.ICE and IICRST bits = 01_B).

For a detailed description of the RIIC and internal resets, see **Section 19.14, Reset Function of RIIC**.

19.14 Reset Function of RIIC

The RIIC has RIIC reset, and internal reset functions. In addition RIIC is cleared by ISORES. **Table 19.26** lists the scope of each reset and reset conditions.

Table 19.26 RIIC Reset Functions (1/2)

UM		ISORES	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start/Restart Condition Detection	Stop Condition Detection
RIICnCR1	ICE	Initialized	0	1	Retained	Retained
	IICRST	Initialized	1	1	Retained	Retained
	CLO	Initialized	Initialized	Retained	Retained	Retained
	SOWP	Initialized	Initialized	Retained	Retained	Retained
	SCLO	Initialized	Initialized	Initialized	Retained	Retained
	SDAO	Initialized	Initialized	Initialized	Retained	Retained
	SCLI	Initialized	Initialized	Retained	Retained	Retained
	SDAI	Initialized	Initialized	Retained	Retained	Retained
RIICnCR2	BBSY	Initialized	Initialized	Initialized *1	Operation	Retained
	MST	Initialized	Initialized	Initialized	Operation (retained)	Initialized
	TRS	Initialized	Initialized	Initialized	Operation (retained)	Initialized
	SP	Initialized	Initialized	Initialized	Initialized	Initialized
	RS	Initialized	Initialized	Initialized	Initialized	Initialized
	ST	Initialized	Initialized	Initialized	Initialized	Retained
RIICnMR1	MTWP	Initialized	Initialized	Retained	Retained	Retained
	CKS[2:0]	Initialized	Initialized	Retained	Retained	Retained
	BCWP	Initialized	Initialized	Retained	Retained	Retained
	BC[2:0]	Initialized	Initialized	Initialized	Initialized	Retained
RIICnMR2		Initialized	Initialized	Retained	Retained	Retained
RIICnMR3	WAIT	Initialized	Initialized	Retained	Retained	Retained
	RDRFS	Initialized	Initialized	Retained	Retained	Retained
	ACKWP	Initialized	Initialized	Retained	Retained	Retained
	ACKBT	Initialized	Initialized	Retained	Retained	Initialized
	ACKBR	Initialized	Initialized	Retained	Retained	Retained
	NF[1:0]	Initialized	Initialized	Retained	Retained	Retained
RIICnFER		Initialized	Initialized	Retained	Retained	Retained
RIICnSER		Initialized	Initialized	Retained	Retained	Retained
RIICnIER		Initialized	Initialized	Retained	Retained	Retained
RIICnSR1	DID	Initialized	Initialized	Initialized	Retained	Initialized
	GCA	Initialized	Initialized	Initialized	Retained	Initialized
	AAS2	Initialized	Initialized	Initialized	Retained	Initialized
	AAS1	Initialized	Initialized	Initialized	Retained	Initialized
	AAS0	Initialized	Initialized	Initialized	Retained	Initialized
RIICnSR2	TDRE	Initialized	Initialized	Initialized	Retained	Initialized
	TEND	Initialized	Initialized	Initialized	Retained	Initialized
	RDRF	Initialized	Initialized	Initialized	Retained	Retained
	NACKF	Initialized	Initialized	Initialized	Retained	Retained
	STOP	Initialized	Initialized	Initialized	Retained	Operation
	START	Initialized	Initialized	Initialized	Operation	Initialized
	AL	Initialized	Initialized	Initialized	Retained	Retained
	TMOF	Initialized	Initialized	Initialized	Retained	Retained
RIICnSAR0, 1, 2		Initialized	Initialized	Retained	Retained	Retained

Table 19.26 RIIC Reset Functions (2/2)

UM	ISORES	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start/Restart Condition Detection	Stop Condition Detection
RIICnBRH, RIICnBRL	Initialized	Initialized	Retained	Retained	Retained
RIICnDRT	Initialized	Initialized	Retained	Retained	Retained
RIICnDRR	Initialized	Initialized	Retained	Retained	Retained
RIICnDRS	Initialized	Initialized	Initialized	Retained	Retained

Note 1. When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.
When an internal reset is applied while the bus is not free, the BBSY flag is not cleared.

Section 20 CAN Interface (RS-CAN)

This section contains a generic description of the CAN Interface (RS-CAN).

The first part of this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RS-CAN.

20.1 Features of RH850/F1M RS-CAN

20.1.1 Number of Units and Channels

This microcontroller has the following number of RS-CAN units.

Table 20.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	1		
Name	RSCANn (n = 0)		

The individual products have the CAN interface channels listed below.

Table 20.2 Unit Configurations and Channels

Unit Name	Unit Channel Number	Channel Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
RSCAN0	0	CAN0	√	√	√
	1	CAN1	√	√	√
	2	CAN2	√	√	√
	3	CAN3	√	√	√
	4	CAN4	√	√	√
	5	CAN5	√	√	√

Table 20.3 Indices (1/2)

Index	Description
n	Throughout this section, the individual RS-CAN units are generically indicated by the index “n” (n = 0); for example, RSCANnGCTR is the global control register of the RSCANn unit.
m	Throughout this section, the individual channels of RS-CAN units is generically indicated by the index “m” (m = 0 to 5); for example, RSCAN0CmSTS is the channel m status register.
j	The individual registers associated with receive rule table are generically indicated by the index “j” (j = 0 to 15); for example, RSCAN0GAFLIDj is the receive rule ID register.
k	The individual transmit/receive FIFO buffers are generically indicated by the index “k” (k = 0 to channel m × 3 + 2); for example, RSCAN0CFCCk is the transmit/receive FIFO buffer configuration/control register.
x	The individual receive FIFO buffers are identified by the index “x” (x = 0 to 7): for example, RSCAN0RFSTSx is the receive FIFO buffer status register.
q	The individual receive buffers are generically indicated by the index “q” (q = 0 to channel m × 16 + 15); for example, RSCAN0RMIDq is the receive buffer ID register.

Table 20.3 Indices (2/2)

Index	Description
p	The individual transmit buffers are generically indicated by the index “p” (p = 0 to channel m × 16 + 15; for example, RSCAN0TMCp is the transmit buffer control register.
r	The individual RAM tests for CAN are generically indicated by the index “r” (r = 0 to 63); for example, RSCAN0RPGACCr is the RAM test page access register.
y	The registers not covered above are indicated by the index “y” (y = 0 to 2); for example, RSCAN0RMNDy is a receive buffer new data register.

The following table lists the values of indices for individual products.

Table 20.4 Indices for Individual Products

Unit Name	Indices for Individual Products		
	144 pins	176 pins	233 pins
RSCAN0	j = 0 to 15		
	k = 0 to 17		
	x = 0 to 7		
	q = 0 to 95		
	p = 0 to 95		
	r = 0 to 63		
	y = 0 to 2		

20.1.2 Register Base Address

RSCAN0 base address is listed in the following table.

RSCAN0 register addresses are given as offsets from the base address.

Table 20.5 Register Base Address

Base Address Name	Base Address
<RSCAN0_base>	FFD0 0000_H

20.1.3 Clock Supply

The RSCAN0 clock supply is shown in the following table.

Table 20.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RSCAN0	clk_xincan	CKSCLK_ICANOSC
	clkc	PPLLCLK2
	pclk	CKSCLK_ICAN
	Register access clock	CKSCLK_ICAN

The operating frequency of the RSCAN0 depends on the transfer rate and the number of channels in use. **Table 20.7** shows the range of the frequency.

Table 20.7 Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1M

Condition		Range of Operating Frequency		
Transfer Rate	No. of Channels in Use	pclk	clk_xincan*1, *3	clkc*1, *2
1 Mbps	6ch	pclk ≥ 53MHz	8 MHz ≤ clk_xincan ≤ pclk/2	12.5 MHz ≤ clkc ≤ pclk/2
	5ch	pclk ≥ 46MHz		
	4ch	pclk ≥ 40MHz		
	3ch	pclk ≥ 32MHz		
	2ch	pclk ≥ 26MHz		
	1ch	pclk ≥ 18MHz		
500 kbps	6ch	pclk ≥ 27MHz	4 MHz ≤ clk_xincan ≤ pclk/2	12.5 MHz ≤ clkc ≤ pclk/2
	5ch	pclk ≥ 23MHz		
	4ch	pclk ≥ 20MHz		
	3ch	pclk ≥ 16MHz		
	2ch	pclk ≥ 13MHz		
	1ch	pclk ≥ 8MHz		
125 kbps	6ch	pclk ≥ 8MHz	4 MHz ≤ clk_xincan ≤ pclk/2	12.5 MHz ≤ clkc ≤ pclk/2
	5ch			
	4ch			
	3ch			
	2ch			
	1ch			

Note 1. The DCS bit in RSCAN0GCFG allows the selection of either clk_xincan or clkc. Configure the clock frequency to less than or equal to pclk/2 (up to 40 MHz).

Note 2. Select clk_xincan when pclk < 25 MHz.

Note 3. The maximum frequency of clk_xincan is 24 MHz.

CAUTION

When the RS-CAN module is used in STOP mode, set the MainOSC as the clock source of the RS-CAN module. For details about how to set the clock source, see Section 11.4.3.10, RS-CAN Clock Domains C_ISO_CAN and C_ISO_CANOSC.

20.1.4 Interrupt Requests

RSCAN0 interrupt requests are listed in the following table.

Table 20.8 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
RSCAN0			
INTRCANGERR0	CAN global error interrupt	22	—
INTRCANGRECC0	CAN receive FIFO interrupt	23	—
CAN0			
INTRCANmERR (m = 0)	CAN0 error interrupt	24	—
INTRCANmREC (m = 0)	CAN0 transmit/receive FIFO receive completion interrupt	25	—
INTRCANmTRX (m = 0)	CAN0 transmit interrupt	26	—
CAN1			
INTRCANmERR (m = 1)	CAN1 error interrupt	113	—
INTRCANmREC (m = 1)	CAN1 transmit/receive FIFO receive completion interrupt	114	—
INTRCANmTRX (m = 1)	CAN1 transmit interrupt	115	—
CAN2			
INTRCANmERR (m = 2)	CAN2 error interrupt	217	—
INTRCANmREC (m = 2)	CAN2 transmit/receive FIFO receive completion interrupt	218	—
INTRCANmTRX (m = 2)	CAN2 transmit interrupt	219	—
CAN3			
INTRCANmERR (m = 3)	CAN3 error interrupt	220	—
INTRCANmREC (m = 3)	CAN3 transmit/receive FIFO receive completion interrupt	221	—
INTRCANmTRX (m = 3)	CAN3 transmit interrupt	222	—
CAN4			
INTRCANmERR (m = 4)	CAN4 error interrupt	272	—
INTRCANmREC (m = 4)	CAN4 transmit/receive FIFO receive completion interrupt	273	—
INTRCANmTRX (m = 4)	CAN4 transmit interrupt	274	—
CAN5			
INTRCANmERR (m = 5)	CAN5 error interrupt	287	—
INTRCANmREC (m = 5)	CAN5 transmit/receive FIFO receive completion interrupt	288	—
INTRCANmTRX (m = 5)	CAN5 transmit interrupt	289	—

NOTE

For the wake-up factors from standby mode, see **Section 12.1.2.1, Wake-Up Factors for Stand-By Modes**.

20.1.5 Reset Sources

RSCAN0 reset sources are listed in the following table. RSCAN0 is initialized by these reset sources.

Table 20.9 Reset Sources

Unit Name	Reset Source
RSCAN0	All reset sources (ISORES)

20.1.6 External Input/Output Signals

External input/output signals of RSCAN0 are listed below.

Table 20.10 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
CAN0		
CANmRX (m = 0)	CAN0 receive data input	CAN0RX
CANmTX (m = 0)	CAN0 transmit data output	CAN0TX
CAN1		
CANmRX (m = 1)	CAN1 receive data input	CAN1RX
CANmTX (m = 1)	CAN1 transmit data output	CAN1TX
CAN2		
CANmRX (m = 2)	CAN2 receive data input	CAN2RX
CANmTX (m = 2)	CAN2 transmit data output	CAN2TX
CAN3		
CANmRX (m = 3)	CAN3 receive data input	CAN3RX
CANmTX (m = 3)	CAN3 transmit data output	CAN3TX
CAN4		
CANmRX (m = 4)	CAN4 receive data input	CAN4RX
CANmTX (m = 4)	CAN4 transmit data output	CAN4TX
CAN5		
CANmRX (m = 5)	CAN5 receive data input	CAN5RX
CANmTX (m = 5)	CAN5 transmit data output	CAN5TX

CAUTION

When port P0_0 is used as CAN0TX, port P0_0 (the $\overline{\text{RESETOUT}}$ signal) outputs a low level during a reset and after release from the reset state.

For details, see Section 2.11.1.1, P0_0: $\overline{\text{RESETOUT}}$.

20.2 Overview

20.2.1 Functional Overview

The RH850/F1M incorporates one CAN interface unit (RS-CAN) which consists of six channels (CAN0 to CAN5) of the CAN controller conforming to the ISO11898-1 specifications. **Table 20.11** shows the RS-CAN module specifications. **Figure 20.1** shows the RS-CAN module block diagram.

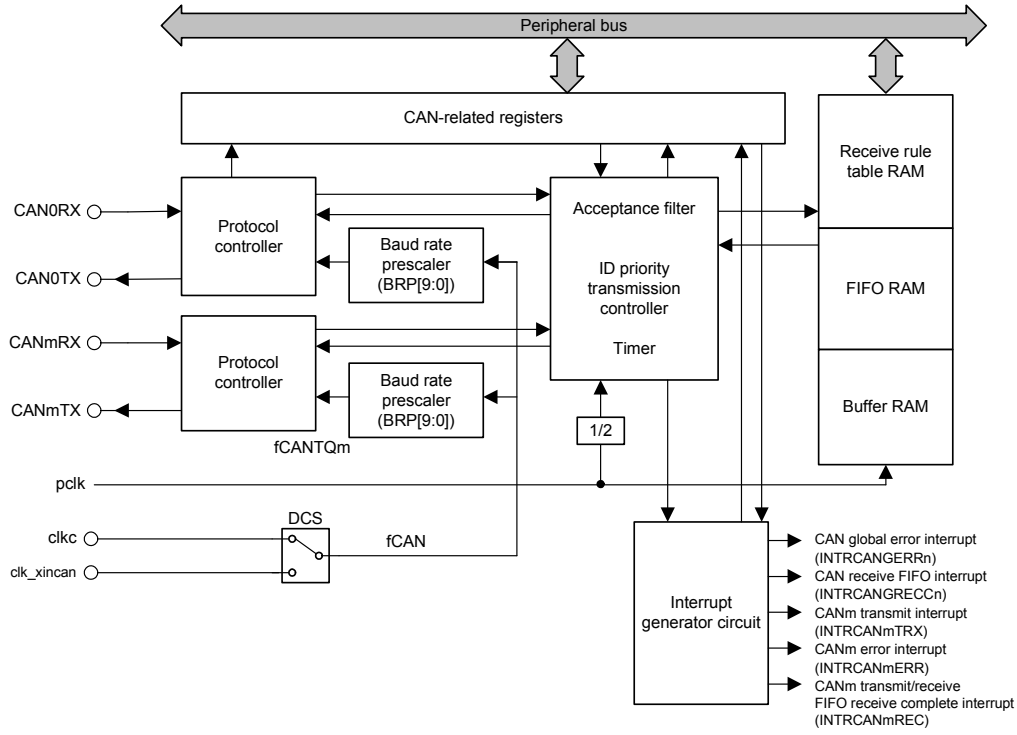
Table 20.11 RS-CAN Module Specifications (1/2)

Item	Specification
Number of channels	6
Protocol	ISO11898-1 compliant
Communication speed	• Maximum 1 Mbps Communication speed (CANm bit time clock) = $\frac{1}{\text{CANm bit time}}$ CANm bit time = CANmTq × Tq count per bit CANmTq = $\frac{\text{(BRP[9:0] bits in the RSCAN0CmCFG register + 1)}}{\text{fCAN}}$ m = 0 to 5 Tq: Time quantum fCAN: Frequency of CAN clock (selected by the DCS bit in the RSCAN0GCFG register)
Buffer	480 buffers in total • Channel-dedicated: 96 buffers (16 buffers × 6 channels) Transmit buffer: 16 buffers per channel Transmit queue: Single queue per channel (shared with the transmit buffer; up to 16 buffers allocatable) • Shared between channels: 384 buffers Receive buffer: 96 buffers Receive FIFO buffer: 8 FIFO buffers (up to 128 buffers allocatable to each) Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 128 buffers allocatable to each) • ECC included
Reception function	• Receives data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be received. • Sets interrupt enable/disable for each FIFO. • Mirror function (reception of messages transmitted from the own CAN node) • Timestamp function (to record message reception time as a 16-bit timer value)
Reception filter function	• Selects receive messages according to 384 receive rules. • Sets the number of receive rules (0 to 128) for each channel. • Acceptance filter processing: Sets ID and mask for each receive rule. • DLC filter processing: Enables DLC filter check for each acceptance rule.
Receive message transfer function	• Routing function Transfers receive messages to arbitrary destinations (can be transferred to up to 8 buffers) Transfer destination: Receive buffer, receive FIFO buffer, and/or transmit/receive FIFO buffer • Labeling function Stores label information together with a message in a receive buffer and FIFO buffer.
Transmission function	• Transmits data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be transmitted. • Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer. • Selects ID priority transmission or transmit buffer number priority transmission. • Transmission can be aborted (possible to confirm with a flag) • One-shot transmission function
Interval transmission function	Transmits messages at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)

Table 20.11 RS-CAN Module Specifications (2/2)

Item	Specification
Transmit queue function	Transmits all stored messages according to the ID priority.
Transmit history function	Stores the history information of transmission-completed messages
Gateway function	Transmits a received message automatically.
Bus off recovery mode selection	Selects the method for returning from bus off state. • ISO11898-1 compliant • Automatic entry to channel halt mode at bus-off entry • Automatic entry to channel halt mode at bus-off end • Transition to channel halt mode by program request • Transition to the error-active state by program request (forcible return from the bus off state)
Error status monitoring	• Monitors CAN protocol errors (stuff error, form error, ACK error, CRC error, bit error, ACK delimiter error, and bus dominant lock). • Detects error status transitions (error warning, error passive, bus off entry, and bus off recovery) • Reads the error counter. • Monitors DLC errors.
Interrupt source	20 sources • Global interrupts (2 sources) Receive FIFO interrupt Global error interrupt • Channel interrupts (3 sources/channel) CANm transmit interrupt (m = 0 to 5) – CANm transmit complete interrupt – CANm transmit abort interrupt – CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode) – CANm transmit history interrupt – CANm transmit queue interrupt CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode) CANm error interrupt
CAN stop mode	Reduces power consumption by stopping clock supply to the RS-CAN module.
CAN clock source	Selects the clkc or the clk_xincan. For the operating frequency range, see **Table 20.7**.
Test function	Test function for user evaluation • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback) • RAM test (read/write test) • Inter-channel communication test

20.2.2 Block Diagram



Note: n = 0, m = 0 to 5
BRP[9:0]: Bits in the RSCAN0CmCFG register
DCS: Bits in the RSCAN0GCFG register
fCANTQm: CANmTq clock
fCAN: CAN clock

Figure 20.1 RS-CAN Module Block Diagram

20.3 Registers

20.3.1 List of Registers

RS-CAN registers are listed in the following table.

For details about <RSCAN0_base>, see **Section 20.1.2, Register Base Address**.

Table 20.12 Registers (1/32)

Module	Register	Symbol	Address
RSCAN0	Channel 0 configuration register	RSCAN0C0CFG	<RSCAN0_base> + 0000_H
RSCAN0	Channel 0 control register	RSCAN0C0CTR	<RSCAN0_base> + 0004_H
RSCAN0	Channel 0 status register	RSCAN0C0STS	<RSCAN0_base> + 0008_H
RSCAN0	Channel 0 error flag register	RSCAN0C0ERFL	<RSCAN0_base> + $000C_H$
RSCAN0	Channel 1 configuration register	RSCAN0C1CFG	<RSCAN0_base> + 0010_H
RSCAN0	Channel 1 control register	RSCAN0C1CTR	<RSCAN0_base> + 0014_H
RSCAN0	Channel 1 status register	RSCAN0C1STS	<RSCAN0_base> + 0018_H
RSCAN0	Channel 1 error flag register	RSCAN0C1ERFL	<RSCAN0_base> + $001C_H$
RSCAN0	Channel 2 configuration register	RSCAN0C2CFG	<RSCAN0_base> + 0020_H
RSCAN0	Channel 2 control register	RSCAN0C2CTR	<RSCAN0_base> + 0024_H
RSCAN0	Channel 2 status register	RSCAN0C2STS	<RSCAN0_base> + 0028_H
RSCAN0	Channel 2 error flag register	RSCAN0C2ERFL	<RSCAN0_base> + $002C_H$
RSCAN0	Channel 3 configuration register	RSCAN0C3CFG	<RSCAN0_base> + 0030_H
RSCAN0	Channel 3 control register	RSCAN0C3CTR	<RSCAN0_base> + 0034_H
RSCAN0	Channel 3 status register	RSCAN0C3STS	<RSCAN0_base> + 0038_H
RSCAN0	Channel 3 error flag register	RSCAN0C3ERFL	<RSCAN0_base> + $003C_H$
RSCAN0	Channel 4 configuration register	RSCAN0C4CFG	<RSCAN0_base> + 0040_H
RSCAN0	Channel 4 control register	RSCAN0C4CTR	<RSCAN0_base> + 0044_H
RSCAN0	Channel 4 status register	RSCAN0C4STS	<RSCAN0_base> + 0048_H
RSCAN0	Channel 4 error flag register	RSCAN0C4ERFL	<RSCAN0_base> + $004C_H$
RSCAN0	Channel 5 configuration register	RSCAN0C5CFG	<RSCAN0_base> + 0050_H
RSCAN0	Channel 5 control register	RSCAN0C5CTR	<RSCAN0_base> + 0054_H
RSCAN0	Channel 5 status register	RSCAN0C5STS	<RSCAN0_base> + 0058_H
RSCAN0	Channel 5 error flag register	RSCAN0C5ERFL	<RSCAN0_base> + $005C_H$
RSCAN0	Global configuration register	RSCAN0GCFG	<RSCAN0_base> + 0084_H
RSCAN0	Global control register	RSCAN0GCTR	<RSCAN0_base> + 0088_H
RSCAN0	Global status register	RSCAN0GSTS	<RSCAN0_base> + $008C_H$
RSCAN0	Global error flag register	RSCAN0GERFL	<RSCAN0_base> + 0090_H
RSCAN0	Global timestamp counter register	RSCAN0GTSC	<RSCAN0_base> + 0094_H
RSCAN0	Receive rule entry control register	RSCAN0GAFLECTR	<RSCAN0_base> + 0098_H
RSCAN0	Receive rule configuration register 0	RSCAN0GAFLCFG0	<RSCAN0_base> + $009C_H$
RSCAN0	Receive rule configuration register 1	RSCAN0GAFLCFG1	<RSCAN0_base> + $00A0_H$
RSCAN0	Receive buffer number register	RSCAN0RMNB	<RSCAN0_base> + $00A4_H$
RSCAN0	Receive buffer new data register 0	RSCAN0RMND0	<RSCAN0_base> + $00A8_H$
RSCAN0	Receive buffer new data register 1	RSCAN0RMND1	<RSCAN0_base> + $00AC_H$
RSCAN0	Receive buffer new data register 2	RSCAN0RMND2	<RSCAN0_base> + $00B0_H$
RSCAN0	Receive FIFO buffer configuration and control register 0	RSCAN0RFCC0	<RSCAN0_base> + $00B8_H$
RSCAN0	Receive FIFO buffer configuration and control register 1	RSCAN0RFCC1	<RSCAN0_base> + $00BC_H$

Table 20.12 Registers (2/32)

Module	Register	Symbol	Address
RSCAN0	Receive FIFO buffer configuration and control register 2	RSCAN0RFCC2	<RSCAN0_base> + $00C0_H$
RSCAN0	Receive FIFO buffer configuration and control register 3	RSCAN0RFCC3	<RSCAN0_base> + $00C4_H$
RSCAN0	Receive FIFO buffer configuration and control register 4	RSCAN0RFCC4	<RSCAN0_base> + $00C8_H$
RSCAN0	Receive FIFO buffer configuration and control register 5	RSCAN0RFCC5	<RSCAN0_base> + $00CC_H$
RSCAN0	Receive FIFO buffer configuration and control register 6	RSCAN0RFCC6	<RSCAN0_base> + $00D0_H$
RSCAN0	Receive FIFO buffer configuration and control register 7	RSCAN0RFCC7	<RSCAN0_base> + $00D4_H$
RSCAN0	Receive FIFO buffer status register 0	RSCAN0RFSTS0	<RSCAN0_base> + $00D8_H$
RSCAN0	Receive FIFO buffer status register 1	RSCAN0RFSTS1	<RSCAN0_base> + $00DC_H$
RSCAN0	Receive FIFO buffer status register 2	RSCAN0RFSTS2	<RSCAN0_base> + $00E0_H$
RSCAN0	Receive FIFO buffer status register 3	RSCAN0RFSTS3	<RSCAN0_base> + $00E4_H$
RSCAN0	Receive FIFO buffer status register 4	RSCAN0RFSTS4	<RSCAN0_base> + $00E8_H$
RSCAN0	Receive FIFO buffer status register 5	RSCAN0RFSTS5	<RSCAN0_base> + $00EC_H$
RSCAN0	Receive FIFO buffer status register 6	RSCAN0RFSTS6	<RSCAN0_base> + $00F0_H$
RSCAN0	Receive FIFO buffer status register 7	RSCAN0RFSTS7	<RSCAN0_base> + $00F4_H$
RSCAN0	Receive FIFO buffer pointer control register 0	RSCAN0RFPCTR0	<RSCAN0_base> + $00F8_H$
RSCAN0	Receive FIFO buffer pointer control register 1	RSCAN0RFPCTR1	<RSCAN0_base> + $00FC_H$
RSCAN0	Receive FIFO buffer pointer control register 2	RSCAN0RFPCTR2	<RSCAN0_base> + 0100_H
RSCAN0	Receive FIFO buffer pointer control register 3	RSCAN0RFPCTR3	<RSCAN0_base> + 0104_H
RSCAN0	Receive FIFO buffer pointer control register 4	RSCAN0RFPCTR4	<RSCAN0_base> + 0108_H
RSCAN0	Receive FIFO buffer pointer control register 5	RSCAN0RFPCTR5	<RSCAN0_base> + $010C_H$
RSCAN0	Receive FIFO buffer pointer control register 6	RSCAN0RFPCTR6	<RSCAN0_base> + 0110_H
RSCAN0	Receive FIFO buffer pointer control register 7	RSCAN0RFPCTR7	<RSCAN0_base> + 0114_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 0	RSCAN0CFCC0	<RSCAN0_base> + 0118_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 1	RSCAN0CFCC1	<RSCAN0_base> + $011C_H$
RSCAN0	Transmit/receive FIFO buffer configuration and control register 2	RSCAN0CFCC2	<RSCAN0_base> + 0120_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 3	RSCAN0CFCC3	<RSCAN0_base> + 0124_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 4	RSCAN0CFCC4	<RSCAN0_base> + 0128_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 5	RSCAN0CFCC5	<RSCAN0_base> + $012C_H$
RSCAN0	Transmit/receive FIFO buffer configuration and control register 6	RSCAN0CFCC6	<RSCAN0_base> + 0130_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 7	RSCAN0CFCC7	<RSCAN0_base> + 0134_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 8	RSCAN0CFCC8	<RSCAN0_base> + 0138_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 9	RSCAN0CFCC9	<RSCAN0_base> + $013C_H$
RSCAN0	Transmit/receive FIFO buffer configuration and control register 10	RSCAN0CFCC10	<RSCAN0_base> + 0140_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 11	RSCAN0CFCC11	<RSCAN0_base> + 0144_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 12	RSCAN0CFCC12	<RSCAN0_base> + 0148_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 13	RSCAN0CFCC13	<RSCAN0_base> + $014C_H$
RSCAN0	Transmit/receive FIFO buffer configuration and control register 14	RSCAN0CFCC14	<RSCAN0_base> + 0150_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 15	RSCAN0CFCC15	<RSCAN0_base> + 0154_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 16	RSCAN0CFCC16	<RSCAN0_base> + 0158_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 17	RSCAN0CFCC17	<RSCAN0_base> + $015C_H$
RSCAN0	Transmit/receive FIFO buffer status register 0	RSCAN0CFSTS0	<RSCAN0_base> + 0178_H
RSCAN0	Transmit/receive FIFO buffer status register 1	RSCAN0CFSTS1	<RSCAN0_base> + $017C_H$
RSCAN0	Transmit/receive FIFO buffer status register 2	RSCAN0CFSTS2	<RSCAN0_base> + 0180_H
RSCAN0	Transmit/receive FIFO buffer status register 3	RSCAN0CFSTS3	<RSCAN0_base> + 0184_H

Table 20.12 Registers (3/32)

Module	Register	Symbol	Address
RSCAN0	Transmit/receive FIFO buffer status register 4	RSCAN0CFSTS4	<RSCAN0_base> + 0188_H
RSCAN0	Transmit/receive FIFO buffer status register 5	RSCAN0CFSTS5	<RSCAN0_base> + $018C_H$
RSCAN0	Transmit/receive FIFO buffer status register 6	RSCAN0CFSTS6	<RSCAN0_base> + 0190_H
RSCAN0	Transmit/receive FIFO buffer status register 7	RSCAN0CFSTS7	<RSCAN0_base> + 0194_H
RSCAN0	Transmit/receive FIFO buffer status register 8	RSCAN0CFSTS8	<RSCAN0_base> + 0198_H
RSCAN0	Transmit/receive FIFO buffer status register 9	RSCAN0CFSTS9	<RSCAN0_base> + $019C_H$
RSCAN0	Transmit/receive FIFO buffer status register 10	RSCAN0CFSTS10	<RSCAN0_base> + $01A0_H$
RSCAN0	Transmit/receive FIFO buffer status register 11	RSCAN0CFSTS11	<RSCAN0_base> + $01A4_H$
RSCAN0	Transmit/receive FIFO buffer status register 12	RSCAN0CFSTS12	<RSCAN0_base> + $01A8_H$
RSCAN0	Transmit/receive FIFO buffer status register 13	RSCAN0CFSTS13	<RSCAN0_base> + $01AC_H$
RSCAN0	Transmit/receive FIFO buffer status register 14	RSCAN0CFSTS14	<RSCAN0_base> + $01B0_H$
RSCAN0	Transmit/receive FIFO buffer status register 15	RSCAN0CFSTS15	<RSCAN0_base> + $01B4_H$
RSCAN0	Transmit/receive FIFO buffer status register 16	RSCAN0CFSTS16	<RSCAN0_base> + $01B8_H$
RSCAN0	Transmit/receive FIFO buffer status register 17	RSCAN0CFSTS17	<RSCAN0_base> + $01BC_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 0	RSCAN0CFPCTR0	<RSCAN0_base> + $01D8_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 1	RSCAN0CFPCTR1	<RSCAN0_base> + $01DC_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 2	RSCAN0CFPCTR2	<RSCAN0_base> + $01E0_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 3	RSCAN0CFPCTR3	<RSCAN0_base> + $01E4_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 4	RSCAN0CFPCTR4	<RSCAN0_base> + $01E8_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 5	RSCAN0CFPCTR5	<RSCAN0_base> + $01EC_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 6	RSCAN0CFPCTR6	<RSCAN0_base> + $01F0_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 7	RSCAN0CFPCTR7	<RSCAN0_base> + $01F4_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 8	RSCAN0CFPCTR8	<RSCAN0_base> + $01F8_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 9	RSCAN0CFPCTR9	<RSCAN0_base> + $01FC_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 10	RSCAN0CFPCTR10	<RSCAN0_base> + 0200_H
RSCAN0	Transmit/receive FIFO buffer pointer control register 11	RSCAN0CFPCTR11	<RSCAN0_base> + 0204_H
RSCAN0	Transmit/receive FIFO buffer pointer control register 12	RSCAN0CFPCTR12	<RSCAN0_base> + 0208_H
RSCAN0	Transmit/receive FIFO buffer pointer control register 13	RSCAN0CFPCTR13	<RSCAN0_base> + $020C_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 14	RSCAN0CFPCTR14	<RSCAN0_base> + 0210_H
RSCAN0	Transmit/receive FIFO buffer pointer control register 15	RSCAN0CFPCTR15	<RSCAN0_base> + 0214_H
RSCAN0	Transmit/receive FIFO buffer pointer control register 16	RSCAN0CFPCTR16	<RSCAN0_base> + 0218_H
RSCAN0	Transmit/receive FIFO buffer pointer control register 17	RSCAN0CFPCTR17	<RSCAN0_base> + $021C_H$
RSCAN0	FIFO empty status register	RSCAN0FESTS	<RSCAN0_base> + 0238_H
RSCAN0	FIFO full status register	RSCAN0FFSTS	<RSCAN0_base> + $023C_H$
RSCAN0	FIFO Msg lost status register	RSCAN0FMSTS	<RSCAN0_base> + 0240_H
RSCAN0	Receive FIFO buffer interrupt flag status register	RSCAN0RFISTS	<RSCAN0_base> + 0244_H
RSCAN0	Transmit/receive FIFO buffer RX interrupt flag status register	RSCAN0CFRISTS	<RSCAN0_base> + 0248_H
RSCAN0	Transmit/receive FIFO buffer TX interrupt flag status register	RSCAN0CFTISTS	<RSCAN0_base> + $024C_H$
RSCAN0	Transmit buffer control register 0	RSCAN0TMC0	<RSCAN0_base> + 0250_H
RSCAN0	Transmit buffer control register 1	RSCAN0TMC1	<RSCAN0_base> + 0251_H
RSCAN0	Transmit buffer control register 2	RSCAN0TMC2	<RSCAN0_base> + 0252_H
RSCAN0	Transmit buffer control register 3	RSCAN0TMC3	<RSCAN0_base> + 0253_H
RSCAN0	Transmit buffer control register 4	RSCAN0TMC4	<RSCAN0_base> + 0254_H
RSCAN0	Transmit buffer control register 5	RSCAN0TMC5	<RSCAN0_base> + 0255_H

Table 20.12 Registers (4/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer control register 6	RSCAN0TMC6	<RSCAN0_base> + 0256_H
RSCAN0	Transmit buffer control register 7	RSCAN0TMC7	<RSCAN0_base> + 0257_H
RSCAN0	Transmit buffer control register 8	RSCAN0TMC8	<RSCAN0_base> + 0258_H
RSCAN0	Transmit buffer control register 9	RSCAN0TMC9	<RSCAN0_base> + 0259_H
RSCAN0	Transmit buffer control register 10	RSCAN0TMC10	<RSCAN0_base> + $025A_H$
RSCAN0	Transmit buffer control register 11	RSCAN0TMC11	<RSCAN0_base> + $025B_H$
RSCAN0	Transmit buffer control register 12	RSCAN0TMC12	<RSCAN0_base> + $025C_H$
RSCAN0	Transmit buffer control register 13	RSCAN0TMC13	<RSCAN0_base> + $025D_H$
RSCAN0	Transmit buffer control register 14	RSCAN0TMC14	<RSCAN0_base> + $025E_H$
RSCAN0	Transmit buffer control register 15	RSCAN0TMC15	<RSCAN0_base> + $025F_H$
RSCAN0	Transmit buffer control register 16	RSCAN0TMC16	<RSCAN0_base> + 0260_H
RSCAN0	Transmit buffer control register 17	RSCAN0TMC17	<RSCAN0_base> + 0261_H
RSCAN0	Transmit buffer control register 18	RSCAN0TMC18	<RSCAN0_base> + 0262_H
RSCAN0	Transmit buffer control register 19	RSCAN0TMC19	<RSCAN0_base> + 0263_H
RSCAN0	Transmit buffer control register 20	RSCAN0TMC20	<RSCAN0_base> + 0264_H
RSCAN0	Transmit buffer control register 21	RSCAN0TMC21	<RSCAN0_base> + 0265_H
RSCAN0	Transmit buffer control register 22	RSCAN0TMC22	<RSCAN0_base> + 0266_H
RSCAN0	Transmit buffer control register 23	RSCAN0TMC23	<RSCAN0_base> + 0267_H
RSCAN0	Transmit buffer control register 24	RSCAN0TMC24	<RSCAN0_base> + 0268_H
RSCAN0	Transmit buffer control register 25	RSCAN0TMC25	<RSCAN0_base> + 0269_H
RSCAN0	Transmit buffer control register 26	RSCAN0TMC26	<RSCAN0_base> + $026A_H$
RSCAN0	Transmit buffer control register 27	RSCAN0TMC27	<RSCAN0_base> + $026B_H$
RSCAN0	Transmit buffer control register 28	RSCAN0TMC28	<RSCAN0_base> + $026C_H$
RSCAN0	Transmit buffer control register 29	RSCAN0TMC29	<RSCAN0_base> + $026D_H$
RSCAN0	Transmit buffer control register 30	RSCAN0TMC30	<RSCAN0_base> + $026E_H$
RSCAN0	Transmit buffer control register 31	RSCAN0TMC31	<RSCAN0_base> + $026F_H$
RSCAN0	Transmit buffer control register 32	RSCAN0TMC32	<RSCAN0_base> + 0270_H
RSCAN0	Transmit buffer control register 33	RSCAN0TMC33	<RSCAN0_base> + 0271_H
RSCAN0	Transmit buffer control register 34	RSCAN0TMC34	<RSCAN0_base> + 0272_H
RSCAN0	Transmit buffer control register 35	RSCAN0TMC35	<RSCAN0_base> + 0273_H
RSCAN0	Transmit buffer control register 36	RSCAN0TMC36	<RSCAN0_base> + 0274_H
RSCAN0	Transmit buffer control register 37	RSCAN0TMC37	<RSCAN0_base> + 0275_H
RSCAN0	Transmit buffer control register 38	RSCAN0TMC38	<RSCAN0_base> + 0276_H
RSCAN0	Transmit buffer control register 39	RSCAN0TMC39	<RSCAN0_base> + 0277_H
RSCAN0	Transmit buffer control register 40	RSCAN0TMC40	<RSCAN0_base> + 0278_H
RSCAN0	Transmit buffer control register 41	RSCAN0TMC41	<RSCAN0_base> + 0279_H
RSCAN0	Transmit buffer control register 42	RSCAN0TMC42	<RSCAN0_base> + $027A_H$
RSCAN0	Transmit buffer control register 43	RSCAN0TMC43	<RSCAN0_base> + $027B_H$
RSCAN0	Transmit buffer control register 44	RSCAN0TMC44	<RSCAN0_base> + $027C_H$
RSCAN0	Transmit buffer control register 45	RSCAN0TMC45	<RSCAN0_base> + $027D_H$
RSCAN0	Transmit buffer control register 46	RSCAN0TMC46	<RSCAN0_base> + $027E_H$
RSCAN0	Transmit buffer control register 47	RSCAN0TMC47	<RSCAN0_base> + $027F_H$
RSCAN0	Transmit buffer control register 48	RSCAN0TMC48	<RSCAN0_base> + 0280_H
RSCAN0	Transmit buffer control register 49	RSCAN0TMC49	<RSCAN0_base> + 0281_H

Table 20.12 Registers (5/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer control register 50	RSCAN0TMC50	<RSCAN0_base> + 0282_H
RSCAN0	Transmit buffer control register 51	RSCAN0TMC51	<RSCAN0_base> + 0283_H
RSCAN0	Transmit buffer control register 52	RSCAN0TMC52	<RSCAN0_base> + 0284_H
RSCAN0	Transmit buffer control register 53	RSCAN0TMC53	<RSCAN0_base> + 0285_H
RSCAN0	Transmit buffer control register 54	RSCAN0TMC54	<RSCAN0_base> + 0286_H
RSCAN0	Transmit buffer control register 55	RSCAN0TMC55	<RSCAN0_base> + 0287_H
RSCAN0	Transmit buffer control register 56	RSCAN0TMC56	<RSCAN0_base> + 0288_H
RSCAN0	Transmit buffer control register 57	RSCAN0TMC57	<RSCAN0_base> + 0289_H
RSCAN0	Transmit buffer control register 58	RSCAN0TMC58	<RSCAN0_base> + $028A_H$
RSCAN0	Transmit buffer control register 59	RSCAN0TMC59	<RSCAN0_base> + $028B_H$
RSCAN0	Transmit buffer control register 60	RSCAN0TMC60	<RSCAN0_base> + $028C_H$
RSCAN0	Transmit buffer control register 61	RSCAN0TMC61	<RSCAN0_base> + $028D_H$
RSCAN0	Transmit buffer control register 62	RSCAN0TMC62	<RSCAN0_base> + $028E_H$
RSCAN0	Transmit buffer control register 63	RSCAN0TMC63	<RSCAN0_base> + $028F_H$
RSCAN0	Transmit buffer control register 64	RSCAN0TMC64	<RSCAN0_base> + 0290_H
RSCAN0	Transmit buffer control register 65	RSCAN0TMC65	<RSCAN0_base> + 0291_H
RSCAN0	Transmit buffer control register 66	RSCAN0TMC66	<RSCAN0_base> + 0292_H
RSCAN0	Transmit buffer control register 67	RSCAN0TMC67	<RSCAN0_base> + 0293_H
RSCAN0	Transmit buffer control register 68	RSCAN0TMC68	<RSCAN0_base> + 0294_H
RSCAN0	Transmit buffer control register 69	RSCAN0TMC69	<RSCAN0_base> + 0295_H
RSCAN0	Transmit buffer control register 70	RSCAN0TMC70	<RSCAN0_base> + 0296_H
RSCAN0	Transmit buffer control register 71	RSCAN0TMC71	<RSCAN0_base> + 0297_H
RSCAN0	Transmit buffer control register 72	RSCAN0TMC72	<RSCAN0_base> + 0298_H
RSCAN0	Transmit buffer control register 73	RSCAN0TMC73	<RSCAN0_base> + 0299_H
RSCAN0	Transmit buffer control register 74	RSCAN0TMC74	<RSCAN0_base> + $029A_H$
RSCAN0	Transmit buffer control register 75	RSCAN0TMC75	<RSCAN0_base> + $029B_H$
RSCAN0	Transmit buffer control register 76	RSCAN0TMC76	<RSCAN0_base> + $029C_H$
RSCAN0	Transmit buffer control register 77	RSCAN0TMC77	<RSCAN0_base> + $029D_H$
RSCAN0	Transmit buffer control register 78	RSCAN0TMC78	<RSCAN0_base> + $029E_H$
RSCAN0	Transmit buffer control register 79	RSCAN0TMC79	<RSCAN0_base> + $029F_H$
RSCAN0	Transmit buffer control register 80	RSCAN0TMC80	<RSCAN0_base> + $02A0_H$
RSCAN0	Transmit buffer control register 81	RSCAN0TMC81	<RSCAN0_base> + $02A1_H$
RSCAN0	Transmit buffer control register 82	RSCAN0TMC82	<RSCAN0_base> + $02A2_H$
RSCAN0	Transmit buffer control register 83	RSCAN0TMC83	<RSCAN0_base> + $02A3_H$
RSCAN0	Transmit buffer control register 84	RSCAN0TMC84	<RSCAN0_base> + $02A4_H$
RSCAN0	Transmit buffer control register 85	RSCAN0TMC85	<RSCAN0_base> + $02A5_H$
RSCAN0	Transmit buffer control register 86	RSCAN0TMC86	<RSCAN0_base> + $02A6_H$
RSCAN0	Transmit buffer control register 87	RSCAN0TMC87	<RSCAN0_base> + $02A7_H$
RSCAN0	Transmit buffer control register 88	RSCAN0TMC88	<RSCAN0_base> + $02A8_H$
RSCAN0	Transmit buffer control register 89	RSCAN0TMC89	<RSCAN0_base> + $02A9_H$
RSCAN0	Transmit buffer control register 90	RSCAN0TMC90	<RSCAN0_base> + $02AA_H$
RSCAN0	Transmit buffer control register 91	RSCAN0TMC91	<RSCAN0_base> + $02AB_H$
RSCAN0	Transmit buffer control register 92	RSCAN0TMC92	<RSCAN0_base> + $02AC_H$
RSCAN0	Transmit buffer control register 93	RSCAN0TMC93	<RSCAN0_base> + $02AD_H$

Table 20.12 Registers (6/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer control register 94	RSCAN0TMC94	<RSCAN0_base> + $02AE_H$
RSCAN0	Transmit buffer control register 95	RSCAN0TMC95	<RSCAN0_base> + $02AF_H$
RSCAN0	Transmit buffer status register 0	RSCAN0TMSTS0	<RSCAN0_base> + $02D0_H$
RSCAN0	Transmit buffer status register 1	RSCAN0TMSTS1	<RSCAN0_base> + $02D1_H$
RSCAN0	Transmit buffer status register 2	RSCAN0TMSTS2	<RSCAN0_base> + $02D2_H$
RSCAN0	Transmit buffer status register 3	RSCAN0TMSTS3	<RSCAN0_base> + $02D3_H$
RSCAN0	Transmit buffer status register 4	RSCAN0TMSTS4	<RSCAN0_base> + $02D4_H$
RSCAN0	Transmit buffer status register 5	RSCAN0TMSTS5	<RSCAN0_base> + $02D5_H$
RSCAN0	Transmit buffer status register 6	RSCAN0TMSTS6	<RSCAN0_base> + $02D6_H$
RSCAN0	Transmit buffer status register 7	RSCAN0TMSTS7	<RSCAN0_base> + $02D7_H$
RSCAN0	Transmit buffer status register 8	RSCAN0TMSTS8	<RSCAN0_base> + $02D8_H$
RSCAN0	Transmit buffer status register 9	RSCAN0TMSTS9	<RSCAN0_base> + $02D9_H$
RSCAN0	Transmit buffer status register 10	RSCAN0TMSTS10	<RSCAN0_base> + $02DA_H$
RSCAN0	Transmit buffer status register 11	RSCAN0TMSTS11	<RSCAN0_base> + $02DB_H$
RSCAN0	Transmit buffer status register 12	RSCAN0TMSTS12	<RSCAN0_base> + $02DC_H$
RSCAN0	Transmit buffer status register 13	RSCAN0TMSTS13	<RSCAN0_base> + $02DD_H$
RSCAN0	Transmit buffer status register 14	RSCAN0TMSTS14	<RSCAN0_base> + $02DE_H$
RSCAN0	Transmit buffer status register 15	RSCAN0TMSTS15	<RSCAN0_base> + $02DF_H$
RSCAN0	Transmit buffer status register 16	RSCAN0TMSTS16	<RSCAN0_base> + $02E0_H$
RSCAN0	Transmit buffer status register 17	RSCAN0TMSTS17	<RSCAN0_base> + $02E1_H$
RSCAN0	Transmit buffer status register 18	RSCAN0TMSTS18	<RSCAN0_base> + $02E2_H$
RSCAN0	Transmit buffer status register 19	RSCAN0TMSTS19	<RSCAN0_base> + $02E3_H$
RSCAN0	Transmit buffer status register 20	RSCAN0TMSTS20	<RSCAN0_base> + $02E4_H$
RSCAN0	Transmit buffer status register 21	RSCAN0TMSTS21	<RSCAN0_base> + $02E5_H$
RSCAN0	Transmit buffer status register 22	RSCAN0TMSTS22	<RSCAN0_base> + $02E6_H$
RSCAN0	Transmit buffer status register 23	RSCAN0TMSTS23	<RSCAN0_base> + $02E7_H$
RSCAN0	Transmit buffer status register 24	RSCAN0TMSTS24	<RSCAN0_base> + $02E8_H$
RSCAN0	Transmit buffer status register 25	RSCAN0TMSTS25	<RSCAN0_base> + $02E9_H$
RSCAN0	Transmit buffer status register 26	RSCAN0TMSTS26	<RSCAN0_base> + $02EA_H$
RSCAN0	Transmit buffer status register 27	RSCAN0TMSTS27	<RSCAN0_base> + $02EB_H$
RSCAN0	Transmit buffer status register 28	RSCAN0TMSTS28	<RSCAN0_base> + $02EC_H$
RSCAN0	Transmit buffer status register 29	RSCAN0TMSTS29	<RSCAN0_base> + $02ED_H$
RSCAN0	Transmit buffer status register 30	RSCAN0TMSTS30	<RSCAN0_base> + $02EE_H$
RSCAN0	Transmit buffer status register 31	RSCAN0TMSTS31	<RSCAN0_base> + $02EF_H$
RSCAN0	Transmit buffer status register 32	RSCAN0TMSTS32	<RSCAN0_base> + $02F0_H$
RSCAN0	Transmit buffer status register 33	RSCAN0TMSTS33	<RSCAN0_base> + $02F1_H$
RSCAN0	Transmit buffer status register 34	RSCAN0TMSTS34	<RSCAN0_base> + $02F2_H$
RSCAN0	Transmit buffer status register 35	RSCAN0TMSTS35	<RSCAN0_base> + $02F3_H$
RSCAN0	Transmit buffer status register 36	RSCAN0TMSTS36	<RSCAN0_base> + $02F4_H$
RSCAN0	Transmit buffer status register 37	RSCAN0TMSTS37	<RSCAN0_base> + $02F5_H$
RSCAN0	Transmit buffer status register 38	RSCAN0TMSTS38	<RSCAN0_base> + $02F6_H$
RSCAN0	Transmit buffer status register 39	RSCAN0TMSTS39	<RSCAN0_base> + $02F7_H$
RSCAN0	Transmit buffer status register 40	RSCAN0TMSTS40	<RSCAN0_base> + $02F8_H$
RSCAN0	Transmit buffer status register 41	RSCAN0TMSTS41	<RSCAN0_base> + $02F9_H$

Table 20.12 Registers (7/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer status register 42	RSCAN0TMSTS42	<RSCAN0_base> + $02FA_H$
RSCAN0	Transmit buffer status register 43	RSCAN0TMSTS43	<RSCAN0_base> + $02FB_H$
RSCAN0	Transmit buffer status register 44	RSCAN0TMSTS44	<RSCAN0_base> + $02FC_H$
RSCAN0	Transmit buffer status register 45	RSCAN0TMSTS45	<RSCAN0_base> + $02FD_H$
RSCAN0	Transmit buffer status register 46	RSCAN0TMSTS46	<RSCAN0_base> + $02FE_H$
RSCAN0	Transmit buffer status register 47	RSCAN0TMSTS47	<RSCAN0_base> + $02FF_H$
RSCAN0	Transmit buffer status register 48	RSCAN0TMSTS48	<RSCAN0_base> + 0300_H
RSCAN0	Transmit buffer status register 49	RSCAN0TMSTS49	<RSCAN0_base> + 0301_H
RSCAN0	Transmit buffer status register 50	RSCAN0TMSTS50	<RSCAN0_base> + 0302_H
RSCAN0	Transmit buffer status register 51	RSCAN0TMSTS51	<RSCAN0_base> + 0303_H
RSCAN0	Transmit buffer status register 52	RSCAN0TMSTS52	<RSCAN0_base> + 0304_H
RSCAN0	Transmit buffer status register 53	RSCAN0TMSTS53	<RSCAN0_base> + 0305_H
RSCAN0	Transmit buffer status register 54	RSCAN0TMSTS54	<RSCAN0_base> + 0306_H
RSCAN0	Transmit buffer status register 55	RSCAN0TMSTS55	<RSCAN0_base> + 0307_H
RSCAN0	Transmit buffer status register 56	RSCAN0TMSTS56	<RSCAN0_base> + 0308_H
RSCAN0	Transmit buffer status register 57	RSCAN0TMSTS57	<RSCAN0_base> + 0309_H
RSCAN0	Transmit buffer status register 58	RSCAN0TMSTS58	<RSCAN0_base> + $030A_H$
RSCAN0	Transmit buffer status register 59	RSCAN0TMSTS59	<RSCAN0_base> + $030B_H$
RSCAN0	Transmit buffer status register 60	RSCAN0TMSTS60	<RSCAN0_base> + $030C_H$
RSCAN0	Transmit buffer status register 61	RSCAN0TMSTS61	<RSCAN0_base> + $030D_H$
RSCAN0	Transmit buffer status register 62	RSCAN0TMSTS62	<RSCAN0_base> + $030E_H$
RSCAN0	Transmit buffer status register 63	RSCAN0TMSTS63	<RSCAN0_base> + $030F_H$
RSCAN0	Transmit buffer status register 64	RSCAN0TMSTS64	<RSCAN0_base> + 0310_H
RSCAN0	Transmit buffer status register 65	RSCAN0TMSTS65	<RSCAN0_base> + 0311_H
RSCAN0	Transmit buffer status register 66	RSCAN0TMSTS66	<RSCAN0_base> + 0312_H
RSCAN0	Transmit buffer status register 67	RSCAN0TMSTS67	<RSCAN0_base> + 0313_H
RSCAN0	Transmit buffer status register 68	RSCAN0TMSTS68	<RSCAN0_base> + 0314_H
RSCAN0	Transmit buffer status register 69	RSCAN0TMSTS69	<RSCAN0_base> + 0315_H
RSCAN0	Transmit buffer status register 70	RSCAN0TMSTS70	<RSCAN0_base> + 0316_H
RSCAN0	Transmit buffer status register 71	RSCAN0TMSTS71	<RSCAN0_base> + 0317_H
RSCAN0	Transmit buffer status register 72	RSCAN0TMSTS72	<RSCAN0_base> + 0318_H
RSCAN0	Transmit buffer status register 73	RSCAN0TMSTS73	<RSCAN0_base> + 0319_H
RSCAN0	Transmit buffer status register 74	RSCAN0TMSTS74	<RSCAN0_base> + $031A_H$
RSCAN0	Transmit buffer status register 75	RSCAN0TMSTS75	<RSCAN0_base> + $031B_H$
RSCAN0	Transmit buffer status register 76	RSCAN0TMSTS76	<RSCAN0_base> + $031C_H$
RSCAN0	Transmit buffer status register 77	RSCAN0TMSTS77	<RSCAN0_base> + $031D_H$
RSCAN0	Transmit buffer status register 78	RSCAN0TMSTS78	<RSCAN0_base> + $031E_H$
RSCAN0	Transmit buffer status register 79	RSCAN0TMSTS79	<RSCAN0_base> + $031F_H$
RSCAN0	Transmit buffer status register 80	RSCAN0TMSTS80	<RSCAN0_base> + 0320_H
RSCAN0	Transmit buffer status register 81	RSCAN0TMSTS81	<RSCAN0_base> + 0321_H
RSCAN0	Transmit buffer status register 82	RSCAN0TMSTS82	<RSCAN0_base> + 0322_H
RSCAN0	Transmit buffer status register 83	RSCAN0TMSTS83	<RSCAN0_base> + 0323_H
RSCAN0	Transmit buffer status register 84	RSCAN0TMSTS84	<RSCAN0_base> + 0324_H
RSCAN0	Transmit buffer status register 85	RSCAN0TMSTS85	<RSCAN0_base> + 0325_H

Table 20.12 Registers (8/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer status register 86	RSCAN0TMSTS86	<RSCAN0_base> + 0326_H
RSCAN0	Transmit buffer status register 87	RSCAN0TMSTS87	<RSCAN0_base> + 0327_H
RSCAN0	Transmit buffer status register 88	RSCAN0TMSTS88	<RSCAN0_base> + 0328_H
RSCAN0	Transmit buffer status register 89	RSCAN0TMSTS89	<RSCAN0_base> + 0329_H
RSCAN0	Transmit buffer status register 90	RSCAN0TMSTS90	<RSCAN0_base> + $032A_H$
RSCAN0	Transmit buffer status register 91	RSCAN0TMSTS91	<RSCAN0_base> + $032B_H$
RSCAN0	Transmit buffer status register 92	RSCAN0TMSTS92	<RSCAN0_base> + $032C_H$
RSCAN0	Transmit buffer status register 93	RSCAN0TMSTS93	<RSCAN0_base> + $032D_H$
RSCAN0	Transmit buffer status register 94	RSCAN0TMSTS94	<RSCAN0_base> + $032E_H$
RSCAN0	Transmit buffer status register 95	RSCAN0TMSTS95	<RSCAN0_base> + $032F_H$
RSCAN0	Transmit buffer transmit request status register 0	RSCAN0TMTRSTS0	<RSCAN0_base> + 0350_H
RSCAN0	Transmit buffer transmit request status register 1	RSCAN0TMTRSTS1	<RSCAN0_base> + 0354_H
RSCAN0	Transmit buffer transmit request status register 2	RSCAN0TMTRSTS2	<RSCAN0_base> + 0358_H
RSCAN0	Transmit buffer transmit abort request status register 0	RSCAN0TMTARSTS0	<RSCAN0_base> + 0360_H
RSCAN0	Transmit buffer transmit abort request status register 1	RSCAN0TMTARSTS1	<RSCAN0_base> + 0364_H
RSCAN0	Transmit buffer transmit abort request status register 2	RSCAN0TMTARSTS2	<RSCAN0_base> + 0368_H
RSCAN0	Transmit buffer transmit complete status register 0	RSCAN0TMTCSTS0	<RSCAN0_base> + 0370_H
RSCAN0	Transmit buffer transmit complete status register 1	RSCAN0TMTCSTS1	<RSCAN0_base> + 0374_H
RSCAN0	Transmit buffer transmit complete status register 2	RSCAN0TMTCSTS2	<RSCAN0_base> + 0378_H
RSCAN0	Transmit buffer transmit abort status register 0	RSCAN0TMTASTS0	<RSCAN0_base> + 0380_H
RSCAN0	Transmit buffer transmit abort status register 1	RSCAN0TMTASTS1	<RSCAN0_base> + 0384_H
RSCAN0	Transmit buffer transmit abort status register 2	RSCAN0TMTASTS2	<RSCAN0_base> + 0388_H
RSCAN0	Transmit buffer interrupt enable configuration register 0	RSCAN0TMIEC0	<RSCAN0_base> + 0390_H
RSCAN0	Transmit buffer interrupt enable configuration register 1	RSCAN0TMIEC1	<RSCAN0_base> + 0394_H
RSCAN0	Transmit buffer interrupt enable configuration register 2	RSCAN0TMIEC2	<RSCAN0_base> + 0398_H
RSCAN0	Transmit queue configuration and control register 0	RSCAN0TXQCC0	<RSCAN0_base> + $03A0_H$
RSCAN0	Transmit queue configuration and control register 1	RSCAN0TXQCC1	<RSCAN0_base> + $03A4_H$
RSCAN0	Transmit queue configuration and control register 2	RSCAN0TXQCC2	<RSCAN0_base> + $03A8_H$
RSCAN0	Transmit queue configuration and control register 3	RSCAN0TXQCC3	<RSCAN0_base> + $03AC_H$
RSCAN0	Transmit queue configuration and control register 4	RSCAN0TXQCC4	<RSCAN0_base> + $03B0_H$
RSCAN0	Transmit queue configuration and control register 5	RSCAN0TXQCC5	<RSCAN0_base> + $03B4_H$
RSCAN0	Transmit queue status register 0	RSCAN0TXQSTS0	<RSCAN0_base> + $03C0_H$
RSCAN0	Transmit queue status register 1	RSCAN0TXQSTS1	<RSCAN0_base> + $03C4_H$
RSCAN0	Transmit queue status register 2	RSCAN0TXQSTS2	<RSCAN0_base> + $03C8_H$
RSCAN0	Transmit queue status register 3	RSCAN0TXQSTS3	<RSCAN0_base> + $03CC_H$
RSCAN0	Transmit queue status register 4	RSCAN0TXQSTS4	<RSCAN0_base> + $03D0_H$
RSCAN0	Transmit queue status register 5	RSCAN0TXQSTS5	<RSCAN0_base> + $03D4_H$
RSCAN0	Transmit queue pointer control register 0	RSCAN0TXQPCTR0	<RSCAN0_base> + $03E0_H$
RSCAN0	Transmit queue pointer control register 1	RSCAN0TXQPCTR1	<RSCAN0_base> + $03E4_H$
RSCAN0	Transmit queue pointer control register 2	RSCAN0TXQPCTR2	<RSCAN0_base> + $03E8_H$
RSCAN0	Transmit queue pointer control register 3	RSCAN0TXQPCTR3	<RSCAN0_base> + $03EC_H$
RSCAN0	Transmit queue pointer control register 4	RSCAN0TXQPCTR4	<RSCAN0_base> + $03F0_H$
RSCAN0	Transmit queue pointer control register 5	RSCAN0TXQPCTR5	<RSCAN0_base> + $03F4_H$
RSCAN0	Transmit history configuration and control register 0	RSCAN0THLCC0	<RSCAN0_base> + 0400_H

Table 20.12 Registers (9/32)

Module	Register	Symbol	Address
RSCAN0	Transmit history configuration and control register 1	RSCAN0THLCC1	<RSCAN0_base> + 0404_H
RSCAN0	Transmit history configuration and control register 2	RSCAN0THLCC2	<RSCAN0_base> + 0408_H
RSCAN0	Transmit history configuration and control register 3	RSCAN0THLCC3	<RSCAN0_base> + $040C_H$
RSCAN0	Transmit history configuration and control register 4	RSCAN0THLCC4	<RSCAN0_base> + 0410_H
RSCAN0	Transmit history configuration and control register 5	RSCAN0THLCC5	<RSCAN0_base> + 0414_H
RSCAN0	Transmit history status register 0	RSCAN0THLSTS0	<RSCAN0_base> + 0420_H
RSCAN0	Transmit history status register 1	RSCAN0THLSTS1	<RSCAN0_base> + 0424_H
RSCAN0	Transmit history status register 2	RSCAN0THLSTS2	<RSCAN0_base> + 0428_H
RSCAN0	Transmit history status register 3	RSCAN0THLSTS3	<RSCAN0_base> + $042C_H$
RSCAN0	Transmit history status register 4	RSCAN0THLSTS4	<RSCAN0_base> + 0430_H
RSCAN0	Transmit history status register 5	RSCAN0THLSTS5	<RSCAN0_base> + 0434_H
RSCAN0	Transmit history pointer control register 0	RSCAN0THLPCTR0	<RSCAN0_base> + 0440_H
RSCAN0	Transmit history pointer control register 1	RSCAN0THLPCTR1	<RSCAN0_base> + 0444_H
RSCAN0	Transmit history pointer control register 2	RSCAN0THLPCTR2	<RSCAN0_base> + 0448_H
RSCAN0	Transmit history pointer control register 3	RSCAN0THLPCTR3	<RSCAN0_base> + $044C_H$
RSCAN0	Transmit history pointer control register 4	RSCAN0THLPCTR4	<RSCAN0_base> + 0450_H
RSCAN0	Transmit history pointer control register 5	RSCAN0THLPCTR5	<RSCAN0_base> + 0454_H
RSCAN0	Global TX interrupt status register 0	RSCAN0GTINTSTS0	<RSCAN0_base> + 0460_H
RSCAN0	Global TX interrupt status register 1	RSCAN0GTINTSTS1	<RSCAN0_base> + 0464_H
RSCAN0	Global test configuration register	RSCAN0GTSTCFG	<RSCAN0_base> + 0468_H
RSCAN0	Global test control register	RSCAN0GTSTCTR	<RSCAN0_base> + $046C_H$
RSCAN0	Global lock key register	RSCAN0GLOCKK	<RSCAN0_base> + $047C_H$
RSCAN0	Receive rule ID register 0	RSCAN0GAFLID0	<RSCAN0_base> + 0500_H
RSCAN0	Receive rule mask register 0	RSCAN0GAFLM0	<RSCAN0_base> + 0504_H
RSCAN0	Receive rule pointer 0 register 0	RSCAN0GAFLP00	<RSCAN0_base> + 0508_H
RSCAN0	Receive rule pointer 1 register 0	RSCAN0GAFLP10	<RSCAN0_base> + $050C_H$
RSCAN0	Receive rule ID register 1	RSCAN0GAFLID1	<RSCAN0_base> + 0510_H
RSCAN0	Receive rule mask register 1	RSCAN0GAFLM1	<RSCAN0_base> + 0514_H
RSCAN0	Receive rule pointer 0 register 1	RSCAN0GAFLP01	<RSCAN0_base> + 0518_H
RSCAN0	Receive rule pointer 1 register 1	RSCAN0GAFLP11	<RSCAN0_base> + $051C_H$
RSCAN0	Receive rule ID register 2	RSCAN0GAFLID2	<RSCAN0_base> + 0520_H
RSCAN0	Receive rule mask register 2	RSCAN0GAFLM2	<RSCAN0_base> + 0524_H
RSCAN0	Receive rule pointer 0 register 2	RSCAN0GAFLP02	<RSCAN0_base> + 0528_H
RSCAN0	Receive rule pointer 1 register 2	RSCAN0GAFLP12	<RSCAN0_base> + $052C_H$
RSCAN0	Receive rule ID register 3	RSCAN0GAFLID3	<RSCAN0_base> + 0530_H
RSCAN0	Receive rule mask register 3	RSCAN0GAFLM3	<RSCAN0_base> + 0534_H
RSCAN0	Receive rule pointer 0 register 3	RSCAN0GAFLP03	<RSCAN0_base> + 0538_H
RSCAN0	Receive rule pointer 1 register 3	RSCAN0GAFLP13	<RSCAN0_base> + $053C_H$
RSCAN0	Receive rule ID register 4	RSCAN0GAFLID4	<RSCAN0_base> + 0540_H
RSCAN0	Receive rule mask register 4	RSCAN0GAFLM4	<RSCAN0_base> + 0544_H
RSCAN0	Receive rule pointer 0 register 4	RSCAN0GAFLP04	<RSCAN0_base> + 0548_H
RSCAN0	Receive rule pointer 1 register 4	RSCAN0GAFLP14	<RSCAN0_base> + $054C_H$
RSCAN0	Receive rule ID register 5	RSCAN0GAFLID5	<RSCAN0_base> + 0550_H
RSCAN0	Receive rule mask register 5	RSCAN0GAFLM5	<RSCAN0_base> + 0554_H

Table 20.12 Registers (10/32)

Module	Register	Symbol	Address
RSCAN0	Receive rule pointer 0 register 5	RSCAN0GAFLP05	<RSCAN0_base> + 0558_H
RSCAN0	Receive rule pointer 1 register 5	RSCAN0GAFLP15	<RSCAN0_base> + $055C_H$
RSCAN0	Receive rule ID register 6	RSCAN0GAFLID6	<RSCAN0_base> + 0560_H
RSCAN0	Receive rule mask register 6	RSCAN0GAFLM6	<RSCAN0_base> + 0564_H
RSCAN0	Receive rule pointer 0 register 6	RSCAN0GAFLP06	<RSCAN0_base> + 0568_H
RSCAN0	Receive rule pointer 1 register 6	RSCAN0GAFLP16	<RSCAN0_base> + $056C_H$
RSCAN0	Receive rule ID register 7	RSCAN0GAFLID7	<RSCAN0_base> + 0570_H
RSCAN0	Receive rule mask register 7	RSCAN0GAFLM7	<RSCAN0_base> + 0574_H
RSCAN0	Receive rule pointer 0 register 7	RSCAN0GAFLP07	<RSCAN0_base> + 0578_H
RSCAN0	Receive rule pointer 1 register 7	RSCAN0GAFLP17	<RSCAN0_base> + $057C_H$
RSCAN0	Receive rule ID register 8	RSCAN0GAFLID8	<RSCAN0_base> + 0580_H
RSCAN0	Receive rule mask register 8	RSCAN0GAFLM8	<RSCAN0_base> + 0584_H
RSCAN0	Receive rule pointer 0 register 8	RSCAN0GAFLP08	<RSCAN0_base> + 0588_H
RSCAN0	Receive rule pointer 1 register 8	RSCAN0GAFLP18	<RSCAN0_base> + $058C_H$
RSCAN0	Receive rule ID register 9	RSCAN0GAFLID9	<RSCAN0_base> + 0590_H
RSCAN0	Receive rule mask register 9	RSCAN0GAFLM9	<RSCAN0_base> + 0594_H
RSCAN0	Receive rule pointer 0 register 9	RSCAN0GAFLP09	<RSCAN0_base> + 0598_H
RSCAN0	Receive rule pointer 1 register 9	RSCAN0GAFLP19	<RSCAN0_base> + $059C_H$
RSCAN0	Receive rule ID register 10	RSCAN0GAFLID10	<RSCAN0_base> + $05A0_H$
RSCAN0	Receive rule mask register 10	RSCAN0GAFLM10	<RSCAN0_base> + $05A4_H$
RSCAN0	Receive rule pointer 0 register 10	RSCAN0GAFLP010	<RSCAN0_base> + $05A8_H$
RSCAN0	Receive rule pointer 1 register 10	RSCAN0GAFLP110	<RSCAN0_base> + $05AC_H$
RSCAN0	Receive rule ID register 11	RSCAN0GAFLID11	<RSCAN0_base> + $05B0_H$
RSCAN0	Receive rule mask register 11	RSCAN0GAFLM11	<RSCAN0_base> + $05B4_H$
RSCAN0	Receive rule pointer 0 register 11	RSCAN0GAFLP011	<RSCAN0_base> + $05B8_H$
RSCAN0	Receive rule pointer 1 register 11	RSCAN0GAFLP111	<RSCAN0_base> + $05BC_H$
RSCAN0	Receive rule ID register 12	RSCAN0GAFLID12	<RSCAN0_base> + $05C0_H$
RSCAN0	Receive rule mask register 12	RSCAN0GAFLM12	<RSCAN0_base> + $05C4_H$
RSCAN0	Receive rule pointer 0 register 12	RSCAN0GAFLP012	<RSCAN0_base> + $05C8_H$
RSCAN0	Receive rule pointer 1 register 12	RSCAN0GAFLP112	<RSCAN0_base> + $05CC_H$
RSCAN0	Receive rule ID register 13	RSCAN0GAFLID13	<RSCAN0_base> + $05D0_H$
RSCAN0	Receive rule mask register 13	RSCAN0GAFLM13	<RSCAN0_base> + $05D4_H$
RSCAN0	Receive rule pointer 0 register 13	RSCAN0GAFLP013	<RSCAN0_base> + $05D8_H$
RSCAN0	Receive rule pointer 1 register 13	RSCAN0GAFLP113	<RSCAN0_base> + $05DC_H$
RSCAN0	Receive rule ID register 14	RSCAN0GAFLID14	<RSCAN0_base> + $05E0_H$
RSCAN0	Receive rule mask register 14	RSCAN0GAFLM14	<RSCAN0_base> + $05E4_H$
RSCAN0	Receive rule pointer 0 register 14	RSCAN0GAFLP014	<RSCAN0_base> + $05E8_H$
RSCAN0	Receive rule pointer 1 register 14	RSCAN0GAFLP114	<RSCAN0_base> + $05EC_H$
RSCAN0	Receive rule ID register 15	RSCAN0GAFLID15	<RSCAN0_base> + $05F0_H$
RSCAN0	Receive rule mask register 15	RSCAN0GAFLM15	<RSCAN0_base> + $05F4_H$
RSCAN0	Receive rule pointer 0 register 15	RSCAN0GAFLP015	<RSCAN0_base> + $05F8_H$
RSCAN0	Receive rule pointer 1 register 15	RSCAN0GAFLP115	<RSCAN0_base> + $05FC_H$
RSCAN0	Receive buffer ID register 0	RSCAN0RMID0	<RSCAN0_base> + 0600_H
RSCAN0	Receive buffer pointer register 0	RSCAN0RMPTR0	<RSCAN0_base> + 0604_H

Table 20.12 Registers (11/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 0	RSCAN0RMDF00	<RSCAN0_base> + 0608_H
RSCAN0	Receive buffer data field 1 register 0	RSCAN0RMDF10	<RSCAN0_base> + $060C_H$
RSCAN0	Receive buffer ID register 1	RSCAN0RMID1	<RSCAN0_base> + 0610_H
RSCAN0	Receive buffer pointer register 1	RSCAN0RMPTR1	<RSCAN0_base> + 0614_H
RSCAN0	Receive buffer data field 0 register 1	RSCAN0RMDF01	<RSCAN0_base> + 0618_H
RSCAN0	Receive buffer data field 1 register 1	RSCAN0RMDF11	<RSCAN0_base> + $061C_H$
RSCAN0	Receive buffer ID register 2	RSCAN0RMID2	<RSCAN0_base> + 0620_H
RSCAN0	Receive buffer pointer register 2	RSCAN0RMPTR2	<RSCAN0_base> + 0624_H
RSCAN0	Receive buffer data field 0 register 2	RSCAN0RMDF02	<RSCAN0_base> + 0628_H
RSCAN0	Receive buffer data field 1 register 2	RSCAN0RMDF12	<RSCAN0_base> + $062C_H$
RSCAN0	Receive buffer ID register 3	RSCAN0RMID3	<RSCAN0_base> + 0630_H
RSCAN0	Receive buffer pointer register 3	RSCAN0RMPTR3	<RSCAN0_base> + 0634_H
RSCAN0	Receive buffer data field 0 register 3	RSCAN0RMDF03	<RSCAN0_base> + 0638_H
RSCAN0	Receive buffer data field 1 register 3	RSCAN0RMDF13	<RSCAN0_base> + $063C_H$
RSCAN0	Receive buffer ID register 4	RSCAN0RMID4	<RSCAN0_base> + 0640_H
RSCAN0	Receive buffer pointer register 4	RSCAN0RMPTR4	<RSCAN0_base> + 0644_H
RSCAN0	Receive buffer data field 0 register 4	RSCAN0RMDF04	<RSCAN0_base> + 0648_H
RSCAN0	Receive buffer data field 1 register 4	RSCAN0RMDF14	<RSCAN0_base> + $064C_H$
RSCAN0	Receive buffer ID register 5	RSCAN0RMID5	<RSCAN0_base> + 0650_H
RSCAN0	Receive buffer pointer register 5	RSCAN0RMPTR5	<RSCAN0_base> + 0654_H
RSCAN0	Receive buffer data field 0 register 5	RSCAN0RMDF05	<RSCAN0_base> + 0658_H
RSCAN0	Receive buffer data field 1 register 5	RSCAN0RMDF15	<RSCAN0_base> + $065C_H$
RSCAN0	Receive buffer ID register 6	RSCAN0RMID6	<RSCAN0_base> + 0660_H
RSCAN0	Receive buffer pointer register 6	RSCAN0RMPTR6	<RSCAN0_base> + 0664_H
RSCAN0	Receive buffer data field 0 register 6	RSCAN0RMDF06	<RSCAN0_base> + 0668_H
RSCAN0	Receive buffer data field 1 register 6	RSCAN0RMDF16	<RSCAN0_base> + $066C_H$
RSCAN0	Receive buffer ID register 7	RSCAN0RMID7	<RSCAN0_base> + 0670_H
RSCAN0	Receive buffer pointer register 7	RSCAN0RMPTR7	<RSCAN0_base> + 0674_H
RSCAN0	Receive buffer data field 0 register 7	RSCAN0RMDF07	<RSCAN0_base> + 0678_H
RSCAN0	Receive buffer data field 1 register 7	RSCAN0RMDF17	<RSCAN0_base> + $067C_H$
RSCAN0	Receive buffer ID register 8	RSCAN0RMID8	<RSCAN0_base> + 0680_H
RSCAN0	Receive buffer pointer register 8	RSCAN0RMPTR8	<RSCAN0_base> + 0684_H
RSCAN0	Receive buffer data field 0 register 8	RSCAN0RMDF08	<RSCAN0_base> + 0688_H
RSCAN0	Receive buffer data field 1 register 8	RSCAN0RMDF18	<RSCAN0_base> + $068C_H$
RSCAN0	Receive buffer ID register 9	RSCAN0RMID9	<RSCAN0_base> + 0690_H
RSCAN0	Receive buffer pointer register 9	RSCAN0RMPTR9	<RSCAN0_base> + 0694_H
RSCAN0	Receive buffer data field 0 register 9	RSCAN0RMDF09	<RSCAN0_base> + 0698_H
RSCAN0	Receive buffer data field 1 register 9	RSCAN0RMDF19	<RSCAN0_base> + $069C_H$
RSCAN0	Receive buffer ID register 10	RSCAN0RMID10	<RSCAN0_base> + $06A0_H$
RSCAN0	Receive buffer pointer register 10	RSCAN0RMPTR10	<RSCAN0_base> + $06A4_H$
RSCAN0	Receive buffer data field 0 register 10	RSCAN0RMDF010	<RSCAN0_base> + $06A8_H$
RSCAN0	Receive buffer data field 1 register 10	RSCAN0RMDF110	<RSCAN0_base> + $06AC_H$
RSCAN0	Receive buffer ID register 11	RSCAN0RMID11	<RSCAN0_base> + $06B0_H$
RSCAN0	Receive buffer pointer register 11	RSCAN0RMPTR11	<RSCAN0_base> + $06B4_H$

Table 20.12 Registers (12/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 11	RSCAN0RMDF011	<RSCAN0_base> + $06B8_H$
RSCAN0	Receive buffer data field 1 register 11	RSCAN0RMDF111	<RSCAN0_base> + $06BC_H$
RSCAN0	Receive buffer ID register 12	RSCAN0RMID12	<RSCAN0_base> + $06C0_H$
RSCAN0	Receive buffer pointer register 12	RSCAN0RMPTR12	<RSCAN0_base> + $06C4_H$
RSCAN0	Receive buffer data field 0 register 12	RSCAN0RMDF012	<RSCAN0_base> + $06C8_H$
RSCAN0	Receive buffer data field 1 register 12	RSCAN0RMDF112	<RSCAN0_base> + $06CC_H$
RSCAN0	Receive buffer ID register 13	RSCAN0RMID13	<RSCAN0_base> + $06D0_H$
RSCAN0	Receive buffer pointer register 13	RSCAN0RMPTR13	<RSCAN0_base> + $06D4_H$
RSCAN0	Receive buffer data field 0 register 13	RSCAN0RMDF013	<RSCAN0_base> + $06D8_H$
RSCAN0	Receive buffer data field 1 register 13	RSCAN0RMDF113	<RSCAN0_base> + $06DC_H$
RSCAN0	Receive buffer ID register 14	RSCAN0RMID14	<RSCAN0_base> + $06E0_H$
RSCAN0	Receive buffer pointer register 14	RSCAN0RMPTR14	<RSCAN0_base> + $06E4_H$
RSCAN0	Receive buffer data field 0 register 14	RSCAN0RMDF014	<RSCAN0_base> + $06E8_H$
RSCAN0	Receive buffer data field 1 register 14	RSCAN0RMDF114	<RSCAN0_base> + $06EC_H$
RSCAN0	Receive buffer ID register 15	RSCAN0RMID15	<RSCAN0_base> + $06F0_H$
RSCAN0	Receive buffer pointer register 15	RSCAN0RMPTR15	<RSCAN0_base> + $06F4_H$
RSCAN0	Receive buffer data field 0 register 15	RSCAN0RMDF015	<RSCAN0_base> + $06F8_H$
RSCAN0	Receive buffer data field 1 register 15	RSCAN0RMDF115	<RSCAN0_base> + $06FC_H$
RSCAN0	Receive buffer ID register 16	RSCAN0RMID16	<RSCAN0_base> + 0700_H
RSCAN0	Receive buffer pointer register 16	RSCAN0RMPTR16	<RSCAN0_base> + 0704_H
RSCAN0	Receive buffer data field 0 register 16	RSCAN0RMDF016	<RSCAN0_base> + 0708_H
RSCAN0	Receive buffer data field 1 register 16	RSCAN0RMDF116	<RSCAN0_base> + $070C_H$
RSCAN0	Receive buffer ID register 17	RSCAN0RMID17	<RSCAN0_base> + 0710_H
RSCAN0	Receive buffer pointer register 17	RSCAN0RMPTR17	<RSCAN0_base> + 0714_H
RSCAN0	Receive buffer data field 0 register 17	RSCAN0RMDF017	<RSCAN0_base> + 0718_H
RSCAN0	Receive buffer data field 1 register 17	RSCAN0RMDF117	<RSCAN0_base> + $071C_H$
RSCAN0	Receive buffer ID register 18	RSCAN0RMID18	<RSCAN0_base> + 0720_H
RSCAN0	Receive buffer pointer register 18	RSCAN0RMPTR18	<RSCAN0_base> + 0724_H
RSCAN0	Receive buffer data field 0 register 18	RSCAN0RMDF018	<RSCAN0_base> + 0728_H
RSCAN0	Receive buffer data field 1 register 18	RSCAN0RMDF118	<RSCAN0_base> + $072C_H$
RSCAN0	Receive buffer ID register 19	RSCAN0RMID19	<RSCAN0_base> + 0730_H
RSCAN0	Receive buffer pointer register 19	RSCAN0RMPTR19	<RSCAN0_base> + 0734_H
RSCAN0	Receive buffer data field 0 register 19	RSCAN0RMDF019	<RSCAN0_base> + 0738_H
RSCAN0	Receive buffer data field 1 register 19	RSCAN0RMDF119	<RSCAN0_base> + $073C_H$
RSCAN0	Receive buffer ID register 20	RSCAN0RMID20	<RSCAN0_base> + 0740_H
RSCAN0	Receive buffer pointer register 20	RSCAN0RMPTR20	<RSCAN0_base> + 0744_H
RSCAN0	Receive buffer data field 0 register 20	RSCAN0RMDF020	<RSCAN0_base> + 0748_H
RSCAN0	Receive buffer data field 1 register 20	RSCAN0RMDF120	<RSCAN0_base> + $074C_H$
RSCAN0	Receive buffer ID register 21	RSCAN0RMID21	<RSCAN0_base> + 0750_H
RSCAN0	Receive buffer pointer register 21	RSCAN0RMPTR21	<RSCAN0_base> + 0754_H
RSCAN0	Receive buffer data field 0 register 21	RSCAN0RMDF021	<RSCAN0_base> + 0758_H
RSCAN0	Receive buffer data field 1 register 21	RSCAN0RMDF121	<RSCAN0_base> + $075C_H$
RSCAN0	Receive buffer ID register 22	RSCAN0RMID22	<RSCAN0_base> + 0760_H
RSCAN0	Receive buffer pointer register 22	RSCAN0RMPTR22	<RSCAN0_base> + 0764_H

Table 20.12 Registers (13/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 22	RSCAN0RMDF022	<RSCAN0_base> + 0768_H
RSCAN0	Receive buffer data field 1 register 22	RSCAN0RMDF122	<RSCAN0_base> + $076C_H$
RSCAN0	Receive buffer ID register 23	RSCAN0RMID23	<RSCAN0_base> + 0770_H
RSCAN0	Receive buffer pointer register 23	RSCAN0RMPTR23	<RSCAN0_base> + 0774_H
RSCAN0	Receive buffer data field 0 register 23	RSCAN0RMDF023	<RSCAN0_base> + 0778_H
RSCAN0	Receive buffer data field 1 register 23	RSCAN0RMDF123	<RSCAN0_base> + $077C_H$
RSCAN0	Receive buffer ID register 24	RSCAN0RMID24	<RSCAN0_base> + 0780_H
RSCAN0	Receive buffer pointer register 24	RSCAN0RMPTR24	<RSCAN0_base> + 0784_H
RSCAN0	Receive buffer data field 0 register 24	RSCAN0RMDF024	<RSCAN0_base> + 0788_H
RSCAN0	Receive buffer data field 1 register 24	RSCAN0RMDF124	<RSCAN0_base> + $078C_H$
RSCAN0	Receive buffer ID register 25	RSCAN0RMID25	<RSCAN0_base> + 0790_H
RSCAN0	Receive buffer pointer register 25	RSCAN0RMPTR25	<RSCAN0_base> + 0794_H
RSCAN0	Receive buffer data field 0 register 25	RSCAN0RMDF025	<RSCAN0_base> + 0798_H
RSCAN0	Receive buffer data field 1 register 25	RSCAN0RMDF125	<RSCAN0_base> + $079C_H$
RSCAN0	Receive buffer ID register 26	RSCAN0RMID26	<RSCAN0_base> + $07A0_H$
RSCAN0	Receive buffer pointer register 26	RSCAN0RMPTR26	<RSCAN0_base> + $07A4_H$
RSCAN0	Receive buffer data field 0 register 26	RSCAN0RMDF026	<RSCAN0_base> + $07A8_H$
RSCAN0	Receive buffer data field 1 register 26	RSCAN0RMDF126	<RSCAN0_base> + $07AC_H$
RSCAN0	Receive buffer ID register 27	RSCAN0RMID27	<RSCAN0_base> + $07B0_H$
RSCAN0	Receive buffer pointer register 27	RSCAN0RMPTR27	<RSCAN0_base> + $07B4_H$
RSCAN0	Receive buffer data field 0 register 27	RSCAN0RMDF027	<RSCAN0_base> + $07B8_H$
RSCAN0	Receive buffer data field 1 register 27	RSCAN0RMDF127	<RSCAN0_base> + $07BC_H$
RSCAN0	Receive buffer ID register 28	RSCAN0RMID28	<RSCAN0_base> + $07C0_H$
RSCAN0	Receive buffer pointer register 28	RSCAN0RMPTR28	<RSCAN0_base> + $07C4_H$
RSCAN0	Receive buffer data field 0 register 28	RSCAN0RMDF028	<RSCAN0_base> + $07C8_H$
RSCAN0	Receive buffer data field 1 register 28	RSCAN0RMDF128	<RSCAN0_base> + $07CC_H$
RSCAN0	Receive buffer ID register 29	RSCAN0RMID29	<RSCAN0_base> + $07D0_H$
RSCAN0	Receive buffer pointer register 29	RSCAN0RMPTR29	<RSCAN0_base> + $07D4_H$
RSCAN0	Receive buffer data field 0 register 29	RSCAN0RMDF029	<RSCAN0_base> + $07D8_H$
RSCAN0	Receive buffer data field 1 register 29	RSCAN0RMDF129	<RSCAN0_base> + $07DC_H$
RSCAN0	Receive buffer ID register 30	RSCAN0RMID30	<RSCAN0_base> + $07E0_H$
RSCAN0	Receive buffer pointer register 30	RSCAN0RMPTR30	<RSCAN0_base> + $07E4_H$
RSCAN0	Receive buffer data field 0 register 30	RSCAN0RMDF030	<RSCAN0_base> + $07E8_H$
RSCAN0	Receive buffer data field 1 register 30	RSCAN0RMDF130	<RSCAN0_base> + $07EC_H$
RSCAN0	Receive buffer ID register 31	RSCAN0RMID31	<RSCAN0_base> + $07F0_H$
RSCAN0	Receive buffer pointer register 31	RSCAN0RMPTR31	<RSCAN0_base> + $07F4_H$
RSCAN0	Receive buffer data field 0 register 31	RSCAN0RMDF031	<RSCAN0_base> + $07F8_H$
RSCAN0	Receive buffer data field 1 register 31	RSCAN0RMDF131	<RSCAN0_base> + $07FC_H$
RSCAN0	Receive buffer ID register 32	RSCAN0RMID32	<RSCAN0_base> + 0800_H
RSCAN0	Receive buffer pointer register 32	RSCAN0RMPTR32	<RSCAN0_base> + 0804_H
RSCAN0	Receive buffer data field 0 register 32	RSCAN0RMDF032	<RSCAN0_base> + 0808_H
RSCAN0	Receive buffer data field 1 register 32	RSCAN0RMDF132	<RSCAN0_base> + $080C_H$
RSCAN0	Receive buffer ID register 33	RSCAN0RMID33	<RSCAN0_base> + 0810_H
RSCAN0	Receive buffer pointer register 33	RSCAN0RMPTR33	<RSCAN0_base> + 0814_H

Table 20.12 Registers (14/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 33	RSCAN0RMDF033	<RSCAN0_base> + 0818_H
RSCAN0	Receive buffer data field 1 register 33	RSCAN0RMDF133	<RSCAN0_base> + $081C_H$
RSCAN0	Receive buffer ID register 34	RSCAN0RMID34	<RSCAN0_base> + 0820_H
RSCAN0	Receive buffer pointer register 34	RSCAN0RMPTR34	<RSCAN0_base> + 0824_H
RSCAN0	Receive buffer data field 0 register 34	RSCAN0RMDF034	<RSCAN0_base> + 0828_H
RSCAN0	Receive buffer data field 1 register 34	RSCAN0RMDF134	<RSCAN0_base> + $082C_H$
RSCAN0	Receive buffer ID register 35	RSCAN0RMID35	<RSCAN0_base> + 0830_H
RSCAN0	Receive buffer pointer register 35	RSCAN0RMPTR35	<RSCAN0_base> + 0834_H
RSCAN0	Receive buffer data field 0 register 35	RSCAN0RMDF035	<RSCAN0_base> + 0838_H
RSCAN0	Receive buffer data field 1 register 35	RSCAN0RMDF135	<RSCAN0_base> + $083C_H$
RSCAN0	Receive buffer ID register 36	RSCAN0RMID36	<RSCAN0_base> + 0840_H
RSCAN0	Receive buffer pointer register 36	RSCAN0RMPTR36	<RSCAN0_base> + 0844_H
RSCAN0	Receive buffer data field 0 register 36	RSCAN0RMDF036	<RSCAN0_base> + 0848_H
RSCAN0	Receive buffer data field 1 register 36	RSCAN0RMDF136	<RSCAN0_base> + $084C_H$
RSCAN0	Receive buffer ID register 37	RSCAN0RMID37	<RSCAN0_base> + 0850_H
RSCAN0	Receive buffer pointer register 37	RSCAN0RMPTR37	<RSCAN0_base> + 0854_H
RSCAN0	Receive buffer data field 0 register 37	RSCAN0RMDF037	<RSCAN0_base> + 0858_H
RSCAN0	Receive buffer data field 1 register 37	RSCAN0RMDF137	<RSCAN0_base> + $085C_H$
RSCAN0	Receive buffer ID register 38	RSCAN0RMID38	<RSCAN0_base> + 0860_H
RSCAN0	Receive buffer pointer register 38	RSCAN0RMPTR38	<RSCAN0_base> + 0864_H
RSCAN0	Receive buffer data field 0 register 38	RSCAN0RMDF038	<RSCAN0_base> + 0868_H
RSCAN0	Receive buffer data field 1 register 38	RSCAN0RMDF138	<RSCAN0_base> + $086C_H$
RSCAN0	Receive buffer ID register 39	RSCAN0RMID39	<RSCAN0_base> + 0870_H
RSCAN0	Receive buffer pointer register 39	RSCAN0RMPTR39	<RSCAN0_base> + 0874_H
RSCAN0	Receive buffer data field 0 register 39	RSCAN0RMDF039	<RSCAN0_base> + 0878_H
RSCAN0	Receive buffer data field 1 register 39	RSCAN0RMDF139	<RSCAN0_base> + $087C_H$
RSCAN0	Receive buffer ID register 40	RSCAN0RMID40	<RSCAN0_base> + 0880_H
RSCAN0	Receive buffer pointer register 40	RSCAN0RMPTR40	<RSCAN0_base> + 0884_H
RSCAN0	Receive buffer data field 0 register 40	RSCAN0RMDF040	<RSCAN0_base> + 0888_H
RSCAN0	Receive buffer data field 1 register 40	RSCAN0RMDF140	<RSCAN0_base> + $088C_H$
RSCAN0	Receive buffer ID register 41	RSCAN0RMID41	<RSCAN0_base> + 0890_H
RSCAN0	Receive buffer pointer register 41	RSCAN0RMPTR41	<RSCAN0_base> + 0894_H
RSCAN0	Receive buffer data field 0 register 41	RSCAN0RMDF041	<RSCAN0_base> + 0898_H
RSCAN0	Receive buffer data field 1 register 41	RSCAN0RMDF141	<RSCAN0_base> + $089C_H$
RSCAN0	Receive buffer ID register 42	RSCAN0RMID42	<RSCAN0_base> + $08A0_H$
RSCAN0	Receive buffer pointer register 42	RSCAN0RMPTR42	<RSCAN0_base> + $08A4_H$
RSCAN0	Receive buffer data field 0 register 42	RSCAN0RMDF042	<RSCAN0_base> + $08A8_H$
RSCAN0	Receive buffer data field 1 register 42	RSCAN0RMDF142	<RSCAN0_base> + $08AC_H$
RSCAN0	Receive buffer ID register 43	RSCAN0RMID43	<RSCAN0_base> + $08B0_H$
RSCAN0	Receive buffer pointer register 43	RSCAN0RMPTR43	<RSCAN0_base> + $08B4_H$
RSCAN0	Receive buffer data field 0 register 43	RSCAN0RMDF043	<RSCAN0_base> + $08B8_H$
RSCAN0	Receive buffer data field 1 register 43	RSCAN0RMDF143	<RSCAN0_base> + $08BC_H$
RSCAN0	Receive buffer ID register 44	RSCAN0RMID44	<RSCAN0_base> + $08C0_H$
RSCAN0	Receive buffer pointer register 44	RSCAN0RMPTR44	<RSCAN0_base> + $08C4_H$

Table 20.12 Registers (15/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 44	RSCAN0RMDF044	<RSCAN0_base> + $08C8_H$
RSCAN0	Receive buffer data field 1 register 44	RSCAN0RMDF144	<RSCAN0_base> + $08CC_H$
RSCAN0	Receive buffer ID register 45	RSCAN0RMID45	<RSCAN0_base> + $08D0_H$
RSCAN0	Receive buffer pointer register 45	RSCAN0RMPTR45	<RSCAN0_base> + $08D4_H$
RSCAN0	Receive buffer data field 0 register 45	RSCAN0RMDF045	<RSCAN0_base> + $08D8_H$
RSCAN0	Receive buffer data field 1 register 45	RSCAN0RMDF145	<RSCAN0_base> + $08DC_H$
RSCAN0	Receive buffer ID register 46	RSCAN0RMID46	<RSCAN0_base> + $08E0_H$
RSCAN0	Receive buffer pointer register 46	RSCAN0RMPTR46	<RSCAN0_base> + $08E4_H$
RSCAN0	Receive buffer data field 0 register 46	RSCAN0RMDF046	<RSCAN0_base> + $08E8_H$
RSCAN0	Receive buffer data field 1 register 46	RSCAN0RMDF146	<RSCAN0_base> + $08EC_H$
RSCAN0	Receive buffer ID register 47	RSCAN0RMID47	<RSCAN0_base> + $08F0_H$
RSCAN0	Receive buffer pointer register 47	RSCAN0RMPTR47	<RSCAN0_base> + $08F4_H$
RSCAN0	Receive buffer data field 0 register 47	RSCAN0RMDF047	<RSCAN0_base> + $08F8_H$
RSCAN0	Receive buffer data field 1 register 47	RSCAN0RMDF147	<RSCAN0_base> + $08FC_H$
RSCAN0	Receive buffer ID register 48	RSCAN0RMID48	<RSCAN0_base> + 0900_H
RSCAN0	Receive buffer pointer register 48	RSCAN0RMPTR48	<RSCAN0_base> + 0904_H
RSCAN0	Receive buffer data field 0 register 48	RSCAN0RMDF048	<RSCAN0_base> + 0908_H
RSCAN0	Receive buffer data field 1 register 48	RSCAN0RMDF148	<RSCAN0_base> + $090C_H$
RSCAN0	Receive buffer ID register 49	RSCAN0RMID49	<RSCAN0_base> + 0910_H
RSCAN0	Receive buffer pointer register 49	RSCAN0RMPTR49	<RSCAN0_base> + 0914_H
RSCAN0	Receive buffer data field 0 register 49	RSCAN0RMDF049	<RSCAN0_base> + 0918_H
RSCAN0	Receive buffer data field 1 register 49	RSCAN0RMDF149	<RSCAN0_base> + $091C_H$
RSCAN0	Receive buffer ID register 50	RSCAN0RMID50	<RSCAN0_base> + 0920_H
RSCAN0	Receive buffer pointer register 50	RSCAN0RMPTR50	<RSCAN0_base> + 0924_H
RSCAN0	Receive buffer data field 0 register 50	RSCAN0RMDF050	<RSCAN0_base> + 0928_H
RSCAN0	Receive buffer data field 1 register 50	RSCAN0RMDF150	<RSCAN0_base> + $092C_H$
RSCAN0	Receive buffer ID register 51	RSCAN0RMID51	<RSCAN0_base> + 0930_H
RSCAN0	Receive buffer pointer register 51	RSCAN0RMPTR51	<RSCAN0_base> + 0934_H
RSCAN0	Receive buffer data field 0 register 51	RSCAN0RMDF051	<RSCAN0_base> + 0938_H
RSCAN0	Receive buffer data field 1 register 51	RSCAN0RMDF151	<RSCAN0_base> + $093C_H$
RSCAN0	Receive buffer ID register 52	RSCAN0RMID52	<RSCAN0_base> + 0940_H
RSCAN0	Receive buffer pointer register 52	RSCAN0RMPTR52	<RSCAN0_base> + 0944_H
RSCAN0	Receive buffer data field 0 register 52	RSCAN0RMDF052	<RSCAN0_base> + 0948_H
RSCAN0	Receive buffer data field 1 register 52	RSCAN0RMDF152	<RSCAN0_base> + $094C_H$
RSCAN0	Receive buffer ID register 53	RSCAN0RMID53	<RSCAN0_base> + 0950_H
RSCAN0	Receive buffer pointer register 53	RSCAN0RMPTR53	<RSCAN0_base> + 0954_H
RSCAN0	Receive buffer data field 0 register 53	RSCAN0RMDF053	<RSCAN0_base> + 0958_H
RSCAN0	Receive buffer data field 1 register 53	RSCAN0RMDF153	<RSCAN0_base> + $095C_H$
RSCAN0	Receive buffer ID register 54	RSCAN0RMID54	<RSCAN0_base> + 0960_H
RSCAN0	Receive buffer pointer register 54	RSCAN0RMPTR54	<RSCAN0_base> + 0964_H
RSCAN0	Receive buffer data field 0 register 54	RSCAN0RMDF054	<RSCAN0_base> + 0968_H
RSCAN0	Receive buffer data field 1 register 54	RSCAN0RMDF154	<RSCAN0_base> + $096C_H$
RSCAN0	Receive buffer ID register 55	RSCAN0RMID55	<RSCAN0_base> + 0970_H
RSCAN0	Receive buffer pointer register 55	RSCAN0RMPTR55	<RSCAN0_base> + 0974_H

Table 20.12 Registers (16/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 55	RSCAN0RMDF055	<RSCAN0_base> + 0978_H
RSCAN0	Receive buffer data field 1 register 55	RSCAN0RMDF155	<RSCAN0_base> + $097C_H$
RSCAN0	Receive buffer ID register 56	RSCAN0RMID56	<RSCAN0_base> + 0980_H
RSCAN0	Receive buffer pointer register 56	RSCAN0RMPTR56	<RSCAN0_base> + 0984_H
RSCAN0	Receive buffer data field 0 register 56	RSCAN0RMDF056	<RSCAN0_base> + 0988_H
RSCAN0	Receive buffer data field 1 register 56	RSCAN0RMDF156	<RSCAN0_base> + $098C_H$
RSCAN0	Receive buffer ID register 57	RSCAN0RMID57	<RSCAN0_base> + 0990_H
RSCAN0	Receive buffer pointer register 57	RSCAN0RMPTR57	<RSCAN0_base> + 0994_H
RSCAN0	Receive buffer data field 0 register 57	RSCAN0RMDF057	<RSCAN0_base> + 0998_H
RSCAN0	Receive buffer data field 1 register 57	RSCAN0RMDF157	<RSCAN0_base> + $099C_H$
RSCAN0	Receive buffer ID register 58	RSCAN0RMID58	<RSCAN0_base> + $09A0_H$
RSCAN0	Receive buffer pointer register 58	RSCAN0RMPTR58	<RSCAN0_base> + $09A4_H$
RSCAN0	Receive buffer data field 0 register 58	RSCAN0RMDF058	<RSCAN0_base> + $09A8_H$
RSCAN0	Receive buffer data field 1 register 58	RSCAN0RMDF158	<RSCAN0_base> + $09AC_H$
RSCAN0	Receive buffer ID register 59	RSCAN0RMID59	<RSCAN0_base> + $09B0_H$
RSCAN0	Receive buffer pointer register 59	RSCAN0RMPTR59	<RSCAN0_base> + $09B4_H$
RSCAN0	Receive buffer data field 0 register 59	RSCAN0RMDF059	<RSCAN0_base> + $09B8_H$
RSCAN0	Receive buffer data field 1 register 59	RSCAN0RMDF159	<RSCAN0_base> + $09BC_H$
RSCAN0	Receive buffer ID register 60	RSCAN0RMID60	<RSCAN0_base> + $09C0_H$
RSCAN0	Receive buffer pointer register 60	RSCAN0RMPTR60	<RSCAN0_base> + $09C4_H$
RSCAN0	Receive buffer data field 0 register 60	RSCAN0RMDF060	<RSCAN0_base> + $09C8_H$
RSCAN0	Receive buffer data field 1 register 60	RSCAN0RMDF160	<RSCAN0_base> + $09CC_H$
RSCAN0	Receive buffer ID register 61	RSCAN0RMID61	<RSCAN0_base> + $09D0_H$
RSCAN0	Receive buffer pointer register 61	RSCAN0RMPTR61	<RSCAN0_base> + $09D4_H$
RSCAN0	Receive buffer data field 0 register 61	RSCAN0RMDF061	<RSCAN0_base> + $09D8_H$
RSCAN0	Receive buffer data field 1 register 61	RSCAN0RMDF161	<RSCAN0_base> + $09DC_H$
RSCAN0	Receive buffer ID register 62	RSCAN0RMID62	<RSCAN0_base> + $09E0_H$
RSCAN0	Receive buffer pointer register 62	RSCAN0RMPTR62	<RSCAN0_base> + $09E4_H$
RSCAN0	Receive buffer data field 0 register 62	RSCAN0RMDF062	<RSCAN0_base> + $09E8_H$
RSCAN0	Receive buffer data field 1 register 62	RSCAN0RMDF162	<RSCAN0_base> + $09EC_H$
RSCAN0	Receive buffer ID register 63	RSCAN0RMID63	<RSCAN0_base> + $09F0_H$
RSCAN0	Receive buffer pointer register 63	RSCAN0RMPTR63	<RSCAN0_base> + $09F4_H$
RSCAN0	Receive buffer data field 0 register 63	RSCAN0RMDF063	<RSCAN0_base> + $09F8_H$
RSCAN0	Receive buffer data field 1 register 63	RSCAN0RMDF163	<RSCAN0_base> + $09FC_H$
RSCAN0	Receive buffer ID register 64	RSCAN0RMID64	<RSCAN0_base> + $0A00_H$
RSCAN0	Receive buffer pointer register 64	RSCAN0RMPTR64	<RSCAN0_base> + $0A04_H$
RSCAN0	Receive buffer data field 0 register 64	RSCAN0RMDF064	<RSCAN0_base> + $0A08_H$
RSCAN0	Receive buffer data field 1 register 64	RSCAN0RMDF164	<RSCAN0_base> + $0A0C_H$
RSCAN0	Receive buffer ID register 65	RSCAN0RMID65	<RSCAN0_base> + $0A10_H$
RSCAN0	Receive buffer pointer register 65	RSCAN0RMPTR65	<RSCAN0_base> + $0A14_H$
RSCAN0	Receive buffer data field 0 register 65	RSCAN0RMDF065	<RSCAN0_base> + $0A18_H$
RSCAN0	Receive buffer data field 1 register 65	RSCAN0RMDF165	<RSCAN0_base> + $0A1C_H$
RSCAN0	Receive buffer ID register 66	RSCAN0RMID66	<RSCAN0_base> + $0A20_H$
RSCAN0	Receive buffer pointer register 66	RSCAN0RMPTR66	<RSCAN0_base> + $0A24_H$

Table 20.12 Registers (17/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 66	RSCAN0RMDF066	<RSCAN0_base> + $0A28_H$
RSCAN0	Receive buffer data field 1 register 66	RSCAN0RMDF166	<RSCAN0_base> + $0A2C_H$
RSCAN0	Receive buffer ID register 67	RSCAN0RMID67	<RSCAN0_base> + $0A30_H$
RSCAN0	Receive buffer pointer register 67	RSCAN0RMPTR67	<RSCAN0_base> + $0A34_H$
RSCAN0	Receive buffer data field 0 register 67	RSCAN0RMDF067	<RSCAN0_base> + $0A38_H$
RSCAN0	Receive buffer data field 1 register 67	RSCAN0RMDF167	<RSCAN0_base> + $0A3C_H$
RSCAN0	Receive buffer ID register 68	RSCAN0RMID68	<RSCAN0_base> + $0A40_H$
RSCAN0	Receive buffer pointer register 68	RSCAN0RMPTR68	<RSCAN0_base> + $0A44_H$
RSCAN0	Receive buffer data field 0 register 68	RSCAN0RMDF068	<RSCAN0_base> + $0A48_H$
RSCAN0	Receive buffer data field 1 register 68	RSCAN0RMDF168	<RSCAN0_base> + $0A4C_H$
RSCAN0	Receive buffer ID register 69	RSCAN0RMID69	<RSCAN0_base> + $0A50_H$
RSCAN0	Receive buffer pointer register 69	RSCAN0RMPTR69	<RSCAN0_base> + $0A54_H$
RSCAN0	Receive buffer data field 0 register 69	RSCAN0RMDF069	<RSCAN0_base> + $0A58_H$
RSCAN0	Receive buffer data field 1 register 69	RSCAN0RMDF169	<RSCAN0_base> + $0A5C_H$
RSCAN0	Receive buffer ID register 70	RSCAN0RMID70	<RSCAN0_base> + $0A60_H$
RSCAN0	Receive buffer pointer register 70	RSCAN0RMPTR70	<RSCAN0_base> + $0A64_H$
RSCAN0	Receive buffer data field 0 register 70	RSCAN0RMDF070	<RSCAN0_base> + $0A68_H$
RSCAN0	Receive buffer data field 1 register 70	RSCAN0RMDF170	<RSCAN0_base> + $0A6C_H$
RSCAN0	Receive buffer ID register 71	RSCAN0RMID71	<RSCAN0_base> + $0A70_H$
RSCAN0	Receive buffer pointer register 71	RSCAN0RMPTR71	<RSCAN0_base> + $0A74_H$
RSCAN0	Receive buffer data field 0 register 71	RSCAN0RMDF071	<RSCAN0_base> + $0A78_H$
RSCAN0	Receive buffer data field 1 register 71	RSCAN0RMDF171	<RSCAN0_base> + $0A7C_H$
RSCAN0	Receive buffer ID register 72	RSCAN0RMID72	<RSCAN0_base> + $0A80_H$
RSCAN0	Receive buffer pointer register 72	RSCAN0RMPTR72	<RSCAN0_base> + $0A84_H$
RSCAN0	Receive buffer data field 0 register 72	RSCAN0RMDF072	<RSCAN0_base> + $0A88_H$
RSCAN0	Receive buffer data field 1 register 72	RSCAN0RMDF172	<RSCAN0_base> + $0A8C_H$
RSCAN0	Receive buffer ID register 73	RSCAN0RMID73	<RSCAN0_base> + $0A90_H$
RSCAN0	Receive buffer pointer register 73	RSCAN0RMPTR73	<RSCAN0_base> + $0A94_H$
RSCAN0	Receive buffer data field 0 register 73	RSCAN0RMDF073	<RSCAN0_base> + $0A98_H$
RSCAN0	Receive buffer data field 1 register 73	RSCAN0RMDF173	<RSCAN0_base> + $0A9C_H$
RSCAN0	Receive buffer ID register 74	RSCAN0RMID74	<RSCAN0_base> + $0AA0_H$
RSCAN0	Receive buffer pointer register 74	RSCAN0RMPTR74	<RSCAN0_base> + $0AA4_H$
RSCAN0	Receive buffer data field 0 register 74	RSCAN0RMDF074	<RSCAN0_base> + $0AA8_H$
RSCAN0	Receive buffer data field 1 register 74	RSCAN0RMDF174	<RSCAN0_base> + $0AAC_H$
RSCAN0	Receive buffer ID register 75	RSCAN0RMID75	<RSCAN0_base> + $0AB0_H$
RSCAN0	Receive buffer pointer register 75	RSCAN0RMPTR75	<RSCAN0_base> + $0AB4_H$
RSCAN0	Receive buffer data field 0 register 75	RSCAN0RMDF075	<RSCAN0_base> + $0AB8_H$
RSCAN0	Receive buffer data field 1 register 75	RSCAN0RMDF175	<RSCAN0_base> + $0ABC_H$
RSCAN0	Receive buffer ID register 76	RSCAN0RMID76	<RSCAN0_base> + $0AC0_H$
RSCAN0	Receive buffer pointer register 76	RSCAN0RMPTR76	<RSCAN0_base> + $0AC4_H$
RSCAN0	Receive buffer data field 0 register 76	RSCAN0RMDF076	<RSCAN0_base> + $0AC8_H$
RSCAN0	Receive buffer data field 1 register 76	RSCAN0RMDF176	<RSCAN0_base> + $0ACC_H$
RSCAN0	Receive buffer ID register 77	RSCAN0RMID77	<RSCAN0_base> + $0AD0_H$
RSCAN0	Receive buffer pointer register 77	RSCAN0RMPTR77	<RSCAN0_base> + $0AD4_H$

Table 20.12 Registers (18/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 77	RSCAN0RMDF077	<RSCAN0_base> + $0AD8_H$
RSCAN0	Receive buffer data field 1 register 77	RSCAN0RMDF177	<RSCAN0_base> + $0ADC_H$
RSCAN0	Receive buffer ID register 78	RSCAN0RMID78	<RSCAN0_base> + $0AE0_H$
RSCAN0	Receive buffer pointer register 78	RSCAN0RMPTR78	<RSCAN0_base> + $0AE4_H$
RSCAN0	Receive buffer data field 0 register 78	RSCAN0RMDF078	<RSCAN0_base> + $0AE8_H$
RSCAN0	Receive buffer data field 1 register 78	RSCAN0RMDF178	<RSCAN0_base> + $0AEC_H$
RSCAN0	Receive buffer ID register 79	RSCAN0RMID79	<RSCAN0_base> + $0AF0_H$
RSCAN0	Receive buffer pointer register 79	RSCAN0RMPTR79	<RSCAN0_base> + $0AF4_H$
RSCAN0	Receive buffer data field 0 register 79	RSCAN0RMDF079	<RSCAN0_base> + $0AF8_H$
RSCAN0	Receive buffer data field 1 register 79	RSCAN0RMDF179	<RSCAN0_base> + $0AFC_H$
RSCAN0	Receive buffer ID register 80	RSCAN0RMID80	<RSCAN0_base> + $0B00_H$
RSCAN0	Receive buffer pointer register 80	RSCAN0RMPTR80	<RSCAN0_base> + $0B04_H$
RSCAN0	Receive buffer data field 0 register 80	RSCAN0RMDF080	<RSCAN0_base> + $0B08_H$
RSCAN0	Receive buffer data field 1 register 80	RSCAN0RMDF180	<RSCAN0_base> + $0B0C_H$
RSCAN0	Receive buffer ID register 81	RSCAN0RMID81	<RSCAN0_base> + $0B10_H$
RSCAN0	Receive buffer pointer register 81	RSCAN0RMPTR81	<RSCAN0_base> + $0B14_H$
RSCAN0	Receive buffer data field 0 register 81	RSCAN0RMDF081	<RSCAN0_base> + $0B18_H$
RSCAN0	Receive buffer data field 1 register 81	RSCAN0RMDF181	<RSCAN0_base> + $0B1C_H$
RSCAN0	Receive buffer ID register 82	RSCAN0RMID82	<RSCAN0_base> + $0B20_H$
RSCAN0	Receive buffer pointer register 82	RSCAN0RMPTR82	<RSCAN0_base> + $0B24_H$
RSCAN0	Receive buffer data field 0 register 82	RSCAN0RMDF082	<RSCAN0_base> + $0B28_H$
RSCAN0	Receive buffer data field 1 register 82	RSCAN0RMDF182	<RSCAN0_base> + $0B2C_H$
RSCAN0	Receive buffer ID register 83	RSCAN0RMID83	<RSCAN0_base> + $0B30_H$
RSCAN0	Receive buffer pointer register 83	RSCAN0RMPTR83	<RSCAN0_base> + $0B34_H$
RSCAN0	Receive buffer data field 0 register 83	RSCAN0RMDF083	<RSCAN0_base> + $0B38_H$
RSCAN0	Receive buffer data field 1 register 83	RSCAN0RMDF183	<RSCAN0_base> + $0B3C_H$
RSCAN0	Receive buffer ID register 84	RSCAN0RMID84	<RSCAN0_base> + $0B40_H$
RSCAN0	Receive buffer pointer register 84	RSCAN0RMPTR84	<RSCAN0_base> + $0B44_H$
RSCAN0	Receive buffer data field 0 register 84	RSCAN0RMDF084	<RSCAN0_base> + $0B48_H$
RSCAN0	Receive buffer data field 1 register 84	RSCAN0RMDF184	<RSCAN0_base> + $0B4C_H$
RSCAN0	Receive buffer ID register 85	RSCAN0RMID85	<RSCAN0_base> + $0B50_H$
RSCAN0	Receive buffer pointer register 85	RSCAN0RMPTR85	<RSCAN0_base> + $0B54_H$
RSCAN0	Receive buffer data field 0 register 85	RSCAN0RMDF085	<RSCAN0_base> + $0B58_H$
RSCAN0	Receive buffer data field 1 register 85	RSCAN0RMDF185	<RSCAN0_base> + $0B5C_H$
RSCAN0	Receive buffer ID register 86	RSCAN0RMID86	<RSCAN0_base> + $0B60_H$
RSCAN0	Receive buffer pointer register 86	RSCAN0RMPTR86	<RSCAN0_base> + $0B64_H$
RSCAN0	Receive buffer data field 0 register 86	RSCAN0RMDF086	<RSCAN0_base> + $0B68_H$
RSCAN0	Receive buffer data field 1 register 86	RSCAN0RMDF186	<RSCAN0_base> + $0B6C_H$
RSCAN0	Receive buffer ID register 87	RSCAN0RMID87	<RSCAN0_base> + $0B70_H$
RSCAN0	Receive buffer pointer register 87	RSCAN0RMPTR87	<RSCAN0_base> + $0B74_H$
RSCAN0	Receive buffer data field 0 register 87	RSCAN0RMDF087	<RSCAN0_base> + $0B78_H$
RSCAN0	Receive buffer data field 1 register 87	RSCAN0RMDF187	<RSCAN0_base> + $0B7C_H$
RSCAN0	Receive buffer ID register 88	RSCAN0RMID88	<RSCAN0_base> + $0B80_H$
RSCAN0	Receive buffer pointer register 88	RSCAN0RMPTR88	<RSCAN0_base> + $0B84_H$

Table 20.12 Registers (19/32)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 0 register 88	RSCAN0RMDF088	<RSCAN0_base> + $0B88_H$
RSCAN0	Receive buffer data field 1 register 88	RSCAN0RMDF188	<RSCAN0_base> + $0B8C_H$
RSCAN0	Receive buffer ID register 89	RSCAN0RMID89	<RSCAN0_base> + $0B90_H$
RSCAN0	Receive buffer pointer register 89	RSCAN0RMPTR89	<RSCAN0_base> + $0B94_H$
RSCAN0	Receive buffer data field 0 register 89	RSCAN0RMDF089	<RSCAN0_base> + $0B98_H$
RSCAN0	Receive buffer data field 1 register 89	RSCAN0RMDF189	<RSCAN0_base> + $0B9C_H$
RSCAN0	Receive buffer ID register 90	RSCAN0RMID90	<RSCAN0_base> + $0BA0_H$
RSCAN0	Receive buffer pointer register 90	RSCAN0RMPTR90	<RSCAN0_base> + $0BA4_H$
RSCAN0	Receive buffer data field 0 register 90	RSCAN0RMDF090	<RSCAN0_base> + $0BA8_H$
RSCAN0	Receive buffer data field 1 register 90	RSCAN0RMDF190	<RSCAN0_base> + $0BAC_H$
RSCAN0	Receive buffer ID register 91	RSCAN0RMID91	<RSCAN0_base> + $0BB0_H$
RSCAN0	Receive buffer pointer register 91	RSCAN0RMPTR91	<RSCAN0_base> + $0BB4_H$
RSCAN0	Receive buffer data field 0 register 91	RSCAN0RMDF091	<RSCAN0_base> + $0BB8_H$
RSCAN0	Receive buffer data field 1 register 91	RSCAN0RMDF191	<RSCAN0_base> + $0BBC_H$
RSCAN0	Receive buffer ID register 92	RSCAN0RMID92	<RSCAN0_base> + $0BC0_H$
RSCAN0	Receive buffer pointer register 92	RSCAN0RMPTR92	<RSCAN0_base> + $0BC4_H$
RSCAN0	Receive buffer data field 0 register 92	RSCAN0RMDF092	<RSCAN0_base> + $0BC8_H$
RSCAN0	Receive buffer data field 1 register 92	RSCAN0RMDF192	<RSCAN0_base> + $0BCC_H$
RSCAN0	Receive buffer ID register 93	RSCAN0RMID93	<RSCAN0_base> + $0BD0_H$
RSCAN0	Receive buffer pointer register 93	RSCAN0RMPTR93	<RSCAN0_base> + $0BD4_H$
RSCAN0	Receive buffer data field 0 register 93	RSCAN0RMDF093	<RSCAN0_base> + $0BD8_H$
RSCAN0	Receive buffer data field 1 register 93	RSCAN0RMDF193	<RSCAN0_base> + $0BDC_H$
RSCAN0	Receive buffer ID register 94	RSCAN0RMID94	<RSCAN0_base> + $0BE0_H$
RSCAN0	Receive buffer pointer register 94	RSCAN0RMPTR94	<RSCAN0_base> + $0BE4_H$
RSCAN0	Receive buffer data field 0 register 94	RSCAN0RMDF094	<RSCAN0_base> + $0BE8_H$
RSCAN0	Receive buffer data field 1 register 94	RSCAN0RMDF194	<RSCAN0_base> + $0BEC_H$
RSCAN0	Receive buffer ID register 95	RSCAN0RMID95	<RSCAN0_base> + $0BF0_H$
RSCAN0	Receive buffer pointer register 95	RSCAN0RMPTR95	<RSCAN0_base> + $0BF4_H$
RSCAN0	Receive buffer data field 0 register 95	RSCAN0RMDF095	<RSCAN0_base> + $0BF8_H$
RSCAN0	Receive buffer data field 1 register 95	RSCAN0RMDF195	<RSCAN0_base> + $0BFC_H$
RSCAN0	Receive FIFO buffer access ID register 0	RSCAN0RFID0	<RSCAN0_base> + $0E00_H$
RSCAN0	Receive FIFO buffer access pointer register 0	RSCAN0RFPTR0	<RSCAN0_base> + $0E04_H$
RSCAN0	Receive FIFO buffer access data field 0 register 0	RSCAN0RFDF00	<RSCAN0_base> + $0E08_H$
RSCAN0	Receive FIFO buffer access data field 1 register 0	RSCAN0RFDF10	<RSCAN0_base> + $0E0C_H$
RSCAN0	Receive FIFO buffer access ID register 1	RSCAN0RFID1	<RSCAN0_base> + $0E10_H$
RSCAN0	Receive FIFO buffer access pointer register 1	RSCAN0RFPTR1	<RSCAN0_base> + $0E14_H$
RSCAN0	Receive FIFO buffer access data field 0 register 1	RSCAN0RFDF01	<RSCAN0_base> + $0E18_H$
RSCAN0	Receive FIFO buffer access data field 1 register 1	RSCAN0RFDF11	<RSCAN0_base> + $0E1C_H$
RSCAN0	Receive FIFO buffer access ID register 2	RSCAN0RFID2	<RSCAN0_base> + $0E20_H$
RSCAN0	Receive FIFO buffer access pointer register 2	RSCAN0RFPTR2	<RSCAN0_base> + $0E24_H$
RSCAN0	Receive FIFO buffer access data field 0 register 2	RSCAN0RFDF02	<RSCAN0_base> + $0E28_H$
RSCAN0	Receive FIFO buffer access data field 1 register 2	RSCAN0RFDF12	<RSCAN0_base> + $0E2C_H$
RSCAN0	Receive FIFO buffer access ID register 3	RSCAN0RFID3	<RSCAN0_base> + $0E30_H$
RSCAN0	Receive FIFO buffer access pointer register 3	RSCAN0RFPTR3	<RSCAN0_base> + $0E34_H$

Table 20.12 Registers (20/32)

Module	Register	Symbol	Address
RSCAN0	Receive FIFO buffer access data field 0 register 3	RSCAN0RFDF03	<RSCAN0_base> + $0E38_H$
RSCAN0	Receive FIFO buffer access data field 1 register 3	RSCAN0RFDF13	<RSCAN0_base> + $0E3C_H$
RSCAN0	Receive FIFO buffer access ID register 4	RSCAN0RFID4	<RSCAN0_base> + $0E40_H$
RSCAN0	Receive FIFO buffer access pointer register 4	RSCAN0RFPTR4	<RSCAN0_base> + $0E44_H$
RSCAN0	Receive FIFO buffer access data field 0 register 4	RSCAN0RFDF04	<RSCAN0_base> + $0E48_H$
RSCAN0	Receive FIFO buffer access data field 1 register 4	RSCAN0RFDF14	<RSCAN0_base> + $0E4C_H$
RSCAN0	Receive FIFO buffer access ID register 5	RSCAN0RFID5	<RSCAN0_base> + $0E50_H$
RSCAN0	Receive FIFO buffer access pointer register 5	RSCAN0RFPTR5	<RSCAN0_base> + $0E54_H$
RSCAN0	Receive FIFO buffer access data field 0 register 5	RSCAN0RFDF05	<RSCAN0_base> + $0E58_H$
RSCAN0	Receive FIFO buffer access data field 1 register 5	RSCAN0RFDF15	<RSCAN0_base> + $0E5C_H$
RSCAN0	Receive FIFO buffer access ID register 6	RSCAN0RFID6	<RSCAN0_base> + $0E60_H$
RSCAN0	Receive FIFO buffer access pointer register 6	RSCAN0RFPTR6	<RSCAN0_base> + $0E64_H$
RSCAN0	Receive FIFO buffer access data field 0 register 6	RSCAN0RFDF06	<RSCAN0_base> + $0E68_H$
RSCAN0	Receive FIFO buffer access data field 1 register 6	RSCAN0RFDF16	<RSCAN0_base> + $0E6C_H$
RSCAN0	Receive FIFO buffer access ID register 7	RSCAN0RFID7	<RSCAN0_base> + $0E70_H$
RSCAN0	Receive FIFO buffer access pointer register 7	RSCAN0RFPTR7	<RSCAN0_base> + $0E74_H$
RSCAN0	Receive FIFO buffer access data field 0 register 7	RSCAN0RFDF07	<RSCAN0_base> + $0E78_H$
RSCAN0	Receive FIFO buffer access data field 1 register 7	RSCAN0RFDF17	<RSCAN0_base> + $0E7C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 0	RSCAN0CFID0	<RSCAN0_base> + $0E80_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 0	RSCAN0CFPTR0	<RSCAN0_base> + $0E84_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 0	RSCAN0CFDF00	<RSCAN0_base> + $0E88_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 0	RSCAN0CFDF10	<RSCAN0_base> + $0E8C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 1	RSCAN0CFID1	<RSCAN0_base> + $0E90_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 1	RSCAN0CFPTR1	<RSCAN0_base> + $0E94_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 1	RSCAN0CFDF01	<RSCAN0_base> + $0E98_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 1	RSCAN0CFDF11	<RSCAN0_base> + $0E9C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 2	RSCAN0CFID2	<RSCAN0_base> + $0EA0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 2	RSCAN0CFPTR2	<RSCAN0_base> + $0EA4_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 2	RSCAN0CFDF02	<RSCAN0_base> + $0EA8_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 2	RSCAN0CFDF12	<RSCAN0_base> + $0EAC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 3	RSCAN0CFID3	<RSCAN0_base> + $0EB0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 3	RSCAN0CFPTR3	<RSCAN0_base> + $0EB4_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 3	RSCAN0CFDF03	<RSCAN0_base> + $0EB8_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 3	RSCAN0CFDF13	<RSCAN0_base> + $0EBC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 4	RSCAN0CFID4	<RSCAN0_base> + $0EC0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 4	RSCAN0CFPTR4	<RSCAN0_base> + $0EC4_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 4	RSCAN0CFDF04	<RSCAN0_base> + $0EC8_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 4	RSCAN0CFDF14	<RSCAN0_base> + $0ECC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 5	RSCAN0CFID5	<RSCAN0_base> + $0ED0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 5	RSCAN0CFPTR5	<RSCAN0_base> + $0ED4_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 5	RSCAN0CFDF05	<RSCAN0_base> + $0ED8_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 5	RSCAN0CFDF15	<RSCAN0_base> + $0EDC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 6	RSCAN0CFID6	<RSCAN0_base> + $0EE0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 6	RSCAN0CFPTR6	<RSCAN0_base> + $0EE4_H$

Table 20.12 Registers (21/32)

Module	Register	Symbol	Address
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 6	RSCAN0CFDF06	<RSCAN0_base> + $0EE8_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 6	RSCAN0CFDF16	<RSCAN0_base> + $0EEC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 7	RSCAN0CFID7	<RSCAN0_base> + $0EF0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 7	RSCAN0CFPTR7	<RSCAN0_base> + $0EF4_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 7	RSCAN0CFDF07	<RSCAN0_base> + $0EF8_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 7	RSCAN0CFDF17	<RSCAN0_base> + $0EFC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 8	RSCAN0CFID8	<RSCAN0_base> + $0F00_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 8	RSCAN0CFPTR8	<RSCAN0_base> + $0F04_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 8	RSCAN0CFDF08	<RSCAN0_base> + $0F08_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 8	RSCAN0CFDF18	<RSCAN0_base> + $0F0C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 9	RSCAN0CFID9	<RSCAN0_base> + $0F10_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 9	RSCAN0CFPTR9	<RSCAN0_base> + $0F14_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 9	RSCAN0CFDF09	<RSCAN0_base> + $0F18_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 9	RSCAN0CFDF19	<RSCAN0_base> + $0F1C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 10	RSCAN0CFID10	<RSCAN0_base> + $0F20_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 10	RSCAN0CFPTR10	<RSCAN0_base> + $0F24_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 10	RSCAN0CFDF010	<RSCAN0_base> + $0F28_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 10	RSCAN0CFDF110	<RSCAN0_base> + $0F2C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 11	RSCAN0CFID11	<RSCAN0_base> + $0F30_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 11	RSCAN0CFPTR11	<RSCAN0_base> + $0F34_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 11	RSCAN0CFDF011	<RSCAN0_base> + $0F38_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 11	RSCAN0CFDF111	<RSCAN0_base> + $0F3C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 12	RSCAN0CFID12	<RSCAN0_base> + $0F40_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 12	RSCAN0CFPTR12	<RSCAN0_base> + $0F44_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 12	RSCAN0CFDF012	<RSCAN0_base> + $0F48_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 12	RSCAN0CFDF112	<RSCAN0_base> + $0F4C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 13	RSCAN0CFID13	<RSCAN0_base> + $0F50_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 13	RSCAN0CFPTR13	<RSCAN0_base> + $0F54_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 13	RSCAN0CFDF013	<RSCAN0_base> + $0F58_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 13	RSCAN0CFDF113	<RSCAN0_base> + $0F5C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 14	RSCAN0CFID14	<RSCAN0_base> + $0F60_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 14	RSCAN0CFPTR14	<RSCAN0_base> + $0F64_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 14	RSCAN0CFDF014	<RSCAN0_base> + $0F68_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 14	RSCAN0CFDF114	<RSCAN0_base> + $0F6C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 15	RSCAN0CFID15	<RSCAN0_base> + $0F70_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 15	RSCAN0CFPTR15	<RSCAN0_base> + $0F74_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 15	RSCAN0CFDF015	<RSCAN0_base> + $0F78_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 15	RSCAN0CFDF115	<RSCAN0_base> + $0F7C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 16	RSCAN0CFID16	<RSCAN0_base> + $0F80_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 16	RSCAN0CFPTR16	<RSCAN0_base> + $0F84_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 16	RSCAN0CFDF016	<RSCAN0_base> + $0F88_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 16	RSCAN0CFDF116	<RSCAN0_base> + $0F8C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 17	RSCAN0CFID17	<RSCAN0_base> + $0F90_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 17	RSCAN0CFPTR17	<RSCAN0_base> + $0F94_H$

Table 20.12 Registers (22/32)

Module	Register	Symbol	Address
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 17	RSCAN0CFDF017	<RSCAN0_base> + $0F98_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 17	RSCAN0CFDF117	<RSCAN0_base> + $0F9C_H$
RSCAN0	Transmit buffer ID register 0	RSCAN0TMID0	<RSCAN0_base> + 1000_H
RSCAN0	Transmit buffer pointer register 0	RSCAN0TMPTR0	<RSCAN0_base> + 1004_H
RSCAN0	Transmit buffer data field 0 register 0	RSCAN0TMDF00	<RSCAN0_base> + 1008_H
RSCAN0	Transmit buffer data field 1 register 0	RSCAN0TMDF10	<RSCAN0_base> + $100C_H$
RSCAN0	Transmit buffer ID register 1	RSCAN0TMID1	<RSCAN0_base> + 1010_H
RSCAN0	Transmit buffer pointer register 1	RSCAN0TMPTR1	<RSCAN0_base> + 1014_H
RSCAN0	Transmit buffer data field 0 register 1	RSCAN0TMDF01	<RSCAN0_base> + 1018_H
RSCAN0	Transmit buffer data field 1 register 1	RSCAN0TMDF11	<RSCAN0_base> + $101C_H$
RSCAN0	Transmit buffer ID register 2	RSCAN0TMID2	<RSCAN0_base> + 1020_H
RSCAN0	Transmit buffer pointer register 2	RSCAN0TMPTR2	<RSCAN0_base> + 1024_H
RSCAN0	Transmit buffer data field 0 register 2	RSCAN0TMDF02	<RSCAN0_base> + 1028_H
RSCAN0	Transmit buffer data field 1 register 2	RSCAN0TMDF12	<RSCAN0_base> + $102C_H$
RSCAN0	Transmit buffer ID register 3	RSCAN0TMID3	<RSCAN0_base> + 1030_H
RSCAN0	Transmit buffer pointer register 3	RSCAN0TMPTR3	<RSCAN0_base> + 1034_H
RSCAN0	Transmit buffer data field 0 register 3	RSCAN0TMDF03	<RSCAN0_base> + 1038_H
RSCAN0	Transmit buffer data field 1 register 3	RSCAN0TMDF13	<RSCAN0_base> + $103C_H$
RSCAN0	Transmit buffer ID register 4	RSCAN0TMID4	<RSCAN0_base> + 1040_H
RSCAN0	Transmit buffer pointer register 4	RSCAN0TMPTR4	<RSCAN0_base> + 1044_H
RSCAN0	Transmit buffer data field 0 register 4	RSCAN0TMDF04	<RSCAN0_base> + 1048_H
RSCAN0	Transmit buffer data field 1 register 4	RSCAN0TMDF14	<RSCAN0_base> + $104C_H$
RSCAN0	Transmit buffer ID register 5	RSCAN0TMID5	<RSCAN0_base> + 1050_H
RSCAN0	Transmit buffer pointer register 5	RSCAN0TMPTR5	<RSCAN0_base> + 1054_H
RSCAN0	Transmit buffer data field 0 register 5	RSCAN0TMDF05	<RSCAN0_base> + 1058_H
RSCAN0	Transmit buffer data field 1 register 5	RSCAN0TMDF15	<RSCAN0_base> + $105C_H$
RSCAN0	Transmit buffer ID register 6	RSCAN0TMID6	<RSCAN0_base> + 1060_H
RSCAN0	Transmit buffer pointer register 6	RSCAN0TMPTR6	<RSCAN0_base> + 1064_H
RSCAN0	Transmit buffer data field 0 register 6	RSCAN0TMDF06	<RSCAN0_base> + 1068_H
RSCAN0	Transmit buffer data field 1 register 6	RSCAN0TMDF16	<RSCAN0_base> + $106C_H$
RSCAN0	Transmit buffer ID register 7	RSCAN0TMID7	<RSCAN0_base> + 1070_H
RSCAN0	Transmit buffer pointer register 7	RSCAN0TMPTR7	<RSCAN0_base> + 1074_H
RSCAN0	Transmit buffer data field 0 register 7	RSCAN0TMDF07	<RSCAN0_base> + 1078_H
RSCAN0	Transmit buffer data field 1 register 7	RSCAN0TMDF17	<RSCAN0_base> + $107C_H$
RSCAN0	Transmit buffer ID register 8	RSCAN0TMID8	<RSCAN0_base> + 1080_H
RSCAN0	Transmit buffer pointer register 8	RSCAN0TMPTR8	<RSCAN0_base> + 1084_H
RSCAN0	Transmit buffer data field 0 register 8	RSCAN0TMDF08	<RSCAN0_base> + 1088_H
RSCAN0	Transmit buffer data field 1 register 8	RSCAN0TMDF18	<RSCAN0_base> + $108C_H$
RSCAN0	Transmit buffer ID register 9	RSCAN0TMID9	<RSCAN0_base> + 1090_H
RSCAN0	Transmit buffer pointer register 9	RSCAN0TMPTR9	<RSCAN0_base> + 1094_H
RSCAN0	Transmit buffer data field 0 register 9	RSCAN0TMDF09	<RSCAN0_base> + 1098_H
RSCAN0	Transmit buffer data field 1 register 9	RSCAN0TMDF19	<RSCAN0_base> + $109C_H$
RSCAN0	Transmit buffer ID register 10	RSCAN0TMID10	<RSCAN0_base> + $10A0_H$
RSCAN0	Transmit buffer pointer register 10	RSCAN0TMPTR10	<RSCAN0_base> + $10A4_H$

Table 20.12 Registers (23/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 10	RSCAN0TMDF010	<RSCAN0_base> + $10A8_H$
RSCAN0	Transmit buffer data field 1 register 10	RSCAN0TMDF110	<RSCAN0_base> + $10AC_H$
RSCAN0	Transmit buffer ID register 11	RSCAN0TMID11	<RSCAN0_base> + $10B0_H$
RSCAN0	Transmit buffer pointer register 11	RSCAN0TMPTR11	<RSCAN0_base> + $10B4_H$
RSCAN0	Transmit buffer data field 0 register 11	RSCAN0TMDF011	<RSCAN0_base> + $10B8_H$
RSCAN0	Transmit buffer data field 1 register 11	RSCAN0TMDF111	<RSCAN0_base> + $10BC_H$
RSCAN0	Transmit buffer ID register 12	RSCAN0TMID12	<RSCAN0_base> + $10C0_H$
RSCAN0	Transmit buffer pointer register 12	RSCAN0TMPTR12	<RSCAN0_base> + $10C4_H$
RSCAN0	Transmit buffer data field 0 register 12	RSCAN0TMDF012	<RSCAN0_base> + $10C8_H$
RSCAN0	Transmit buffer data field 1 register 12	RSCAN0TMDF112	<RSCAN0_base> + $10CC_H$
RSCAN0	Transmit buffer ID register 13	RSCAN0TMID13	<RSCAN0_base> + $10D0_H$
RSCAN0	Transmit buffer pointer register 13	RSCAN0TMPTR13	<RSCAN0_base> + $10D4_H$
RSCAN0	Transmit buffer data field 0 register 13	RSCAN0TMDF013	<RSCAN0_base> + $10D8_H$
RSCAN0	Transmit buffer data field 1 register 13	RSCAN0TMDF113	<RSCAN0_base> + $10DC_H$
RSCAN0	Transmit buffer ID register 14	RSCAN0TMID14	<RSCAN0_base> + $10E0_H$
RSCAN0	Transmit buffer pointer register 14	RSCAN0TMPTR14	<RSCAN0_base> + $10E4_H$
RSCAN0	Transmit buffer data field 0 register 14	RSCAN0TMDF014	<RSCAN0_base> + $10E8_H$
RSCAN0	Transmit buffer data field 1 register 14	RSCAN0TMDF114	<RSCAN0_base> + $10EC_H$
RSCAN0	Transmit buffer ID register 15	RSCAN0TMID15	<RSCAN0_base> + $10F0_H$
RSCAN0	Transmit buffer pointer register 15	RSCAN0TMPTR15	<RSCAN0_base> + $10F4_H$
RSCAN0	Transmit buffer data field 0 register 15	RSCAN0TMDF015	<RSCAN0_base> + $10F8_H$
RSCAN0	Transmit buffer data field 1 register 15	RSCAN0TMDF115	<RSCAN0_base> + $10FC_H$
RSCAN0	Transmit buffer ID register 16	RSCAN0TMID16	<RSCAN0_base> + 1100_H
RSCAN0	Transmit buffer pointer register 16	RSCAN0TMPTR16	<RSCAN0_base> + 1104_H
RSCAN0	Transmit buffer data field 0 register 16	RSCAN0TMDF016	<RSCAN0_base> + 1108_H
RSCAN0	Transmit buffer data field 1 register 16	RSCAN0TMDF116	<RSCAN0_base> + $110C_H$
RSCAN0	Transmit buffer ID register 17	RSCAN0TMID17	<RSCAN0_base> + 1110_H
RSCAN0	Transmit buffer pointer register 17	RSCAN0TMPTR17	<RSCAN0_base> + 1114_H
RSCAN0	Transmit buffer data field 0 register 17	RSCAN0TMDF017	<RSCAN0_base> + 1118_H
RSCAN0	Transmit buffer data field 1 register 17	RSCAN0TMDF117	<RSCAN0_base> + $111C_H$
RSCAN0	Transmit buffer ID register 18	RSCAN0TMID18	<RSCAN0_base> + 1120_H
RSCAN0	Transmit buffer pointer register 18	RSCAN0TMPTR18	<RSCAN0_base> + 1124_H
RSCAN0	Transmit buffer data field 0 register 18	RSCAN0TMDF018	<RSCAN0_base> + 1128_H
RSCAN0	Transmit buffer data field 1 register 18	RSCAN0TMDF118	<RSCAN0_base> + $112C_H$
RSCAN0	Transmit buffer ID register 19	RSCAN0TMID19	<RSCAN0_base> + 1130_H
RSCAN0	Transmit buffer pointer register 19	RSCAN0TMPTR19	<RSCAN0_base> + 1134_H
RSCAN0	Transmit buffer data field 0 register 19	RSCAN0TMDF019	<RSCAN0_base> + 1138_H
RSCAN0	Transmit buffer data field 1 register 19	RSCAN0TMDF119	<RSCAN0_base> + $113C_H$
RSCAN0	Transmit buffer ID register 20	RSCAN0TMID20	<RSCAN0_base> + 1140_H
RSCAN0	Transmit buffer pointer register 20	RSCAN0TMPTR20	<RSCAN0_base> + 1144_H
RSCAN0	Transmit buffer data field 0 register 20	RSCAN0TMDF020	<RSCAN0_base> + 1148_H
RSCAN0	Transmit buffer data field 1 register 20	RSCAN0TMDF120	<RSCAN0_base> + $114C_H$
RSCAN0	Transmit buffer ID register 21	RSCAN0TMID21	<RSCAN0_base> + 1150_H
RSCAN0	Transmit buffer pointer register 21	RSCAN0TMPTR21	<RSCAN0_base> + 1154_H

Table 20.12 Registers (24/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 21	RSCAN0TMDF021	<RSCAN0_base> + 1158_H
RSCAN0	Transmit buffer data field 1 register 21	RSCAN0TMDF121	<RSCAN0_base> + $115C_H$
RSCAN0	Transmit buffer ID register 22	RSCAN0TMID22	<RSCAN0_base> + 1160_H
RSCAN0	Transmit buffer pointer register 22	RSCAN0TMPTR22	<RSCAN0_base> + 1164_H
RSCAN0	Transmit buffer data field 0 register 22	RSCAN0TMDF022	<RSCAN0_base> + 1168_H
RSCAN0	Transmit buffer data field 1 register 22	RSCAN0TMDF122	<RSCAN0_base> + $116C_H$
RSCAN0	Transmit buffer ID register 23	RSCAN0TMID23	<RSCAN0_base> + 1170_H
RSCAN0	Transmit buffer pointer register 23	RSCAN0TMPTR23	<RSCAN0_base> + 1174_H
RSCAN0	Transmit buffer data field 0 register 23	RSCAN0TMDF023	<RSCAN0_base> + 1178_H
RSCAN0	Transmit buffer data field 1 register 23	RSCAN0TMDF123	<RSCAN0_base> + $117C_H$
RSCAN0	Transmit buffer ID register 24	RSCAN0TMID24	<RSCAN0_base> + 1180_H
RSCAN0	Transmit buffer pointer register 24	RSCAN0TMPTR24	<RSCAN0_base> + 1184_H
RSCAN0	Transmit buffer data field 0 register 24	RSCAN0TMDF024	<RSCAN0_base> + 1188_H
RSCAN0	Transmit buffer data field 1 register 24	RSCAN0TMDF124	<RSCAN0_base> + $118C_H$
RSCAN0	Transmit buffer ID register 25	RSCAN0TMID25	<RSCAN0_base> + 1190_H
RSCAN0	Transmit buffer pointer register 25	RSCAN0TMPTR25	<RSCAN0_base> + 1194_H
RSCAN0	Transmit buffer data field 0 register 25	RSCAN0TMDF025	<RSCAN0_base> + 1198_H
RSCAN0	Transmit buffer data field 1 register 25	RSCAN0TMDF125	<RSCAN0_base> + $119C_H$
RSCAN0	Transmit buffer ID register 26	RSCAN0TMID26	<RSCAN0_base> + $11A0_H$
RSCAN0	Transmit buffer pointer register 26	RSCAN0TMPTR26	<RSCAN0_base> + $11A4_H$
RSCAN0	Transmit buffer data field 0 register 26	RSCAN0TMDF026	<RSCAN0_base> + $11A8_H$
RSCAN0	Transmit buffer data field 1 register 26	RSCAN0TMDF126	<RSCAN0_base> + $11AC_H$
RSCAN0	Transmit buffer ID register 27	RSCAN0TMID27	<RSCAN0_base> + $11B0_H$
RSCAN0	Transmit buffer pointer register 27	RSCAN0TMPTR27	<RSCAN0_base> + $11B4_H$
RSCAN0	Transmit buffer data field 0 register 27	RSCAN0TMDF027	<RSCAN0_base> + $11B8_H$
RSCAN0	Transmit buffer data field 1 register 27	RSCAN0TMDF127	<RSCAN0_base> + $11BC_H$
RSCAN0	Transmit buffer ID register 28	RSCAN0TMID28	<RSCAN0_base> + $11C0_H$
RSCAN0	Transmit buffer pointer register 28	RSCAN0TMPTR28	<RSCAN0_base> + $11C4_H$
RSCAN0	Transmit buffer data field 0 register 28	RSCAN0TMDF028	<RSCAN0_base> + $11C8_H$
RSCAN0	Transmit buffer data field 1 register 28	RSCAN0TMDF128	<RSCAN0_base> + $11CC_H$
RSCAN0	Transmit buffer ID register 29	RSCAN0TMID29	<RSCAN0_base> + $11D0_H$
RSCAN0	Transmit buffer pointer register 29	RSCAN0TMPTR29	<RSCAN0_base> + $11D4_H$
RSCAN0	Transmit buffer data field 0 register 29	RSCAN0TMDF029	<RSCAN0_base> + $11D8_H$
RSCAN0	Transmit buffer data field 1 register 29	RSCAN0TMDF129	<RSCAN0_base> + $11DC_H$
RSCAN0	Transmit buffer ID register 30	RSCAN0TMID30	<RSCAN0_base> + $11E0_H$
RSCAN0	Transmit buffer pointer register 30	RSCAN0TMPTR30	<RSCAN0_base> + $11E4_H$
RSCAN0	Transmit buffer data field 0 register 30	RSCAN0TMDF030	<RSCAN0_base> + $11E8_H$
RSCAN0	Transmit buffer data field 1 register 30	RSCAN0TMDF130	<RSCAN0_base> + $11EC_H$
RSCAN0	Transmit buffer ID register 31	RSCAN0TMID31	<RSCAN0_base> + $11F0_H$
RSCAN0	Transmit buffer pointer register 31	RSCAN0TMPTR31	<RSCAN0_base> + $11F4_H$
RSCAN0	Transmit buffer data field 0 register 31	RSCAN0TMDF031	<RSCAN0_base> + $11F8_H$
RSCAN0	Transmit buffer data field 1 register 31	RSCAN0TMDF131	<RSCAN0_base> + $11FC_H$
RSCAN0	Transmit buffer ID register 32	RSCAN0TMID32	<RSCAN0_base> + 1200_H
RSCAN0	Transmit buffer pointer register 32	RSCAN0TMPTR32	<RSCAN0_base> + 1204_H

Table 20.12 Registers (25/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 32	RSCAN0TMDF032	<RSCAN0_base> + 1208_H
RSCAN0	Transmit buffer data field 1 register 32	RSCAN0TMDF132	<RSCAN0_base> + $120C_H$
RSCAN0	Transmit buffer ID register 33	RSCAN0TMID33	<RSCAN0_base> + 1210_H
RSCAN0	Transmit buffer pointer register 33	RSCAN0TMPTR33	<RSCAN0_base> + 1214_H
RSCAN0	Transmit buffer data field 0 register 33	RSCAN0TMDF033	<RSCAN0_base> + 1218_H
RSCAN0	Transmit buffer data field 1 register 33	RSCAN0TMDF133	<RSCAN0_base> + $121C_H$
RSCAN0	Transmit buffer ID register 34	RSCAN0TMID34	<RSCAN0_base> + 1220_H
RSCAN0	Transmit buffer pointer register 34	RSCAN0TMPTR34	<RSCAN0_base> + 1224_H
RSCAN0	Transmit buffer data field 0 register 34	RSCAN0TMDF034	<RSCAN0_base> + 1228_H
RSCAN0	Transmit buffer data field 1 register 34	RSCAN0TMDF134	<RSCAN0_base> + $122C_H$
RSCAN0	Transmit buffer ID register 35	RSCAN0TMID35	<RSCAN0_base> + 1230_H
RSCAN0	Transmit buffer pointer register 35	RSCAN0TMPTR35	<RSCAN0_base> + 1234_H
RSCAN0	Transmit buffer data field 0 register 35	RSCAN0TMDF035	<RSCAN0_base> + 1238_H
RSCAN0	Transmit buffer data field 1 register 35	RSCAN0TMDF135	<RSCAN0_base> + $123C_H$
RSCAN0	Transmit buffer ID register 36	RSCAN0TMID36	<RSCAN0_base> + 1240_H
RSCAN0	Transmit buffer pointer register 36	RSCAN0TMPTR36	<RSCAN0_base> + 1244_H
RSCAN0	Transmit buffer data field 0 register 36	RSCAN0TMDF036	<RSCAN0_base> + 1248_H
RSCAN0	Transmit buffer data field 1 register 36	RSCAN0TMDF136	<RSCAN0_base> + $124C_H$
RSCAN0	Transmit buffer ID register 37	RSCAN0TMID37	<RSCAN0_base> + 1250_H
RSCAN0	Transmit buffer pointer register 37	RSCAN0TMPTR37	<RSCAN0_base> + 1254_H
RSCAN0	Transmit buffer data field 0 register 37	RSCAN0TMDF037	<RSCAN0_base> + 1258_H
RSCAN0	Transmit buffer data field 1 register 37	RSCAN0TMDF137	<RSCAN0_base> + $125C_H$
RSCAN0	Transmit buffer ID register 38	RSCAN0TMID38	<RSCAN0_base> + 1260_H
RSCAN0	Transmit buffer pointer register 38	RSCAN0TMPTR38	<RSCAN0_base> + 1264_H
RSCAN0	Transmit buffer data field 0 register 38	RSCAN0TMDF038	<RSCAN0_base> + 1268_H
RSCAN0	Transmit buffer data field 1 register 38	RSCAN0TMDF138	<RSCAN0_base> + $126C_H$
RSCAN0	Transmit buffer ID register 39	RSCAN0TMID39	<RSCAN0_base> + 1270_H
RSCAN0	Transmit buffer pointer register 39	RSCAN0TMPTR39	<RSCAN0_base> + 1274_H
RSCAN0	Transmit buffer data field 0 register 39	RSCAN0TMDF039	<RSCAN0_base> + 1278_H
RSCAN0	Transmit buffer data field 1 register 39	RSCAN0TMDF139	<RSCAN0_base> + $127C_H$
RSCAN0	Transmit buffer ID register 40	RSCAN0TMID40	<RSCAN0_base> + 1280_H
RSCAN0	Transmit buffer pointer register 40	RSCAN0TMPTR40	<RSCAN0_base> + 1284_H
RSCAN0	Transmit buffer data field 0 register 40	RSCAN0TMDF040	<RSCAN0_base> + 1288_H
RSCAN0	Transmit buffer data field 1 register 40	RSCAN0TMDF140	<RSCAN0_base> + $128C_H$
RSCAN0	Transmit buffer ID register 41	RSCAN0TMID41	<RSCAN0_base> + 1290_H
RSCAN0	Transmit buffer pointer register 41	RSCAN0TMPTR41	<RSCAN0_base> + 1294_H
RSCAN0	Transmit buffer data field 0 register 41	RSCAN0TMDF041	<RSCAN0_base> + 1298_H
RSCAN0	Transmit buffer data field 1 register 41	RSCAN0TMDF141	<RSCAN0_base> + $129C_H$
RSCAN0	Transmit buffer ID register 42	RSCAN0TMID42	<RSCAN0_base> + $12A0_H$
RSCAN0	Transmit buffer pointer register 42	RSCAN0TMPTR42	<RSCAN0_base> + $12A4_H$
RSCAN0	Transmit buffer data field 0 register 42	RSCAN0TMDF042	<RSCAN0_base> + $12A8_H$
RSCAN0	Transmit buffer data field 1 register 42	RSCAN0TMDF142	<RSCAN0_base> + $12AC_H$
RSCAN0	Transmit buffer ID register 43	RSCAN0TMID43	<RSCAN0_base> + $12B0_H$
RSCAN0	Transmit buffer pointer register 43	RSCAN0TMPTR43	<RSCAN0_base> + $12B4_H$

Table 20.12 Registers (26/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 43	RSCAN0TMDF043	<RSCAN0_base> + $12B8_H$
RSCAN0	Transmit buffer data field 1 register 43	RSCAN0TMDF143	<RSCAN0_base> + $12BC_H$
RSCAN0	Transmit buffer ID register 44	RSCAN0TMID44	<RSCAN0_base> + $12C0_H$
RSCAN0	Transmit buffer pointer register 44	RSCAN0TMPTR44	<RSCAN0_base> + $12C4_H$
RSCAN0	Transmit buffer data field 0 register 44	RSCAN0TMDF044	<RSCAN0_base> + $12C8_H$
RSCAN0	Transmit buffer data field 1 register 44	RSCAN0TMDF144	<RSCAN0_base> + $12CC_H$
RSCAN0	Transmit buffer ID register 45	RSCAN0TMID45	<RSCAN0_base> + $12D0_H$
RSCAN0	Transmit buffer pointer register 45	RSCAN0TMPTR45	<RSCAN0_base> + $12D4_H$
RSCAN0	Transmit buffer data field 0 register 45	RSCAN0TMDF045	<RSCAN0_base> + $12D8_H$
RSCAN0	Transmit buffer data field 1 register 45	RSCAN0TMDF145	<RSCAN0_base> + $12DC_H$
RSCAN0	Transmit buffer ID register 46	RSCAN0TMID46	<RSCAN0_base> + $12E0_H$
RSCAN0	Transmit buffer pointer register 46	RSCAN0TMPTR46	<RSCAN0_base> + $12E4_H$
RSCAN0	Transmit buffer data field 0 register 46	RSCAN0TMDF046	<RSCAN0_base> + $12E8_H$
RSCAN0	Transmit buffer data field 1 register 46	RSCAN0TMDF146	<RSCAN0_base> + $12EC_H$
RSCAN0	Transmit buffer ID register 47	RSCAN0TMID47	<RSCAN0_base> + $12F0_H$
RSCAN0	Transmit buffer pointer register 47	RSCAN0TMPTR47	<RSCAN0_base> + $12F4_H$
RSCAN0	Transmit buffer data field 0 register 47	RSCAN0TMDF047	<RSCAN0_base> + $12F8_H$
RSCAN0	Transmit buffer data field 1 register 47	RSCAN0TMDF147	<RSCAN0_base> + $12FC_H$
RSCAN0	Transmit buffer ID register 48	RSCAN0TMID48	<RSCAN0_base> + 1300_H
RSCAN0	Transmit buffer pointer register 48	RSCAN0TMPTR48	<RSCAN0_base> + 1304_H
RSCAN0	Transmit buffer data field 0 register 48	RSCAN0TMDF048	<RSCAN0_base> + 1308_H
RSCAN0	Transmit buffer data field 1 register 48	RSCAN0TMDF148	<RSCAN0_base> + $130C_H$
RSCAN0	Transmit buffer ID register 49	RSCAN0TMID49	<RSCAN0_base> + 1310_H
RSCAN0	Transmit buffer pointer register 49	RSCAN0TMPTR49	<RSCAN0_base> + 1314_H
RSCAN0	Transmit buffer data field 0 register 49	RSCAN0TMDF049	<RSCAN0_base> + 1318_H
RSCAN0	Transmit buffer data field 1 register 49	RSCAN0TMDF149	<RSCAN0_base> + $131C_H$
RSCAN0	Transmit buffer ID register 50	RSCAN0TMID50	<RSCAN0_base> + 1320_H
RSCAN0	Transmit buffer pointer register 50	RSCAN0TMPTR50	<RSCAN0_base> + 1324_H
RSCAN0	Transmit buffer data field 0 register 50	RSCAN0TMDF050	<RSCAN0_base> + 1328_H
RSCAN0	Transmit buffer data field 1 register 50	RSCAN0TMDF150	<RSCAN0_base> + $132C_H$
RSCAN0	Transmit buffer ID register 51	RSCAN0TMID51	<RSCAN0_base> + 1330_H
RSCAN0	Transmit buffer pointer register 51	RSCAN0TMPTR51	<RSCAN0_base> + 1334_H
RSCAN0	Transmit buffer data field 0 register 51	RSCAN0TMDF051	<RSCAN0_base> + 1338_H
RSCAN0	Transmit buffer data field 1 register 51	RSCAN0TMDF151	<RSCAN0_base> + $133C_H$
RSCAN0	Transmit buffer ID register 52	RSCAN0TMID52	<RSCAN0_base> + 1340_H
RSCAN0	Transmit buffer pointer register 52	RSCAN0TMPTR52	<RSCAN0_base> + 1344_H
RSCAN0	Transmit buffer data field 0 register 52	RSCAN0TMDF052	<RSCAN0_base> + 1348_H
RSCAN0	Transmit buffer data field 1 register 52	RSCAN0TMDF152	<RSCAN0_base> + $134C_H$
RSCAN0	Transmit buffer ID register 53	RSCAN0TMID53	<RSCAN0_base> + 1350_H
RSCAN0	Transmit buffer pointer register 53	RSCAN0TMPTR53	<RSCAN0_base> + 1354_H
RSCAN0	Transmit buffer data field 0 register 53	RSCAN0TMDF053	<RSCAN0_base> + 1358_H
RSCAN0	Transmit buffer data field 1 register 53	RSCAN0TMDF153	<RSCAN0_base> + $135C_H$
RSCAN0	Transmit buffer ID register 54	RSCAN0TMID54	<RSCAN0_base> + 1360_H
RSCAN0	Transmit buffer pointer register 54	RSCAN0TMPTR54	<RSCAN0_base> + 1364_H

Table 20.12 Registers (27/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 54	RSCAN0TMDF054	<RSCAN0_base> + 1368_H
RSCAN0	Transmit buffer data field 1 register 54	RSCAN0TMDF154	<RSCAN0_base> + $136C_H$
RSCAN0	Transmit buffer ID register 55	RSCAN0TMID55	<RSCAN0_base> + 1370_H
RSCAN0	Transmit buffer pointer register 55	RSCAN0TMPTR55	<RSCAN0_base> + 1374_H
RSCAN0	Transmit buffer data field 0 register 55	RSCAN0TMDF055	<RSCAN0_base> + 1378_H
RSCAN0	Transmit buffer data field 1 register 55	RSCAN0TMDF155	<RSCAN0_base> + $137C_H$
RSCAN0	Transmit buffer ID register 56	RSCAN0TMID56	<RSCAN0_base> + 1380_H
RSCAN0	Transmit buffer pointer register 56	RSCAN0TMPTR56	<RSCAN0_base> + 1384_H
RSCAN0	Transmit buffer data field 0 register 56	RSCAN0TMDF056	<RSCAN0_base> + 1388_H
RSCAN0	Transmit buffer data field 1 register 56	RSCAN0TMDF156	<RSCAN0_base> + $138C_H$
RSCAN0	Transmit buffer ID register 57	RSCAN0TMID57	<RSCAN0_base> + 1390_H
RSCAN0	Transmit buffer pointer register 57	RSCAN0TMPTR57	<RSCAN0_base> + 1394_H
RSCAN0	Transmit buffer data field 0 register 57	RSCAN0TMDF057	<RSCAN0_base> + 1398_H
RSCAN0	Transmit buffer data field 1 register 57	RSCAN0TMDF157	<RSCAN0_base> + $139C_H$
RSCAN0	Transmit buffer ID register 58	RSCAN0TMID58	<RSCAN0_base> + $13A0_H$
RSCAN0	Transmit buffer pointer register 58	RSCAN0TMPTR58	<RSCAN0_base> + $13A4_H$
RSCAN0	Transmit buffer data field 0 register 58	RSCAN0TMDF058	<RSCAN0_base> + $13A8_H$
RSCAN0	Transmit buffer data field 1 register 58	RSCAN0TMDF158	<RSCAN0_base> + $13AC_H$
RSCAN0	Transmit buffer ID register 59	RSCAN0TMID59	<RSCAN0_base> + $13B0_H$
RSCAN0	Transmit buffer pointer register 59	RSCAN0TMPTR59	<RSCAN0_base> + $13B4_H$
RSCAN0	Transmit buffer data field 0 register 59	RSCAN0TMDF059	<RSCAN0_base> + $13B8_H$
RSCAN0	Transmit buffer data field 1 register 59	RSCAN0TMDF159	<RSCAN0_base> + $13BC_H$
RSCAN0	Transmit buffer ID register 60	RSCAN0TMID60	<RSCAN0_base> + $13C0_H$
RSCAN0	Transmit buffer pointer register 60	RSCAN0TMPTR60	<RSCAN0_base> + $13C4_H$
RSCAN0	Transmit buffer data field 0 register 60	RSCAN0TMDF060	<RSCAN0_base> + $13C8_H$
RSCAN0	Transmit buffer data field 1 register 60	RSCAN0TMDF160	<RSCAN0_base> + $13CC_H$
RSCAN0	Transmit buffer ID register 61	RSCAN0TMID61	<RSCAN0_base> + $13D0_H$
RSCAN0	Transmit buffer pointer register 61	RSCAN0TMPTR61	<RSCAN0_base> + $13D4_H$
RSCAN0	Transmit buffer data field 0 register 61	RSCAN0TMDF061	<RSCAN0_base> + $13D8_H$
RSCAN0	Transmit buffer data field 1 register 61	RSCAN0TMDF161	<RSCAN0_base> + $13DC_H$
RSCAN0	Transmit buffer ID register 62	RSCAN0TMID62	<RSCAN0_base> + $13E0_H$
RSCAN0	Transmit buffer pointer register 62	RSCAN0TMPTR62	<RSCAN0_base> + $13E4_H$
RSCAN0	Transmit buffer data field 0 register 62	RSCAN0TMDF062	<RSCAN0_base> + $13E8_H$
RSCAN0	Transmit buffer data field 1 register 62	RSCAN0TMDF162	<RSCAN0_base> + $13EC_H$
RSCAN0	Transmit buffer ID register 63	RSCAN0TMID63	<RSCAN0_base> + $13F0_H$
RSCAN0	Transmit buffer pointer register 63	RSCAN0TMPTR63	<RSCAN0_base> + $13F4_H$
RSCAN0	Transmit buffer data field 0 register 63	RSCAN0TMDF063	<RSCAN0_base> + $13F8_H$
RSCAN0	Transmit buffer data field 1 register 63	RSCAN0TMDF163	<RSCAN0_base> + $13FC_H$
RSCAN0	Transmit buffer ID register 64	RSCAN0TMID64	<RSCAN0_base> + 1400_H
RSCAN0	Transmit buffer pointer register 64	RSCAN0TMPTR64	<RSCAN0_base> + 1404_H
RSCAN0	Transmit buffer data field 0 register 64	RSCAN0TMDF064	<RSCAN0_base> + 1408_H
RSCAN0	Transmit buffer data field 1 register 64	RSCAN0TMDF164	<RSCAN0_base> + $140C_H$
RSCAN0	Transmit buffer ID register 65	RSCAN0TMID65	<RSCAN0_base> + 1410_H
RSCAN0	Transmit buffer pointer register 65	RSCAN0TMPTR65	<RSCAN0_base> + 1414_H

Table 20.12 Registers (28/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 65	RSCAN0TMDF065	<RSCAN0_base> + 1418_H
RSCAN0	Transmit buffer data field 1 register 65	RSCAN0TMDF165	<RSCAN0_base> + $141C_H$
RSCAN0	Transmit buffer ID register 66	RSCAN0TMID66	<RSCAN0_base> + 1420_H
RSCAN0	Transmit buffer pointer register 66	RSCAN0TMPTR66	<RSCAN0_base> + 1424_H
RSCAN0	Transmit buffer data field 0 register 66	RSCAN0TMDF066	<RSCAN0_base> + 1428_H
RSCAN0	Transmit buffer data field 1 register 66	RSCAN0TMDF166	<RSCAN0_base> + $142C_H$
RSCAN0	Transmit buffer ID register 67	RSCAN0TMID67	<RSCAN0_base> + 1430_H
RSCAN0	Transmit buffer pointer register 67	RSCAN0TMPTR67	<RSCAN0_base> + 1434_H
RSCAN0	Transmit buffer data field 0 register 67	RSCAN0TMDF067	<RSCAN0_base> + 1438_H
RSCAN0	Transmit buffer data field 1 register 67	RSCAN0TMDF167	<RSCAN0_base> + $143C_H$
RSCAN0	Transmit buffer ID register 68	RSCAN0TMID68	<RSCAN0_base> + 1440_H
RSCAN0	Transmit buffer pointer register 68	RSCAN0TMPTR68	<RSCAN0_base> + 1444_H
RSCAN0	Transmit buffer data field 0 register 68	RSCAN0TMDF068	<RSCAN0_base> + 1448_H
RSCAN0	Transmit buffer data field 1 register 68	RSCAN0TMDF168	<RSCAN0_base> + $144C_H$
RSCAN0	Transmit buffer ID register 69	RSCAN0TMID69	<RSCAN0_base> + 1450_H
RSCAN0	Transmit buffer pointer register 69	RSCAN0TMPTR69	<RSCAN0_base> + 1454_H
RSCAN0	Transmit buffer data field 0 register 69	RSCAN0TMDF069	<RSCAN0_base> + 1458_H
RSCAN0	Transmit buffer data field 1 register 69	RSCAN0TMDF169	<RSCAN0_base> + $145C_H$
RSCAN0	Transmit buffer ID register 70	RSCAN0TMID70	<RSCAN0_base> + 1460_H
RSCAN0	Transmit buffer pointer register 70	RSCAN0TMPTR70	<RSCAN0_base> + 1464_H
RSCAN0	Transmit buffer data field 0 register 70	RSCAN0TMDF070	<RSCAN0_base> + 1468_H
RSCAN0	Transmit buffer data field 1 register 70	RSCAN0TMDF170	<RSCAN0_base> + $146C_H$
RSCAN0	Transmit buffer ID register 71	RSCAN0TMID71	<RSCAN0_base> + 1470_H
RSCAN0	Transmit buffer pointer register 71	RSCAN0TMPTR71	<RSCAN0_base> + 1474_H
RSCAN0	Transmit buffer data field 0 register 71	RSCAN0TMDF071	<RSCAN0_base> + 1478_H
RSCAN0	Transmit buffer data field 1 register 71	RSCAN0TMDF171	<RSCAN0_base> + $147C_H$
RSCAN0	Transmit buffer ID register 72	RSCAN0TMID72	<RSCAN0_base> + 1480_H
RSCAN0	Transmit buffer pointer register 72	RSCAN0TMPTR72	<RSCAN0_base> + 1484_H
RSCAN0	Transmit buffer data field 0 register 72	RSCAN0TMDF072	<RSCAN0_base> + 1488_H
RSCAN0	Transmit buffer data field 1 register 72	RSCAN0TMDF172	<RSCAN0_base> + $148C_H$
RSCAN0	Transmit buffer ID register 73	RSCAN0TMID73	<RSCAN0_base> + 1490_H
RSCAN0	Transmit buffer pointer register 73	RSCAN0TMPTR73	<RSCAN0_base> + 1494_H
RSCAN0	Transmit buffer data field 0 register 73	RSCAN0TMDF073	<RSCAN0_base> + 1498_H
RSCAN0	Transmit buffer data field 1 register 73	RSCAN0TMDF173	<RSCAN0_base> + $149C_H$
RSCAN0	Transmit buffer ID register 74	RSCAN0TMID74	<RSCAN0_base> + $14A0_H$
RSCAN0	Transmit buffer pointer register 74	RSCAN0TMPTR74	<RSCAN0_base> + $14A4_H$
RSCAN0	Transmit buffer data field 0 register 74	RSCAN0TMDF074	<RSCAN0_base> + $14A8_H$
RSCAN0	Transmit buffer data field 1 register 74	RSCAN0TMDF174	<RSCAN0_base> + $14AC_H$
RSCAN0	Transmit buffer ID register 75	RSCAN0TMID75	<RSCAN0_base> + $14B0_H$
RSCAN0	Transmit buffer pointer register 75	RSCAN0TMPTR75	<RSCAN0_base> + $14B4_H$
RSCAN0	Transmit buffer data field 0 register 75	RSCAN0TMDF075	<RSCAN0_base> + $14B8_H$
RSCAN0	Transmit buffer data field 1 register 75	RSCAN0TMDF175	<RSCAN0_base> + $14BC_H$
RSCAN0	Transmit buffer ID register 76	RSCAN0TMID76	<RSCAN0_base> + $14C0_H$
RSCAN0	Transmit buffer pointer register 76	RSCAN0TMPTR76	<RSCAN0_base> + $14C4_H$

Table 20.12 Registers (29/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 76	RSCAN0TMDF076	<RSCAN0_base> + $14C8_H$
RSCAN0	Transmit buffer data field 1 register 76	RSCAN0TMDF176	<RSCAN0_base> + $14CC_H$
RSCAN0	Transmit buffer ID register 77	RSCAN0TMID77	<RSCAN0_base> + $14D0_H$
RSCAN0	Transmit buffer pointer register 77	RSCAN0TMPTR77	<RSCAN0_base> + $14D4_H$
RSCAN0	Transmit buffer data field 0 register 77	RSCAN0TMDF077	<RSCAN0_base> + $14D8_H$
RSCAN0	Transmit buffer data field 1 register 77	RSCAN0TMDF177	<RSCAN0_base> + $14DC_H$
RSCAN0	Transmit buffer ID register 78	RSCAN0TMID78	<RSCAN0_base> + $14E0_H$
RSCAN0	Transmit buffer pointer register 78	RSCAN0TMPTR78	<RSCAN0_base> + $14E4_H$
RSCAN0	Transmit buffer data field 0 register 78	RSCAN0TMDF078	<RSCAN0_base> + $14E8_H$
RSCAN0	Transmit buffer data field 1 register 78	RSCAN0TMDF178	<RSCAN0_base> + $14EC_H$
RSCAN0	Transmit buffer ID register 79	RSCAN0TMID79	<RSCAN0_base> + $14F0_H$
RSCAN0	Transmit buffer pointer register 79	RSCAN0TMPTR79	<RSCAN0_base> + $14F4_H$
RSCAN0	Transmit buffer data field 0 register 79	RSCAN0TMDF079	<RSCAN0_base> + $14F8_H$
RSCAN0	Transmit buffer data field 1 register 79	RSCAN0TMDF179	<RSCAN0_base> + $14FC_H$
RSCAN0	Transmit buffer ID register 80	RSCAN0TMID80	<RSCAN0_base> + 1500_H
RSCAN0	Transmit buffer pointer register 80	RSCAN0TMPTR80	<RSCAN0_base> + 1504_H
RSCAN0	Transmit buffer data field 0 register 80	RSCAN0TMDF080	<RSCAN0_base> + 1508_H
RSCAN0	Transmit buffer data field 1 register 80	RSCAN0TMDF180	<RSCAN0_base> + $150C_H$
RSCAN0	Transmit buffer ID register 81	RSCAN0TMID81	<RSCAN0_base> + 1510_H
RSCAN0	Transmit buffer pointer register 81	RSCAN0TMPTR81	<RSCAN0_base> + 1514_H
RSCAN0	Transmit buffer data field 0 register 81	RSCAN0TMDF081	<RSCAN0_base> + 1518_H
RSCAN0	Transmit buffer data field 1 register 81	RSCAN0TMDF181	<RSCAN0_base> + $151C_H$
RSCAN0	Transmit buffer ID register 82	RSCAN0TMID82	<RSCAN0_base> + 1520_H
RSCAN0	Transmit buffer pointer register 82	RSCAN0TMPTR82	<RSCAN0_base> + 1524_H
RSCAN0	Transmit buffer data field 0 register 82	RSCAN0TMDF082	<RSCAN0_base> + 1528_H
RSCAN0	Transmit buffer data field 1 register 82	RSCAN0TMDF182	<RSCAN0_base> + $152C_H$
RSCAN0	Transmit buffer ID register 83	RSCAN0TMID83	<RSCAN0_base> + 1530_H
RSCAN0	Transmit buffer pointer register 83	RSCAN0TMPTR83	<RSCAN0_base> + 1534_H
RSCAN0	Transmit buffer data field 0 register 83	RSCAN0TMDF083	<RSCAN0_base> + 1538_H
RSCAN0	Transmit buffer data field 1 register 83	RSCAN0TMDF183	<RSCAN0_base> + $153C_H$
RSCAN0	Transmit buffer ID register 84	RSCAN0TMID84	<RSCAN0_base> + 1540_H
RSCAN0	Transmit buffer pointer register 84	RSCAN0TMPTR84	<RSCAN0_base> + 1544_H
RSCAN0	Transmit buffer data field 0 register 84	RSCAN0TMDF084	<RSCAN0_base> + 1548_H
RSCAN0	Transmit buffer data field 1 register 84	RSCAN0TMDF184	<RSCAN0_base> + $154C_H$
RSCAN0	Transmit buffer ID register 85	RSCAN0TMID85	<RSCAN0_base> + 1550_H
RSCAN0	Transmit buffer pointer register 85	RSCAN0TMPTR85	<RSCAN0_base> + 1554_H
RSCAN0	Transmit buffer data field 0 register 85	RSCAN0TMDF085	<RSCAN0_base> + 1558_H
RSCAN0	Transmit buffer data field 1 register 85	RSCAN0TMDF185	<RSCAN0_base> + $155C_H$
RSCAN0	Transmit buffer ID register 86	RSCAN0TMID86	<RSCAN0_base> + 1560_H
RSCAN0	Transmit buffer pointer register 86	RSCAN0TMPTR86	<RSCAN0_base> + 1564_H
RSCAN0	Transmit buffer data field 0 register 86	RSCAN0TMDF086	<RSCAN0_base> + 1568_H
RSCAN0	Transmit buffer data field 1 register 86	RSCAN0TMDF186	<RSCAN0_base> + $156C_H$
RSCAN0	Transmit buffer ID register 87	RSCAN0TMID87	<RSCAN0_base> + 1570_H
RSCAN0	Transmit buffer pointer register 87	RSCAN0TMPTR87	<RSCAN0_base> + 1574_H

Table 20.12 Registers (30/32)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 0 register 87	RSCAN0TMDF087	<RSCAN0_base> + 1578_H
RSCAN0	Transmit buffer data field 1 register 87	RSCAN0TMDF187	<RSCAN0_base> + $157C_H$
RSCAN0	Transmit buffer ID register 88	RSCAN0TMID88	<RSCAN0_base> + 1580_H
RSCAN0	Transmit buffer pointer register 88	RSCAN0TMPTR88	<RSCAN0_base> + 1584_H
RSCAN0	Transmit buffer data field 0 register 88	RSCAN0TMDF088	<RSCAN0_base> + 1588_H
RSCAN0	Transmit buffer data field 1 register 88	RSCAN0TMDF188	<RSCAN0_base> + $158C_H$
RSCAN0	Transmit buffer ID register 89	RSCAN0TMID89	<RSCAN0_base> + 1590_H
RSCAN0	Transmit buffer pointer register 89	RSCAN0TMPTR89	<RSCAN0_base> + 1594_H
RSCAN0	Transmit buffer data field 0 register 89	RSCAN0TMDF089	<RSCAN0_base> + 1598_H
RSCAN0	Transmit buffer data field 1 register 89	RSCAN0TMDF189	<RSCAN0_base> + $159C_H$
RSCAN0	Transmit buffer ID register 90	RSCAN0TMID90	<RSCAN0_base> + $15A0_H$
RSCAN0	Transmit buffer pointer register 90	RSCAN0TMPTR90	<RSCAN0_base> + $15A4_H$
RSCAN0	Transmit buffer data field 0 register 90	RSCAN0TMDF090	<RSCAN0_base> + $15A8_H$
RSCAN0	Transmit buffer data field 1 register 90	RSCAN0TMDF190	<RSCAN0_base> + $15AC_H$
RSCAN0	Transmit buffer ID register 91	RSCAN0TMID91	<RSCAN0_base> + $15B0_H$
RSCAN0	Transmit buffer pointer register 91	RSCAN0TMPTR91	<RSCAN0_base> + $15B4_H$
RSCAN0	Transmit buffer data field 0 register 91	RSCAN0TMDF091	<RSCAN0_base> + $15B8_H$
RSCAN0	Transmit buffer data field 1 register 91	RSCAN0TMDF191	<RSCAN0_base> + $15BC_H$
RSCAN0	Transmit buffer ID register 92	RSCAN0TMID92	<RSCAN0_base> + $15C0_H$
RSCAN0	Transmit buffer pointer register 92	RSCAN0TMPTR92	<RSCAN0_base> + $15C4_H$
RSCAN0	Transmit buffer data field 0 register 92	RSCAN0TMDF092	<RSCAN0_base> + $15C8_H$
RSCAN0	Transmit buffer data field 1 register 92	RSCAN0TMDF192	<RSCAN0_base> + $15CC_H$
RSCAN0	Transmit buffer ID register 93	RSCAN0TMID93	<RSCAN0_base> + $15D0_H$
RSCAN0	Transmit buffer pointer register 93	RSCAN0TMPTR93	<RSCAN0_base> + $15D4_H$
RSCAN0	Transmit buffer data field 0 register 93	RSCAN0TMDF093	<RSCAN0_base> + $15D8_H$
RSCAN0	Transmit buffer data field 1 register 93	RSCAN0TMDF193	<RSCAN0_base> + $15DC_H$
RSCAN0	Transmit buffer ID register 94	RSCAN0TMID94	<RSCAN0_base> + $15E0_H$
RSCAN0	Transmit buffer pointer register 94	RSCAN0TMPTR94	<RSCAN0_base> + $15E4_H$
RSCAN0	Transmit buffer data field 0 register 94	RSCAN0TMDF094	<RSCAN0_base> + $15E8_H$
RSCAN0	Transmit buffer data field 1 register 94	RSCAN0TMDF194	<RSCAN0_base> + $15EC_H$
RSCAN0	Transmit buffer ID register 95	RSCAN0TMID95	<RSCAN0_base> + $15F0_H$
RSCAN0	Transmit buffer pointer register 95	RSCAN0TMPTR95	<RSCAN0_base> + $15F4_H$
RSCAN0	Transmit buffer data field 0 register 95	RSCAN0TMDF095	<RSCAN0_base> + $15F8_H$
RSCAN0	Transmit buffer data field 1 register 95	RSCAN0TMDF195	<RSCAN0_base> + $15FC_H$
RSCAN0	Transmit history access register 0	RSCAN0THLACC0	<RSCAN0_base> + 1800_H
RSCAN0	Transmit history access register 1	RSCAN0THLACC1	<RSCAN0_base> + 1804_H
RSCAN0	Transmit history access register 2	RSCAN0THLACC2	<RSCAN0_base> + 1808_H
RSCAN0	Transmit history access register 3	RSCAN0THLACC3	<RSCAN0_base> + $180C_H$
RSCAN0	Transmit history access register 4	RSCAN0THLACC4	<RSCAN0_base> + 1810_H
RSCAN0	Transmit history access register 5	RSCAN0THLACC5	<RSCAN0_base> + 1814_H
RSCAN0	RAM test page access register 0	RSCAN0RPGACC0	<RSCAN0_base> + 1900_H
RSCAN0	RAM test page access register 1	RSCAN0RPGACC1	<RSCAN0_base> + 1904_H
RSCAN0	RAM test page access register 2	RSCAN0RPGACC2	<RSCAN0_base> + 1908_H
RSCAN0	RAM test page access register 3	RSCAN0RPGACC3	<RSCAN0_base> + $190C_H$

Table 20.12 Registers (31/32)

Module	Register	Symbol	Address
RSCAN0	RAM test page access register 4	RSCAN0RPGACC4	<RSCAN0_base> + 1910_H
RSCAN0	RAM test page access register 5	RSCAN0RPGACC5	<RSCAN0_base> + 1914_H
RSCAN0	RAM test page access register 6	RSCAN0RPGACC6	<RSCAN0_base> + 1918_H
RSCAN0	RAM test page access register 7	RSCAN0RPGACC7	<RSCAN0_base> + $191C_H$
RSCAN0	RAM test page access register 8	RSCAN0RPGACC8	<RSCAN0_base> + 1920_H
RSCAN0	RAM test page access register 9	RSCAN0RPGACC9	<RSCAN0_base> + 1924_H
RSCAN0	RAM test page access register 10	RSCAN0RPGACC10	<RSCAN0_base> + 1928_H
RSCAN0	RAM test page access register 11	RSCAN0RPGACC11	<RSCAN0_base> + $192C_H$
RSCAN0	RAM test page access register 12	RSCAN0RPGACC12	<RSCAN0_base> + 1930_H
RSCAN0	RAM test page access register 13	RSCAN0RPGACC13	<RSCAN0_base> + 1934_H
RSCAN0	RAM test page access register 14	RSCAN0RPGACC14	<RSCAN0_base> + 1938_H
RSCAN0	RAM test page access register 15	RSCAN0RPGACC15	<RSCAN0_base> + $193C_H$
RSCAN0	RAM test page access register 16	RSCAN0RPGACC16	<RSCAN0_base> + 1940_H
RSCAN0	RAM test page access register 17	RSCAN0RPGACC17	<RSCAN0_base> + 1944_H
RSCAN0	RAM test page access register 18	RSCAN0RPGACC18	<RSCAN0_base> + 1948_H
RSCAN0	RAM test page access register 19	RSCAN0RPGACC19	<RSCAN0_base> + $194C_H$
RSCAN0	RAM test page access register 20	RSCAN0RPGACC20	<RSCAN0_base> + 1950_H
RSCAN0	RAM test page access register 21	RSCAN0RPGACC21	<RSCAN0_base> + 1954_H
RSCAN0	RAM test page access register 22	RSCAN0RPGACC22	<RSCAN0_base> + 1958_H
RSCAN0	RAM test page access register 23	RSCAN0RPGACC23	<RSCAN0_base> + $195C_H$
RSCAN0	RAM test page access register 24	RSCAN0RPGACC24	<RSCAN0_base> + 1960_H
RSCAN0	RAM test page access register 25	RSCAN0RPGACC25	<RSCAN0_base> + 1964_H
RSCAN0	RAM test page access register 26	RSCAN0RPGACC26	<RSCAN0_base> + 1968_H
RSCAN0	RAM test page access register 27	RSCAN0RPGACC27	<RSCAN0_base> + $196C_H$
RSCAN0	RAM test page access register 28	RSCAN0RPGACC28	<RSCAN0_base> + 1970_H
RSCAN0	RAM test page access register 29	RSCAN0RPGACC29	<RSCAN0_base> + 1974_H
RSCAN0	RAM test page access register 30	RSCAN0RPGACC30	<RSCAN0_base> + 1978_H
RSCAN0	RAM test page access register 31	RSCAN0RPGACC31	<RSCAN0_base> + $197C_H$
RSCAN0	RAM test page access register 32	RSCAN0RPGACC32	<RSCAN0_base> + 1980_H
RSCAN0	RAM test page access register 33	RSCAN0RPGACC33	<RSCAN0_base> + 1984_H
RSCAN0	RAM test page access register 34	RSCAN0RPGACC34	<RSCAN0_base> + 1988_H
RSCAN0	RAM test page access register 35	RSCAN0RPGACC35	<RSCAN0_base> + $198C_H$
RSCAN0	RAM test page access register 36	RSCAN0RPGACC36	<RSCAN0_base> + 1990_H
RSCAN0	RAM test page access register 37	RSCAN0RPGACC37	<RSCAN0_base> + 1994_H
RSCAN0	RAM test page access register 38	RSCAN0RPGACC38	<RSCAN0_base> + 1998_H
RSCAN0	RAM test page access register 39	RSCAN0RPGACC39	<RSCAN0_base> + $199C_H$
RSCAN0	RAM test page access register 40	RSCAN0RPGACC40	<RSCAN0_base> + $19A0_H$
RSCAN0	RAM test page access register 41	RSCAN0RPGACC41	<RSCAN0_base> + $19A4_H$
RSCAN0	RAM test page access register 42	RSCAN0RPGACC42	<RSCAN0_base> + $19A8_H$
RSCAN0	RAM test page access register 43	RSCAN0RPGACC43	<RSCAN0_base> + $19AC_H$
RSCAN0	RAM test page access register 44	RSCAN0RPGACC44	<RSCAN0_base> + $19B0_H$
RSCAN0	RAM test page access register 45	RSCAN0RPGACC45	<RSCAN0_base> + $19B4_H$
RSCAN0	RAM test page access register 46	RSCAN0RPGACC46	<RSCAN0_base> + $19B8_H$
RSCAN0	RAM test page access register 47	RSCAN0RPGACC47	<RSCAN0_base> + $19BC_H$

Table 20.12 Registers (32/32)

Module	Register	Symbol	Address
RSCAN0	RAM test page access register 48	RSCAN0RPGACC48	<RSCAN0_base> + $19C0_H$
RSCAN0	RAM test page access register 49	RSCAN0RPGACC49	<RSCAN0_base> + $19C4_H$
RSCAN0	RAM test page access register 50	RSCAN0RPGACC50	<RSCAN0_base> + $19C8_H$
RSCAN0	RAM test page access register 51	RSCAN0RPGACC51	<RSCAN0_base> + $19CC_H$
RSCAN0	RAM test page access register 52	RSCAN0RPGACC52	<RSCAN0_base> + $19D0_H$
RSCAN0	RAM test page access register 53	RSCAN0RPGACC53	<RSCAN0_base> + $19D4_H$
RSCAN0	RAM test page access register 54	RSCAN0RPGACC54	<RSCAN0_base> + $19D8_H$
RSCAN0	RAM test page access register 55	RSCAN0RPGACC55	<RSCAN0_base> + $19DC_H$
RSCAN0	RAM test page access register 56	RSCAN0RPGACC56	<RSCAN0_base> + $19E0_H$
RSCAN0	RAM test page access register 57	RSCAN0RPGACC57	<RSCAN0_base> + $19E4_H$
RSCAN0	RAM test page access register 58	RSCAN0RPGACC58	<RSCAN0_base> + $19E8_H$
RSCAN0	RAM test page access register 59	RSCAN0RPGACC59	<RSCAN0_base> + $19EC_H$
RSCAN0	RAM test page access register 60	RSCAN0RPGACC60	<RSCAN0_base> + $19F0_H$
RSCAN0	RAM test page access register 61	RSCAN0RPGACC61	<RSCAN0_base> + $19F4_H$
RSCAN0	RAM test page access register 62	RSCAN0RPGACC62	<RSCAN0_base> + $19F8_H$
RSCAN0	RAM test page access register 63	RSCAN0RPGACC63	<RSCAN0_base> + $19FC_H$

Table 20.13 Transmit Buffer p Allocated to Each Channel

	CANm
Transmit buffer p	Transmit buffer 16 × m + 0
	Transmit buffer 16 × m + 1
	Transmit buffer 16 × m + 2
	Transmit buffer 16 × m + 3
	Transmit buffer 16 × m + 4
	Transmit buffer 16 × m + 5
	Transmit buffer 16 × m + 6
	Transmit buffer 16 × m + 7
	Transmit buffer 16 × m + 8
	Transmit buffer 16 × m + 9
	Transmit buffer 16 × m + 10
	Transmit buffer 16 × m + 11
	Transmit buffer 16 × m + 12
	Transmit buffer 16 × m + 13
	Transmit buffer 16 × m + 14
	Transmit buffer 16 × m + 15

Table 20.14 Transmit/Receive FIFO Buffer k Allocated to Each Channel

	CANm
Transmit/receive FIFO buffer k	Transmit/receive FIFO buffer 3 × m + 0
	Transmit/receive FIFO buffer 3 × m + 1
	Transmit/receive FIFO buffer 3 × m + 2

Table 20.15 Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[3:0]

Setting of Bits CFTML[3:0]	Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer
0000_B	Transmit buffer 16 × m + 0
0001_B	Transmit buffer 16 × m + 1
0010_B	Transmit buffer 16 × m + 2
0011_B	Transmit buffer 16 × m + 3
0100_B	Transmit buffer 16 × m + 4
0101_B	Transmit buffer 16 × m + 5
0110_B	Transmit buffer 16 × m + 6
0111_B	Transmit buffer 16 × m + 7
1000_B	Transmit buffer 16 × m + 8
1001_B	Transmit buffer 16 × m + 9
1010_B	Transmit buffer 16 × m + 10
1011_B	Transmit buffer 16 × m + 11
1100_B	Transmit buffer 16 × m + 12
1101_B	Transmit buffer 16 × m + 13
1110_B	Transmit buffer 16 × m + 14
1111_B	Transmit buffer 16 × m + 15

Table 20.16 Transmit Buffer p Allocated to the Transmit Queue of Each Channel

Setting of Bits TXQDC[3:0]	Transmit Buffer p Allocated to the Transmit Queue
0000_B	Setting prohibited
0001_B	Setting prohibited
0010_B	Transmit buffer 16 × m + 15 to 16 × m + 13
0011_B	Transmit buffer 16 × m + 15 to 16 × m + 12
0100_B	Transmit buffer 16 × m + 15 to 16 × m + 11
0101_B	Transmit buffer 16 × m + 15 to 16 × m + 10
0110_B	Transmit buffer 16 × m + 15 to 16 × m + 9
0111_B	Transmit buffer 16 × m + 15 to 16 × m + 8
1000_B	Transmit buffer 16 × m + 15 to 16 × m + 7
1001_B	Transmit buffer 16 × m + 15 to 16 × m + 6
1010_B	Transmit buffer 16 × m + 15 to 16 × m + 5
1011_B	Transmit buffer 16 × m + 15 to 16 × m + 4
1100_B	Transmit buffer 16 × m + 15 to 16 × m + 3
1101_B	Transmit buffer 16 × m + 15 to 16 × m + 2
1110_B	Transmit buffer 16 × m + 15 to 16 × m + 1
1111_B	Transmit buffer 16 × m + 15 to 16 × m + 0

20.3.2 RSCAN0CmCFG — Channel Configuration Register (m = 0 to 5)

Access: RSCAN0CmCFG register can be read or written in 32-bit units.
RSCAN0CmCFGL, RSCAN0CmCFGH registers can be read or written in 16-bit units.
RSCAN0CmCFGLL, RSCAN0CmCFGLH, RSCAN0CmCFGHL, RSCAN0CmCFGHH registers can be read or written in 8-bit units.

Address: RSCAN0CmCFG: <RSCAN0_base> + 0000_H + (10_H × m)

RSCAN0CmCFGL: <RSCAN0_base> + 0000_H + (10_H × m),
RSCAN0CmCFGH: <RSCAN0_base> + 0002_H + (10_H × m)

RSCAN0CmCFGLL: <RSCAN0_base> + 0000_H + (10_H × m),
RSCAN0CmCFGLH: <RSCAN0_base> + 0001_H + (10_H × m),
RSCAN0CmCFGHL: <RSCAN0_base> + 0002_H + (10_H × m),
RSCAN0CmCFGHH: <RSCAN0_base> + 0003_H + (10_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SJW[1:0]		—	TSEG2[2:0]			TSEG1[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	BRP[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.17 RSCAN0CmCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25, 24	SJW[1:0]	Resynchronization Jump Width Control b25 b24 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 20	TSEG2[2:0]	Time Segment 2 Control b22 b21 b20 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq

Table 20.17 RSCAN0CmCFG Register Contents (2/2)

Bit Position	Bit Name	Function
19 to 16	TSEG1[3:0]	Time Segment 1 Control b19 b18 b17 b16 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	BRP[9:0]	Prescaler Division Ratio Setting When these bits are set to P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

Modify the RSCAN0CmCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode before transitioning to channel communication mode or channel halt mode. For a description of the bit timing parameters and settings, see **Section 20.10.1, Initial Settings**.

SJW[1:0] Bits

These bits are used to specify a Tq value for the resynchronization jump width. Values from 1Tq to 4Tq can be specified.
Set a value less than or equal to the value of the TSEG2 bits.

TSEG2[2:0] Bits

These bits are used to specify a Tq value for the length of phase segment 2 (PHASE_SEG2).
Values from 2Tq to 8Tq can be specified.
Set a value smaller than the value of the TSEG1 bits.

TSEG1[3:0] Bits

These bits are used to specify a Tq value for the total length of the propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1).
Values from 4Tq to 16Tq can be specified..

BRP[9:0] Bits

The CANmTq clock (fCANTQm) is calculated by dividing the CAN clock (fCAN) by the baud rate prescaler, ((BRP[9:0]) + 1). One clock cycle of the CANmTq clock is 1 Time Quantum (Tq).

20.3.3 RSCAN0CmCTR — Channel Control Register (m = 0 to 5)

Access: RSCAN0CmCTR register can be read or written in 32-bit units.
RSCAN0CmCTRL, RSCAN0CmCTRH registers can be read or written in 16-bit units.
RSCAN0CmCTRLL, RSCAN0CmCTRLH, RSCAN0CmCTRHL, RSCAN0CmCTRHH registers can be read or written in 8-bit units.

Address: RSCAN0CmCTR: <RSCAN0_base> + 0004_H + (10_H × m)

RSCAN0CmCTRL: <RSCAN0_base> + 0004_H + (10_H × m),
RSCAN0CmCTRH: <RSCAN0_base> + 0006_H + (10_H × m)

RSCAN0CmCTRLL: <RSCAN0_base> + 0004_H + (10_H × m),
RSCAN0CmCTRLH: <RSCAN0_base> + 0005_H + (10_H × m),
RSCAN0CmCTRHL: <RSCAN0_base> + 0006_H + (10_H × m),
RSCAN0CmCTRHH: <RSCAN0_base> + 0007_H + (10_H × m)

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	—	—	—	TAIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	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
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 20.18 RSCAN0CmCTR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26, 25	CTMS[1:0]	Communication Test Mode Select b26 b25 0 0: Standard 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)
24	CTME	Communication Test Mode Enable 0: Communication test mode is disabled. 1: Communication test mode is enabled.
23	ERRD	Error Display Mode Select 0: Error flags are displayed only for the first error information after bits 14 to 8 in RSCAN0CmERFL are all cleared. 1: Error flags for all error information are displayed.
22, 21	BOM[1:0]	Bus Off Recovery Mode Select b22 b21 0 0: ISO11898-1 compliant 0 1: Entry to channel halt mode automatically at bus-off entry 1 0: Entry to channel halt mode automatically at bus-off end 1 1: Entry to channel halt mode (in bus-off state) by program request
20 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TAIE	Transmit Abort Interrupt Enable 0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled.
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.

Table 20.18 RSCAN0CmCTR Register Contents (2/2)

Bit Position	Bit Name	Function
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.
13	OLIE	Overload Frame Transmit Interrupt Enable 0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled.
12	BORIE	Bus Off Recovery Interrupt Enable 0: Bus off recovery interrupt is disabled. 1: Bus off recovery interrupt is enabled.
11	BOEIE	Bus Off Entry Interrupt Enable 0: Bus off entry interrupt is disabled. 1: Bus off entry interrupt is enabled.
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	RTBO	Forcible Return from Bus-off When this bit is set to 1, forcible return from the bus off state is made. This bit is always read as 0.
2	CSLPR	Channel Stop Mode 0: Other than channel stop mode 1: Channel stop mode
1, 0	CHMDC[1:0]	Mode Select b1 b0 0 0: Channel communication mode 0 1: Channel reset mode 1 0: Channel halt mode 1 1: Setting prohibited

CTMS[1:0] Bits

These bits are used to select a communication test mode. Modify these bits in channel halt mode only. These bits are set to 0 in channel reset mode.

CTME Bit

Setting this bit to 1 enables communication test mode. Modify this bit in channel halt mode. This bit is set to 0 in channel reset mode.

ERRD Bit

This bit is used to control the display mode of bits 14 to 8 in the RSCAN0CmERFL register.
When this bit is cleared to 0, only the flags of the first error event are set to 1. If two or more errors occur in the first error event, all the flags of the detected errors are set to 1.

When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order.

Modify this bit only in channel reset mode or channel halt mode.

BOM[1:0] Bits

These bits are used to select the bus off recovery mode of the RS-CAN module.

When the BOM[1:0] bits are set to 00_B, return from the bus off state to the error active state is compliant with the CAN specifications. That is, the RS-CAN module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CHMDC[1:0] bits are set to 10_B (channel halt mode) before recessive bits are detected 128 times, the RS-CAN module does not transition to channel halt mode until recessive bits are detected 128 times.

When the RS-CAN module reaches the bus off state when the BOM[1:0] bits are set to 01_B, the CHMDC[1:0] bits in the RSCAN0CmCTR register (m = 0 to 5) are set to 10_B and the RS-CAN module transitions to channel halt mode. No bus off recovery interrupt request is generated and the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register are cleared to 00_H.

When the RS-CAN module reaches the bus off state when the BOM[1:0] bits are set to 10_B, the CHMDC[1:0] bits are set to 10_B and the RS-CAN module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H.

When the BOM[1:0] bits are set to 11_B and the CHMDC[1:0] bits are set to 10_B while the RS-CAN module is in the bus off state, the RS-CAN module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H. However, if 11 consecutive recessive bits are detected 128 times and the RS-CAN module has recovered to the error active state from the bus off state before the CHMDC[1:0] bits are set to 10_B, a bus off recovery interrupt request is generated.

If a program writes to the CHMDC[1:0] bit at the same time as the RS-CAN module transition to channel halt mode (at bus off entry when the BOM[1:0] bits are 01B or at bus off end when the BOM[1:0] bits are 10_B), the program's writing takes precedence. Modify the BOM[1:0] bits only in channel reset mode.

TAIE Bit

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

ALIE Bit

When the ALF flag in the RSCAN0CmERFL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BLIE Bit

When the BLF flag in the RSCAN0CmERFL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

OLIE Bit

When the OVLF flag in the RSCAN0CmERFL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BORIE Bit

When the BORF flag in the RSCAN0CmERFL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BOEIE Bit

When the BOEF flag in the RSCAN0CmERFL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EPIE Bit

When the EPF flag in the RSCAN0CmERFL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EWIE Bit

When the EWF flag in the RSCAN0CmERFL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BEIE Bit

When the BEF flag in the RSCAN0CmERFL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

RTBO Bit

Setting this bit to 1 in the bus off state forcibly returns the state from the bus off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register to 00_H and also clears the BOSTS flag in the RSCAN0CmSTS register to 0 (not in bus off state). The other registers remain unchanged. No bus off recovery interrupt request is generated upon return from the bus off state in this case. Use this bit only when the BOM[1:0] bits in the RSCAN0CmCTR register are 00_B (ISO11898-1 compliant).

A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the RS-CAN module transitions to the error active state. Set this bit to 1 in channel communication mode.

CSLPR Bit

Setting this bit to 1 places the channel into channel stop mode.
Clearing this bit to 0 makes the channel exit channel stop mode.
This bit should not be modified in channel communication mode or channel halt mode.

CHMDC[1:0] Bits

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see **Section 20.5.2, Channel Modes**. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to 11_B. When the CAN module has automatically transitioned to channel halt mode based on the setting of the BOM[1:0] bits, the CHMDC[1:0] bits automatically become 10_B.

20.3.4 RSCAN0CmSTS — Channel Status Register (m = 0 to 5)

Access: RSCAN0CmSTS register is a read-only register that can be read in 32-bit units.
RSCAN0CmSTSL, RSCAN0CmSTSH registers are the read-only registers that can be read in 16-bit units.
RSCAN0CmSTSLL, RSCAN0CmSTSHL, RSCAN0CmSTSHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0CmSTS: <RSCAN0_base> + 0008_H + (10_H × m)

RSCAN0CmSTSL: <RSCAN0_base> + 0008_H + (10_H × m),
RSCAN0CmSTSH: <RSCAN0_base> + $000A_H$ + (10_H × m)

RSCAN0CmSTSLL: <RSCAN0_base> + 0008_H + (10_H × m),
RSCAN0CmSTSHL: <RSCAN0_base> + $000A_H$ + (10_H × m),
RSCAN0CmSTSHH: <RSCAN0_base> + $000B_H$ + (10_H × m)

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.19 RSCAN0CmSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	TEC[7:0]	The transmit error counter (TEC) can be read.
23 to 16	REC[7:0]	The receive error counter (REC) can be read.
15 to 8	Reserved	When read, the value after reset is returned.
7	COMSTS	Communication Status Flag 0: Communication is not ready. 1: Communication is ready.
6	RECSTS	Receive Status Flag 0: Bus idle, in transmission or bus off state 1: In reception
5	TRMSTS	Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state
4	BOSTS	Bus Off Status Flag 0: Not in bus off state 1: In bus off state
3	EPSTS	Error Passive Status Flag 0: Not in error passive state 1: In error passive state
2	CSLPSTS	Channel Stop Status Flag 0: Not in channel stop mode 1: In channel stop mode
1	CHLTSTS	Channel Halt Status Flag 0: Not in channel halt mode 1: In channel halt mode
0	CRSTSTS	Channel Reset Status Flag 0: Not in channel reset mode 1: In channel reset mode

TEC[7:0] Bits

These bits contain the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN specification (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

REC[7:0] Bits

These bits contain the receive error counter value. For receive error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

COMSTS Flag

This bit indicates that communication is ready.

This flag becomes 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

RECSTS Flag

This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

TRMSTS Flag

This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

BOSTS Flag

This flag is set to 1 when the bus off state (TEC[7:0] > 255) is entered. It is cleared to 0 when the CAN module has exited the bus off state.

EPSTS Flag

This flag is set to 1 when the RS-CAN module has entered the error passive state ((128 ≤ TEC[7:0] ≤ 255) or (128 ≤ REC[7:0])), It is cleared to 0 when the RS-CAN module has exited the error passive state or has entered channel reset mode.

CSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel stop mode, and is cleared to 0 when the CAN module has returned from channel stop mode.

CHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel halt mode, and is cleared to 0 when the CAN module has returned from channel halt mode.

CRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel reset mode, and is cleared to 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 when the CAN module transitions from channel reset mode to channel stop mode.

20.3.5 RSCAN0CmERFL — Channel Error Flag Register (m = 0 to 5)

Access: RSCAN0CmERFL register can be read or written in 32-bit units.
RSCAN0CmERFLL register can be read or written in 16-bit units.
RSCAN0CmERFLH register is a read-only register that can be read in 16-bit units.
RSCAN0CmERFLLL, RSCAN0CmERFLLH registers can be read or written in 8-bit units.
RSCAN0CmERFLHL, RSCAN0CmERFLHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0CmERFL: <RSCAN0_base> + $000C_H$ + (10_H × m)

RSCAN0CmERFLL: <RSCAN0_base> + $000C_H$ + (10_H × m),
RSCAN0CmERFLH: <RSCAN0_base> + $000E_H$ + (10_H × m)

RSCAN0CmERFLLL: <RSCAN0_base> + $000C_H$ + (10_H × m),
RSCAN0CmERFLLH: <RSCAN0_base> + $000D_H$ + (10_H × m),
RSCAN0CmERFLHL: <RSCAN0_base> + $000E_H$ + (10_H × m),
RSCAN0CmERFLHH: <RSCAN0_base> + $000F_H$ + (10_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCREG[14:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADERR	B0ERR	B1ERR	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
R/W	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.20 RSCAN0CmERFL Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 16	CRCREG[14:0]	CRC Calculation Data A CRC value calculated based on the transmit message or receive message is indicated.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	ADERR	ACK Delimiter Error Flag 0: No ACK delimiter error is detected. 1: ACK delimiter error is detected.
13	B0ERR	Dominant Bit Error Flag 0: No dominant bit error is detected. 1: Dominant bit error is detected.
12	B1ERR	Recessive Bit Error Flag 0: No recessive bit error is detected. 1: Recessive bit error is detected.
11	CERR	CRC Error Flag 0: No CRC error is detected. 1: CRC error is detected.
10	AERR	ACK Error Flag 0: No ACK error is detected. 1: ACK error is detected.
9	FERR	Form Error Flag 0: No form error is detected. 1: Form error is detected.

Table 20.20 RSCAN0CmERFL Register Contents (2/2)

Bit Position	Bit Name	Function
8	SERR	Stuff Error Flag 0: No stuff error is detected. 1: Stuff error is detected.
7	ALF	Arbitration-lost Flag 0: No arbitration-lost is detected. 1: Arbitration-lost is detected.
6	BLF	Bus Lock Flag 0: No channel bus is detected. 1: Channel bus is detected.
5	OVLF	Overload Flag 0: No overload is detected. 1: Overload is detected.
4	BORF	Bus Off Recovery Flag 0: No bus off recovery is detected. 1: Bus off recovery is detected.
3	BOEF	Bus Off Entry Flag 0: No bus off entry is detected. 1: Bus off entry is detected.
2	EPF	Error Passive Flag 0: No error passive is detected. 1: Error passive is detected.
1	EWF	Error Warning Flag 0: No error warning is detected. 1: Error warning is detected.
0	BEF	Bus Error Flag 0: No channel bus error is detected. 1: Channel bus error is detected.

See the CAN specification (ISO11898-1) for a description of error occurrence conditions. To clear each flag of this register, the program must write a 0 to the corresponding bit. These flags cannot be set to 1 by the program. If any of these flags is set to 1 at the same time that the program writes 0 to the flag, the flag is still set to 1. The channel reset mode transition clears all of these flags to 0.

If the ERRD bit in the RSCAN0CmCTR register is set to 0 (ie, only the flags for the first error event are displayed) and an error related to bits 14 to 8 of RSCAN0CmERFL is detected, the flag bits are only set by the error event if bits 14 to 8 were all 0 at the when time the error occurred.

CRCREG[14:0] Flag

When the CTME bit in the RSCAN0CmCTR register is set to 1 (communication test mode is enabled), the CRC value calculated based on the transmit or receive message can be read. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0.

ADERR Flag

This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

B0ERR Flag

This flag is set to 1 when a recessive bit has been detected though a dominant bit was transmitted.

B1ERR Flag

This flag is set to 1 when a dominant bit has been detected though a recessive bit was transmitted.

CERR Flag

This flag is set to 1 when a CRC error has been detected.

AERR Flag

This flag is set to 1 when an ACK error has been detected.

FERR Flag

This flag is set to 1 when a form error has been detected.

SERR Flag

This flag is set to 1 when a stuff error has been detected.

ALF Flag

This flag is set to 1 when an arbitration-lost has been detected.

BLF Flag

This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of a dominant lock is restarted when either of the following conditions is met.

- A recessive bit is detected after the BLF bit has been cleared from 1 to 0.
- The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been cleared from 1 to 0.

OVLF Flag

This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

BORF Flag

This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 01_B (channel reset mode).
- The RTBO bit in the RSCAN0CmCTR register is set to 1 (forcible return from the bus off state is made).
- The BOM[1:0] bits in the RSCAN0CmCTR register are set to 01_B (transition to channel halt mode at bus off entry).
- The CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 10_B (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to 11_B (transition to channel halt mode upon a request from the program during bus off).

BOEF Flag

This flag is set to 1 when the bus off state is reached (TEC[7:0] value > 255). This flag is also set to 1 if the bus off state is reached when the BOM[1:0] bits in the RSCAN0CmCTR register (m = 0 to 5) set to 01_B (transition to channel halt mode at bus off entry).

EPF Flag

This flag becomes 1 when the error passive state is reached (REC[7:0] or TEC[7:0] value > 127). This flag becomes 1 only when the REC[7:0] or TEC[7:0] value first exceeds 127. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then the REC[7:0] or TEC[7:0] value exceeds 127 again.

EWF Flag

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 95. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then the REC[7:0] or TEC[7:0] value exceeds 95 again.

BEF Flag

This flag is set to 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags in the RSCAN0CmERFL register is set to 1.

NOTE

To clear the flag of this register to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

20.3.6 RSCAN0GCFG — Global Configuration Register

Access: RSCAN0GCFG register can be read or written in 32-bit units.
RSCAN0GCFGL, RSCAN0GCFGH registers can be read or written in 16-bit units.
RSCAN0GCFGLL, RSCAN0GCFGLH, RSCAN0GCFGHL, RSCAN0GCFGHH registers can be read or written in 8-bit units.

Address: RSCAN0GCFG: <RSCAN0_base> + 0084_H

RSCAN0GCFGL: <RSCAN0_base> + 0084_H, RSCAN0GCFGH: <RSCAN0_base> + 0086_H

RSCAN0GCFGLL: <RSCAN0_base> + 0084_H, RSCAN0GCFGLH: <RSCAN0_base> + 0085_H,
RSCAN0GCFGHL: <RSCAN0_base> + 0086_H, RSCAN0GCFGHH: <RSCAN0_base> + 0087_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITRCP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSBTCS[2:0]			TSSS	TSP[3:0]				—	—	—	DCS	MME	DRE	DCE	TPRI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 20.21 RSCAN0GCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	ITRCP[15:0]	Interval Timer Prescaler Set When these bits are set to M, the pclk is divided by M. Setting 0000_H is prohibited when the interval timer is in use.
15 to 13	TSBTCS[2:0]	Timestamp Clock Source Select b15 b14 b13 0 0 0: Channel 0 bit time clock 0 0 1: Channel 1 bit time clock 0 1 0: Channel 2 bit time clock 0 1 1: Channel 3 bit time clock 1 0 0: Channel 4 bit time clock 1 0 1: Channel 5 bit time clock 1 1 0: Setting prohibited 1 1 1: Setting prohibited
12	TSSS	Timestamp Source Select 0: pclk/2*1 1: Bit time clock
11 to 8	TSP[3:0]	Timestamp Clock Source Division b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768

Table 20.21 RSCAN0GCFG Register Contents (2/2)

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	DCS	CAN Clock Source Select*2 0: clkc 1: clk_xincan
3	MME	Mirror Function Enable 0: Mirror function is disabled. 1: Mirror function is enabled.
2	DRE	DLC Replacement Enable 0: DLC replacement is disabled. 1: DLC replacement is enabled.
1	DCE	DLC Check Enable 0: DLC check is disabled. 1: DLC check is enabled.
0	TPRI	Transmit Priority Select 0: ID priority 1: Transmit buffer number priority

Note 1. When specifying pclk/2 as the timestamp counter count source, set bits TSBTCS[2:0] to 000_B.

Note 2. For the CAN clock frequency settings, see **Table 20.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1M**.

Modify the RSCAN0GCFG register only in global reset mode.

ITRCP[15:0] Bits

These bits are used to set a clock source division value of the interval timer for FIFO buffers. See **Section 20.7.3.1, Interval Transmission Function**.

TSBTCS[2:0] Bits

When the TSSS bit is 1, these bits are used to select the channel of the bit time clock that will be the clock source of the timestamp counter.

TSSS Bit

This bit is used to select a clock source of the timestamp counter.

TSP[3:0] Bits

A clock obtained by dividing the clock source selected with the TSBTCS[2:0] bits and TSSS bit according to the TSP[3:0] bits is used as the timestamp counter count source.

DCS Bit

When this bit is set to 0, clkc is used as the clock source of the CAN clock (fCAN).

When this bit is set to 1, clk_xincan is used as the clock source of the CAN clock (fCAN).

For the CAN clock frequency settings, see **Table 20.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1M**.

MME Bit

Setting this bit to 1 makes the mirror function available.

DRE Bit

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00_H is stored in each data byte beyond the DLC value of the receive rule.

The DLC replacement function is only available when the DCE bit is set to 1 (DLC check is enabled).

DCE Bit

Setting this bit to 1 makes the DLC check function available. When disabling the DLC check function, set the GAFLDLC[3:0] bits in the RSCAN0GAFLP0j register to 0000_B before clearing the DCE bit in the RSCAN0GCFG register to 0.

TPRI Bit

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the lowest transmit buffer number of those has the highest priority.

While the transmit queue is in use, this bit should be set to 0.

20.3.7 RSCAN0GCTR — Global Control Register

Access: RSCAN0GCTR register can be read or written in 32-bit units.
RSCAN0GCTRL, RSCAN0GCTRH registers can be read or written in 16-bit units.
RSCAN0GCTRLL, RSCAN0GCTRLH, RSCAN0GCTRHL registers can be read or written in 8-bit units.

Address: RSCAN0GCTR: <RSCAN0_base> + 0088_H

RSCAN0GCTRL: <RSCAN0_base> + 0088_H, RSCAN0GCTRH: <RSCAN0_base> + $008A_H$

RSCAN0GCTRLL: <RSCAN0_base> + 0088_H, RSCAN0GCTRLH: <RSCAN0_base> + 0089_H,
RSCAN0GCTRHL: <RSCAN0_base> + $008A_H$

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 20.22 RSCAN0GCTR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
16	TSRST	Timestamp Counter Reset Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0.
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	THLEIE	Transmit History Buffer Overflow Interrupt Enable 0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled.
9	MEIE	FIFO Message Lost Interrupt Enable 0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled.
8	DEIE	DLC Error Interrupt Enable 0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled.
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	GSLPR	Global Stop Mode 0: Other than global stop mode 1: Global stop mode
1, 0	GMDC[1:0]	Global Mode Select b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited

TSRST Bit

This bit is used to reset the timestamp counter. When this bit is set to 1, the RSCAN0GTSC register is cleared to 0000_H.

THLEIE Bit

When the THLEIE bit is set to 1 and the THLES flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

MEIE Bit

When the MEIE bit is set to 1 and the MES flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

DEIE Bit

When the DEIE bit is set to 1 and the DEF flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

GSLPR Bit

Setting this bit to 1 places the RS-CAN module into global stop mode.
Clearing this bit to 0 makes the RS-CAN module leave from global stop mode.
This bit should not be modified in global operating mode or global test mode.

GMDC[1:0] Bits

These bits are used to select the mode of entire RS-CAN module (global operating mode, global reset mode, or global test mode). For details, see **Section 20.5.1, Global Modes**. Setting the GSLPR bit to 1 when in global reset mode places the RS-CAN module into global stop mode.

20.3.8 RSCAN0GSTS — Global Status Register

Access: RSCAN0GSTS register is a read-only register that can be read in 32-bit units.
RSCAN0GSTSL register is a read-only register that can be read in 16-bit units.
RSCAN0GSTSLL register is a read-only register that can be read in 8-bit units.

Address: RSCAN0GSTS: <RSCAN0_base> + $008C_H$

RSCAN0GSTSL: <RSCAN0_base> + $008C_H$

RSCAN0GSTSLL: <RSCAN0_base> + $008C_H$

Value after reset: $0000\ 000D_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	GRAMINIT	GSLPSTS	GHLTSTS	GRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.23 RSCAN0GSTS Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	GRAMINIT	CAN RAM Initialization Status Flag 0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing.
2	GSLPSTS	Global Stop Status Flag 0: Not in global stop mode 1: In global stop mode
1	GHLTSTS	Global Test Status Flag 0: Not in global test mode 1: In global test mode
0	GRSTSTS	Global Reset Status Flag 0: Not in global reset mode 1: In global reset mode

GRAMINIT Flag

This flag indicates the initialization status of the CAN RAM.

This flag is set to 1 after the MCU has been reset, and is cleared to 0 when CAN RAM initialization is completed.

GSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to global stop mode, and is cleared to 0 when the CAN module has returned from global stop mode.

GHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global test mode, and is cleared to 0 when the CAN module has exited global test mode.

GRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global reset mode, and is cleared to 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

20.3.9 RSCAN0GERFL — Global Error Flag Register

Access: RSCAN0GERFL register can be read or written in 32-bit units.
RSCAN0GERFLL register can be read or written in 16-bit units.
RSCAN0GERFLLL register can be read or written in 8-bit units.

Address: RSCAN0GERFL: <RSCAN0_base> + 0090_H

RSCAN0GERFLL: <RSCAN0_base> + 0090_H

RSCAN0GERFLLL: <RSCAN0_base> + 0090_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	THLES	MES	DEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.24 RSCAN0GERFL Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	Reserved	When read, the undefined value is returned. When writing, write the value after reset.
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	Reserved	When read, the undefined value is returned. When writing, write the value after reset.
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	THLES	Transmit History Buffer Overflow Status Flag 0: No transmit history buffer overflow has occurred. 1: A transmit history buffer overflow has occurred.
1	MES	FIFO Message Lost Status Flag 0: No FIFO message lost error has occurred. 1: A FIFO message lost error has occurred.
0	DEF	DLC Error Flag 0: No DLC error has occurred. 1: A DLC error has occurred.

All flags in the RSCAN0GERFL register are cleared to 0 in global reset mode.

THLES Flag

The THLES flag is set to 1 when any one of the THLELT flags in the RSCAN0THLSTSm register (m = 0 to 5) is set to 1.

This flag is cleared to 0 when the THLELT flags of all channels are set to 0.

MES Flag

The MES flag is set to 1 when any one of the RFMLT flags in the RSCAN0RFSTSx register (x = 0 to 7) or the CFMLT flags in the RSCAN0CFSTSk register (k = 0 to 17) is set to 1.

This flag is cleared to 0 when all RFMLT flags and CFMLT flags are set to 0.

DEF Flag

The DEF flag is set to 1 when an error has been detected during the DLC check. The program can clear this flag by writing 0 to this bit.

NOTE

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

20.3.10 RSCAN0GTINTSTS0 — Global TX Interrupt Status Register 0

Access: RSCAN0GTINTSTS0 register is a read-only register that can be read in 32-bit units.
RSCAN0GTINTSTS0L, RSCAN0GTINTSTS0H registers are the read-only registers that can be read in 16-bit units.
RSCAN0GTINTSTS0LL, RSCAN0GTINTSTS0LH, RSCAN0GTINTSTS0HL, RSCAN0GTINTSTS0HH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0GTINTSTS0: <RSCAN0_base> + 0460_H

RSCAN0GTINTSTS0L: <RSCAN0_base> + 0460_H, RSCAN0GTINTSTS0H: <RSCAN0_base> + 0462_H

RSCAN0GTINTSTS0LL: <RSCAN0_base> + 0460_H, RSCAN0GTINTSTS0LH: <RSCAN0_base> + 0461_H,
RSCAN0GTINTSTS0HL: <RSCAN0_base> + 0462_H, RSCAN0GTINTSTS0HH: <RSCAN0_base> + 0463_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	THIF3	CFTIF3	TQIF3	TAIF3	TSIF3	—	—	—	THIF2	CFTIF2	TQIF2	TAIF2	TSIF2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 20.25 RSCAN0GTINTSTS0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned.
28	THIF3	Channel 3 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
27	CFTIF3	Channel 3 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
26	TQIF3	Channel 3 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
25	TAIF3	Channel 3 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested
24	TSIF3	Channel 3 Transmit Buffer Transmit Complete Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
23 to 21	Reserved	When read, the value after reset is returned.
20	THIF2	Channel 2 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
19	CFTIF2	Channel 2 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
18	TQIF2	Channel 2 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
17	TAIF2	Channel 2 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.

Table 20.25 RSCAN0GTINTSTS0 Register Contents (2/2)

Bit Position	Bit Name	Function
16	TSIF2	Channel 2 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
15 to 13	Reserved	When read, the value after reset is returned.
12	THIF1	Channel 1 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF1	Channel 1 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
10	TQIF1	Channel 1 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
9	TAIF1	Channel 1 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
8	TSIF1	Channel 1 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
7 to 5	Reserved	When read, the value after reset is returned.
4	THIF0	Channel 0 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF0	Channel 0 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF0	Channel 0 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF0	Channel 0 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
0	TSIF0	Channel 0 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RSCAN0TMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIE bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RSCAN0CmCTR register is 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQIE bit in the RSCAN0TXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCAN0TXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCAN0TXQSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the TXQIE bit is cleared to 0.

CFTIFm Bits

When the CFTXIE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RSCAN0CFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RSCAN0THLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCAN0THLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCAN0THLSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

20.3.11 RSCAN0GTINTSTS1 — Global TX Interrupt Status Register 1

Access: RSCAN0GTINTSTS1 register is a read-only register that can be read in 32-bit units.
RSCAN0GTINTSTS1L register is a read-only register that can be read in 16-bit units.
RSCAN0GTINTSTS1LL, RSCAN0GTINTSTS1LH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0GTINTSTS1: <RSCAN0_base> + 0464_H

RSCAN0GTINTSTS1L: <RSCAN0_base> + 0464_H

RSCAN0GTINTSTS1LL: <RSCAN0_base> + 0464_H, RSCAN0GTINTSTS1LH: <RSCAN0_base> + 0465_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF5	CFTIF5	TQIF5	TAIF5	TSIF5	—	—	—	THIF4	CFTIF4	TQIF4	TAIF4	TSIF4
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 20.26 RSCAN0GTINTSTS1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned.
12	THIF5	Channel 5 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF5	Channel 5 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
10	TQIF5	Channel 5 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
9	TAIF5	Channel 5 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
8	TSIF5	Channel 5 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
7 to 5	Reserved	When read, the value after reset is returned.
4	THIF4	Channel 4 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF4	Channel 4 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF4	Channel 4 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF4	Channel 4 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.

Table 20.26 RSCAN0GTINTSTS1 Register Contents (2/2)

Bit Position	Bit Name	Function
0	TSIF4	Channel 4 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RSCAN0TMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIE bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RSCAN0CmCTR register is set to 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQIE bit in the RSCAN0TXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCAN0TXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCAN0TXQSTSm register is cleared to 0, this bit is cleared to 0. Clearing the TXQIE bit to 0 also clears this flag to 0.

CFTIFm Bits

When the CFTXIE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RSCAN0CFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RSCAN0THLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCAN0THLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCAN0THLSTSm register is cleared to 0, this bit is cleared to 0.

This flag is also cleared to 0 when the THLIE bit is cleared to 0.

20.3.12 RSCAN0GTSC — Global Timestamp Counter Register

Access: RSCAN0GTSC register is a read-only register that can be read in 32-bit units.
RSCAN0GTSCL register is a read-only register that can be read in 16-bit units.

Address: RSCAN0GTSC: <RSCAN0_base> + 0094_H

RSCAN0GTSCL: <RSCAN0_base> + 0094_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.27 RSCAN0GTSC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 0	TS[15:0]	Timestamp Value The timestamp counter value can be read. Counter Value: 0000_H to $FFFF_H$

TS[15:0] Bits

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. When the SOF is detected, the TS[15:0] bit value is captured and later stored in the receive buffer or the FIFO buffer. The timestamp counter is initialized in global reset mode.

The timestamp counter starts and stops counting differently, depending on the count source.

- When the TSSS bit in the RSCAN0GCFG register is 0 (pclk):
  The timestamp counter starts counting when the RS-CAN module has transitioned to global operating mode.
  This counter stops counting when the RS-CAN module has transitioned to global stop mode or global test mode.
- When the TSSS bit is 1 (CANm bit time clock):
  The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.
  This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

20.3.13 RSCAN0GAFLECTR — Receive Rule Entry Control Register

Access: RSCAN0GAFLECTR register can be read or written in 32-bit units.
RSCAN0GAFLECTRL register can be read or written in 16-bit units.
RSCAN0GAFLECTRLL, RSCAN0GAFLECTRLH registers can be read or written in 8-bit units.

Address: RSCAN0GAFLECTR: <RSCAN0_base> + 0098_H

RSCAN0GAFLECTRL: <RSCAN0_base> + 0098_H

RSCAN0GAFLECTRLL: <RSCAN0_base> + 0098_H, RSCAN0GAFLECTRLH: <RSCAN0_base> + 0099_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	AFLDAE	—	—	—	AFLPN[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 20.28 RSCAN0GAFLECTR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	AFLDAE	Receive Rule Table Write Enable 0: Receive rule table write is disabled. 1: Receive rule table write is enabled.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	AFLPN[4:0]	Receive Rule Table Page Number Configuration A page number can be selected from a range of page 0 (00000_B) to page 23 (10111_B).

AFLDAE Bit

Setting this bit to 0 disables the write to the receive rule table. After writes to the receive rule table are completed, set this bit to 0 to disable the write to the table. The receive rule table can be read regardless of the value of this bit.

Set the AFLDAE bit to 1 only in global reset mode.

AFLPN[4:0] Bits

These bits are used to set the page number of the receive rule table. Sixteen receive rules can be set per page.

Set these bits to a value within the range of 00000_B to 10111_B.

20.3.14 RSCAN0GAFLCFG0 — Receive Rule Configuration Register 0

Access: RSCAN0GAFLCFG0 register can be read or written in 32-bit units.
RSCAN0GAFLCFG0L, RSCAN0GAFLCFG0H registers can be read or written in 16-bit units.
RSCAN0GAFLCFG0LL, RSCAN0GAFLCFG0LH, RSCAN0GAFLCFG0HL, RSCAN0GAFLCFG0HH registers can be read or written in 8-bit units.

Address: RSCAN0GAFLCFG0: <RSCAN0_base> + $009C_H$

RSCAN0GAFLCFG0L: <RSCAN0_base> + $009C_H$, RSCAN0GAFLCFG0H: <RSCAN0_base> + $009E_H$

RSCAN0GAFLCFG0LL: <RSCAN0_base> + $009C_H$, RSCAN0GAFLCFG0LH: <RSCAN0_base> + $009D_H$,
RSCAN0GAFLCFG0HL: <RSCAN0_base> + $009E_H$, RSCAN0GAFLCFG0HH: <RSCAN0_base> + $009F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC0[7:0]								RNC1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RNC2[7:0]								RNC3[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.29 RSCAN0GAFLCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC0[7:0]	Number of Rules for Channel 0 Set the number of receive rules exclusively used for channel 0.
23 to 16	RNC1[7:0]	Number of Rules for Channel 1 Set the number of receive rules exclusively used for channel 1.
15 to 8	RNC2[7:0]	Number of Rules for Channel 2 Set the number of receive rules exclusively used for channel 2.
7 to 0	RNC3[7:0]	Number of Rules for Channel 3 Set the number of receive rules exclusively used for channel 3.

Modify the RSCAN0GAFLCFG0 register only in global reset mode.

Up to 64 × (number of channels) rules can be registered in the receive rule table as the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total of the number of rules allocated to each channel is not larger than the number of rules that can be registered in the entire unit.

RNC0[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 0 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC1[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 1 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC2[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 2 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC3[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 3 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

20.3.15 RSCAN0GAFLCFG1 — Receive Rule Configuration Register 1

Access: RSCAN0GAFLCFG1 register can be read or written in 32-bit units.
RSCAN0GAFLCFG1H register can be read or written in 16-bit units.
RSCAN0GAFLCFG1HL, RSCAN0GAFLCFG1HH registers can be read or written in 8-bit units.

Address: RSCAN0GAFLCFG1: <RSCAN0_base> + $00A0_H$

RSCAN0GAFLCFG1H: <RSCAN0_base> + $00A2_H$

RSCAN0GAFLCFG1HL: <RSCAN0_base> + $00A2_H$, RSCAN0GAFLCFG1HH: <RSCAN0_base> + $00A3_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC4[7:0]								RNC5[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.30 RSCAN0GAFLCFG1 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC4[7:0]	Number of Rules for Channel 4 Set the number of receive rules exclusively used for channel 4.
23 to 16	RNC5[7:0]	Number of Rules for Channel 5 Set the number of receive rules exclusively used for channel 5.
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Modify the RSCAN0GAFLCFG1 register only in global reset mode.

Up to 64 × (number of channels) rules can be registered in the receive rule table as the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total of the number of rules allocated to each channel is not larger than the number of rules that can be registered in the entire unit.

RNC4[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 4 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC5[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 5 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

20.3.16 RSCAN0GAFLIDj — Receive Rule ID Register (j = 0 to 15)

Access: RSCAN0GAFLIDj register can be read or written in 32-bit units.
RSCAN0GAFLIDjL, RSCAN0GAFLIDjH registers can be read or written in 16-bit units.
RSCAN0GAFLIDjLL, RSCAN0GAFLIDjLH, RSCAN0GAFLIDjHL, RSCAN0GAFLIDjHH registers can be read or written in 8-bit units.

Address: RSCAN0GAFLIDj: <RSCAN0_base> + 0500_H + (10_H × j)

RSCAN0GAFLIDjL: <RSCAN0_base> + 0500_H + (10_H × j),
RSCAN0GAFLIDjH: <RSCAN0_base> + 0502_H + (10_H × j)

RSCAN0GAFLIDjLL: <RSCAN0_base> + 0500_H + (10_H × j),
RSCAN0GAFLIDjLH: <RSCAN0_base> + 0501_H + (10_H × j),
RSCAN0GAFLIDjHL: <RSCAN0_base> + 0502_H + (10_H × j),
RSCAN0GAFLIDjHH: <RSCAN0_base> + 0503_H + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.31 RSCAN0GAFLIDj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDE	IDE Select 0: Standard ID 1: Extended ID
30	GAFLRTR	RTR Select 0: Data frame 1: Remote frame
29	GAFLLB	Receive Rule Target Message Select 0: When a message transmitted from another CAN node is received 1: When the own transmitted message is received
28 to 0	GAFLID[28:0]	ID Set the ID of the receive rule. For the standard ID, set the ID in bits b10 to b0 and set bits b28 to b11 to 0.

Modify the RSCAN0GAFLIDj register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDE Bit

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

GAFLRTR Bit

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

GAFLLB Bit

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when the CAN node is receiving its own transmitted messages.

GAFLID[28:0] Bits

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID of the received message during the acceptance filter processing.

20.3.17 RSCAN0GAFLMj — Receive Rule Mask Register (j = 0 to 15)

Access: RSCAN0GAFLMj register can be read or written in 32-bit units.
RSCAN0GAFLMjL, RSCAN0GAFLMjH registers can be read or written in 16-bit units.
RSCAN0GAFLMjLL, RSCAN0GAFLMjLH, RSCAN0GAFLMjHL, RSCAN0GAFLMjHH registers can be read or written in 8-bit units.

Address: RSCAN0GAFLMj: <RSCAN0_base> + 0504_H + (10_H × j)

RSCAN0GAFLMjL: <RSCAN0_base> + 0504_H + (10_H × j),
RSCAN0GAFLMjH: <RSCAN0_base> + 0506_H + (10_H × j)

RSCAN0GAFLMjLL: <RSCAN0_base> + 0504_H + (10_H × j),
RSCAN0GAFLMjLH: <RSCAN0_base> + 0505_H + (10_H × j),
RSCAN0GAFLMjHL: <RSCAN0_base> + 0506_H + (10_H × j),
RSCAN0GAFLMjHH: <RSCAN0_base> + 0507_H + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDEM	GAFLRTRM	—	GAFLIDM[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLIDM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.32 RSCAN0GAFLMj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDEM	IDE Mask 0: The IDE bit is not compared. 1: The IDE bit is compared.
30	GAFLRTRM	RTR Mask 0: The RTR bit is not compared. 1: The RTR bit is compared
29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 0	GAFLIDM[28:0]	ID Mask 0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.

Modify the RSCAN0GAFLMj register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDEM Bit

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the RSCAN0GAFLIDj register.

When this bit is cleared to 0, the IDs of all the receive messages and the specified IDs are regarded as matched. To set the GAFLIDEM bit to 0, set the GAFLIDM[28:0] bits to all 0 at the same time.

GAFLRTRM Bit

This bit is used to mask the RTR bit of the receive rule.

GAFLIDM[28:0] Bits

These bits are used to mask the corresponding ID bit of the receive rule.

20.3.18 RSCAN0GAFLP0j — Receive Rule Pointer 0 Register (j = 0 to 15)

Access: RSCAN0GAFLP0j register can be read or written in 32-bit units.
RSCAN0GAFLP0jL, RSCAN0GAFLP0jH registers can be read or written in 16-bit units.
RSCAN0GAFLP0jLH, RSCAN0GAFLP0jHL, RSCAN0GAFLP0jHH registers can be read or written in 8-bit units.

Address: RSCAN0GAFLP0j: <RSCAN0_base> + 0508_H + (10_H × j)

RSCAN0GAFLP0jL: <RSCAN0_base> + 0508_H + (10_H × j),
RSCAN0GAFLP0jH: <RSCAN0_base> + $050A_H$ + (10_H × j)

RSCAN0GAFLP0jLH: <RSCAN0_base> + 0509_H + (10_H × j),
RSCAN0GAFLP0jHL: <RSCAN0_base> + $050A_H$ + (10_H × j),
RSCAN0GAFLP0jHH: <RSCAN0_base> + $050B_H$ + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLDLC[3:0]				GAFLPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLRMV	GAFLRMDP[6:0]							—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 20.33 RSCAN0GAFLP0j Register Contents

Bit Position	Bit Name	Function
31 to 28	GAFLDLC[3:0]	Receive Rule DLC b31 b30 b29 b28 0 0 0 0: DLC check is disabled. 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	GAFLPTR[11:0]	Receive Rule Label Set the 12-bit label information.
15	GAFLRMV	Receive Buffer Enable 0: No receive buffer is used. 1: A receive buffer is used.
14 to 8	GAFLRMDP[6:0]	Receive Buffer Number Select Set the receive buffer number to store receive messages.
7 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Modify the RSCAN0GAFLP0j register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLDLC[3:0] Bits

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000_B disables the DLC check function allowing messages with any data length to pass the DLC check.

GAFLPTR[11:0] Bits

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

GAFLRMV Bit

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

GAFLRMDP[6:0] Bits

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the NRXMB[7:0] bits in the RSCAN0RMNB register.

20.3.19 RSCAN0GAFLP1j — Receive Rule Pointer 1 Register (j = 0 to 15)

Access: RSCAN0GAFLP1j register can be read or written in 32-bit units.
RSCAN0GAFLP1jL, RSCAN0GAFLP1jH registers can be read or written in 16-bit units.
RSCAN0GAFLP1jLL, RSCAN0GAFLP1jLH, RSCAN0GAFLP1jHL, RSCAN0GAFLP1jHH registers can be read or written in 8-bit units.

Address: RSCAN0GAFLP1j: <RSCAN0_base> + $050C_H$ + (10_H × j)

RSCAN0GAFLP1jL: <RSCAN0_base> + $050C_H$ + (10_H × j),
RSCAN0GAFLP1jH: <RSCAN0_base> + $050E_H$ + (10_H × j)

RSCAN0GAFLP1jLL: <RSCAN0_base> + $050C_H$ + (10_H × j),
RSCAN0GAFLP1jLH: <RSCAN0_base> + $050D_H$ + (10_H × j),
RSCAN0GAFLP1jHL: <RSCAN0_base> + $050E_H$ + (10_H × j),
RSCAN0GAFLP1jHH: <RSCAN0_base> + $050F_H$ + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GAFLFDP[25:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLFDP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.34 RSCAN0GAFLP1j Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 8	GAFLFDP[25:8]	Transmit/Receive FIFO Buffer k Select (Bit position –8 = target transmit/receive FIFO buffer number k) 0: Transmit/receive FIFO buffer is not selected. 1: Transmit/receive FIFO buffer is selected.
7 to 0	GAFLFDP[7:0]	Receive FIFO Buffer x Select (Bit position = target receive FIFO buffer number x) 0: Receive FIFO buffer is not selected. 1: Receive FIFO buffer is selected.

Modify the RSCAN0GAFLP1j register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLFDP [25:0] Bits

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to eight FIFO buffers are selectable. However, when the GAFLRMV bit in the RSCAN0GAFLP0j register is set to 1 (a message is stored in the receive buffer), up to seven FIFO buffers can be selected. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFM[1:0] bits in the RSCAN0CFCCk register are set to 00_B (receive mode) or 10_B (gateway mode) are selectable.

20.3.20 RSCAN0RMNB — Receive Buffer Number Register

Access: RSCAN0RMNB register can be read or written in 32-bit units.
RSCAN0RMNBL register can be read or written in 16-bit units.
RSCAN0RMNBLL register can be read or written in 8-bit units.

Address: RSCAN0RMNB: <RSCAN0_base> + $00A4_H$

RSCAN0RMNBL: <RSCAN0_base> + $00A4_H$

RSCAN0RMNBLL: <RSCAN0_base> + $00A4_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	NRXMB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.35 RSCAN0RMNB Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	NRXMB[7:0]	Receive Buffer Number Configuration Set the number of receive buffers. Set a value of 0 to 96.

Modify the RSCAN0RMNB register only in global reset mode.

NRXMB[7:0] Bits

These bits are used to set the total number of receive buffers of the RS-CAN module. The maximum value is 16 × (number of channels).

Setting these bits all to 0 makes receive buffers unavailable.

20.3.21 RSCAN0RMNDy — Receive Buffer New Data Register (y = 0 to 2)

Access: RSCAN0RMNDy register can be read or written in 32-bit units.
RSCAN0RMNDyL, RSCAN0RMNDyH registers can be read or written in 16-bit units.
RSCAN0RMNDyLL, RSCAN0RMNDyLH, RSCAN0RMNDyHL, RSCAN0RMNDyHH registers can be read or written in 8-bit units.

Address: RSCAN0RMNDy: <RSCAN0_base> + $00A8_H$ + (04_H × y)

RSCAN0RMNDyL: <RSCAN0_base> + $00A8_H$ + (04_H × y),
RSCAN0RMNDyH: <RSCAN0_base> + $00AA_H$ + (04_H × y)

RSCAN0RMNDyLL: <RSCAN0_base> + $00A8_H$ + (04_H × y),
RSCAN0RMNDyLH: <RSCAN0_base> + $00A9_H$ + (04_H × y),
RSCAN0RMNDyHL: <RSCAN0_base> + $00AA_H$ + (04_H × y),
RSCAN0RMNDyHH: <RSCAN0_base> + $00AB_H$ + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMNSq (q = y × 32 + 31 to y × 32 + 16 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMNSq (q = y × 32 + 15 to y × 32 + 0 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.36 RSCAN0RMNDy Register Contents

Bit Position	Bit Name	Function
31 to 16	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 31 to y × 32 + 16) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.
15 to 0	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 15 to y × 32 + 0) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.

Write 0 to the RSCAN0RMNDy register in global operating mode or global test mode.

RMNSq Flags (q = 0 to 95)

Each RMNS flag is set to 1 when the processing for storing a message in the corresponding receive buffer starts.

To clear a flag to 0, the program must write 0 to the flag. Use a store instruction to write “0” to the flag and “1” to other flags. These bits cannot be set to 0 while a message is being stored. It takes ten clock cycles of pclk to store a message.

These flags are cleared to 0 in global reset mode.

20.3.22 RSCAN0RMIDq — Receive Buffer ID Register (q = 0 to 95)

Access: RSCAN0RMIDq register is a read-only register that can be read in 32-bit units.
RSCAN0RMIDqL, RSCAN0RMIDqH registers are the read-only registers that can be read in 16-bit units.
RSCAN0RMIDqLL, RSCAN0RMIDqLH, RSCAN0RMIDqHL, RSCAN0RMIDqHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0RMIDq: <RSCAN0_base> + 0600_H + (10_H × q)

RSCAN0RMIDqL: <RSCAN0_base> + 0600_H + (10_H × q),
RSCAN0RMIDqH: <RSCAN0_base> + 0602_H + (10_H × q)

RSCAN0RMIDqLL: <RSCAN0_base> + 0600_H + (10_H × q),
RSCAN0RMIDqLH: <RSCAN0_base> + 0601_H + (10_H × q),
RSCAN0RMIDqHL: <RSCAN0_base> + 0602_H + (10_H × q),
RSCAN0RMIDqHH: <RSCAN0_base> + 0603_H + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMIDE	RMRTR	—	RMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.37 RSCAN0RMIDq Register Contents

Bit Position	Bit Name	Function
31	RMIDE	Receive Buffer IDE 0: Standard ID 1: Extended ID
30	RMRTR	Receive Buffer RTR 0: Data frame 1: Remote frame
29	Reserved	When read, the value after reset is returned.
28 to 0	RMID[28:0]	Receive Buffer ID Data These bits contain the standard ID or extended ID of the received message. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RMIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

RMRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer.

RMID[28:0] Bits

These bits contain the ID of the message stored in the receive buffer.

20.3.23 RSCAN0RMPTRq — Receive Buffer Pointer Register (q = 0 to 95)

Access: RSCAN0RMPTRq register is a read-only register that can be read in 32-bit units.
RSCAN0RMPTRqL, RSCAN0RMPTRqH registers are the read-only registers that can be read in 16-bit units.
RSCAN0RMPTRqLL, RSCAN0RMPTRqLH, RSCAN0RMPTRqHL, RSCAN0RMPTRqHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0RMPTRq: <RSCAN0_base> + 0604_H + (10_H × q)

RSCAN0RMPTRqL: <RSCAN0_base> + 0604_H + (10_H × q),
RSCAN0RMPTRqH: <RSCAN0_base> + 0606_H + (10_H × q)

RSCAN0RMPTRqLL: <RSCAN0_base> + 0604_H + (10_H × q),
RSCAN0RMPTRqLH: <RSCAN0_base> + 0605_H + (10_H × q),
RSCAN0RMPTRqHL: <RSCAN0_base> + 0606_H + (10_H × q),
RSCAN0RMPTRqHH: <RSCAN0_base> + 0607_H + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDLC[3:0]				RMPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.38 RSCAN0RMPTRq Register Contents

Bit Position	Bit Name	Function
31 to 28	RMDLC[3:0]	Receive Buffer DLC Data b31 b30 b29 b28 0 0 0 0: No data byte 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RMPTR[11:0]	Receive Buffer Label Data Label information of the received message.
15 to 0	RMTS[15:0]	Receive Buffer Timestamp Data Timestamp value of the received message.

RMDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive buffer.

RMPTR[11:0] Bits

These bits indicate the label information of the message stored in the receive buffer.

RMTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive buffer.

20.3.24 RSCAN0RMDF0q — Receive Buffer Data Field 0 Register (q = 0 to 95)

Access: RSCAN0RMDF0q register is a read-only register that can be read in 32-bit units.
RSCAN0RMDF0qL, RSCAN0RMDF0qH registers are the read-only registers that can be read in 16-bit units.
RSCAN0RMDF0qLL, RSCAN0RMDF0qLH, RSCAN0RMDF0qHL, RSCAN0RMDF0qHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0RMDF0q: <RSCAN0_base> + 0608_H + (10_H × q)

RSCAN0RMDF0qL: <RSCAN0_base> + 0608_H + (10_H × q),
RSCAN0RMDF0qH: <RSCAN0_base> + $060A_H$ + (10_H × q)

RSCAN0RMDF0qLL: <RSCAN0_base> + 0608_H + (10_H × q),
RSCAN0RMDF0qLH: <RSCAN0_base> + 0609_H + (10_H × q),
RSCAN0RMDF0qHL: <RSCAN0_base> + $060A_H$ + (10_H × q),
RSCAN0RMDF0qHH: <RSCAN0_base> + $060B_H$ + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB3[7:0]								RMDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB1[7:0]								RMDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.39 RSCAN0RMDF0q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB3[7:0]	Receive Buffer Data Byte 3
23 to 16	RMDB2[7:0]	Receive Buffer Data Byte 2
15 to 8	RMDB1[7:0]	Receive Buffer Data Byte 1
7 to 0	RMDB0[7:0]	Receive Buffer Data Byte 0 Data for a message stored in the receive buffer can be read.

When the RMDLC[3:0] bit value in the RSCAN0RMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

20.3.25 RSCAN0RMDF1q — Receive Buffer Data Field 1 Register (q = 0 to 95)

Access: RSCAN0RMDF1q register is a read-only register that can be read in 32-bit units.
RSCAN0RMDF1qL, RSCAN0RMDF1qH registers are the read-only registers that can be read in 16-bit units.
RSCAN0RMDF1qLL, RSCAN0RMDF1qLH, RSCAN0RMDF1qHL, RSCAN0RMDF1qHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0RMDF1q: <RSCAN0_base> + $060C_H$ + (10_H × q)

RSCAN0RMDF1qL: <RSCAN0_base> + $060C_H$ + (10_H × q),
RSCAN0RMDF1qH: <RSCAN0_base> + $060E_H$ + (10_H × q)

RSCAN0RMDF1qLL: <RSCAN0_base> + $060C_H$ + (10_H × q),
RSCAN0RMDF1qLH: <RSCAN0_base> + $060D_H$ + (10_H × q),
RSCAN0RMDF1qHL: <RSCAN0_base> + $060E_H$ + (10_H × q),
RSCAN0RMDF1qHH: <RSCAN0_base> + $060F_H$ + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB7[7:0]								RMDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB5[7:0]								RMDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.40 RSCAN0RMDF1q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB7[7:0]	Receive Buffer Data Byte 7
23 to 16	RMDB6[7:0]	Receive Buffer Data Byte 6
15 to 8	RMDB5[7:0]	Receive Buffer Data Byte 5
7 to 0	RMDB4[7:0]	Receive Buffer Data Byte 4 Data for a message stored in the receive buffer can be read.

When the RMDLC[3:0] bit value in the RSCAN0RMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

20.3.26 RSCAN0RFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7)

Access: RSCAN0RFCCx register can be read or written in 32-bit units.
RSCAN0RFCCxL register can be read or written in 16-bit units.
RSCAN0RFCCxLL, RSCAN0RFCCxLH registers can be read or written in 8-bit units.

Address: RSCAN0RFCCx: <RSCAN0_base> + $00B8_H$ + (04_H × x)

RSCAN0RFCCxL: <RSCAN0_base> + $00B8_H$ + (04_H × x)

RSCAN0RFCCxLL: <RSCAN0_base> + $00B8_H$ + (04_H × x),
RSCAN0RFCCxLH: <RSCAN0_base> + $00B9_H$ + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	—	—	—	—	—	RFIE	RFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 20.41 RSCAN0RFCCx Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 13	RFIGCV[2:0]	Receive FIFO Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.
12	RFIM	Receive FIFO Interrupt Source Select 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received.
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	RFDC[2:0]	Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	RFIE	Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled.

Table 20.41 RSCAN0RFCCx Register Contents (2/2)

Bit Position	Bit Name	Function
0	RFE	Receive FIFO Buffer Enable 0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used.

RFIGCV[2:0] Bits

These bits are used to specify the number of received messages for generating a receive FIFO interrupt request when the RFIM bit is set to 0 with a fraction for the total number of buffers (the setting of RFDC[2:0]).

When the RFDC[2:0] bits are set to 001_B (4 messages), set the RFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B. Modify these bits only in global reset mode.

RFIM Bit

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

RFDC[2:0] Bits

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. When these bits are set to 000_B, no receive FIFO buffer should be used. Modify these bits only in global reset mode.

RFIE Bit

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit set to 0 (no receive FIFO buffer is used).

RFE Bit

Setting the RFE bit to 1 makes receive FIFO buffers available. Clearing this bit to 0 sets the RFEMP flag in the RSCAN0RFSTSx register to 1 (buffer empty). Modify this bit in global operating mode or global test mode.

After all other bits in the RSCANnRFCCx register have been set, set this bit to 1 by using other instructions.

20.3.27 RSCAN0RFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7)

Access: RSCAN0RFSTSx register can be read or written in 32-bit units.
RSCAN0RFSTSxL register can be read or written in 16-bit units.
RSCAN0RFSTSxLL register can be read or written in 8-bit units.
RSCAN0RFSTSxLH register is a read-only register that can be read in 8-bit units.

Address: RSCAN0RFSTSx: <RSCAN0_base> + $00D8_H$ + (04_H × x)

RSCAN0RFSTSxL: <RSCAN0_base> + $00D8_H$ + (04_H × x)

RSCAN0RFSTSxLL: <RSCAN0_base> + $00D8_H$ + (04_H × x),
RSCAN0RFSTSxLH: <RSCAN0_base> + $00D9_H$ + (04_H × x)

Value after reset: $0000\ 0001_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMC[7:0]								—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.42 RSCAN0RFSTSx Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	RFMC[7:0]	Receive FIFO Unread Message Counter The number of unread messages stored in the receive FIFO buffer is displayed.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	RFIF	Receive FIFO Interrupt Request Flag 0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present.
2	RFMLT	Receive FIFO Message Lost Flag 0: No receive FIFO message is lost. 1: A receive FIFO message is lost.
1	RFFLL	Receive FIFO Buffer Full Status Flag 0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full.
0	RFEMP	Receive FIFO Buffer Empty Status Flag 0: The receive FIFO buffer contains unread message. 1: The receive FIFO buffer contains no unread message (buffer empty).

RFMC[7:0] Flag

These flags indicate the number of unread messages in the receive FIFO buffer. These flags become 00_H when the RFE bit in the RSCAN0RFCCx register is set to 0.

RFIF Flag

This flag is set to 1 when the receive FIFO interrupt request generation conditions set by the RFIGCV[2:0] bits and the RFIM bit in the RSCAN0RFCCx register are met. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.
To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFMLT Flag

This flag is set to 1 when an attempt is made to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag is cleared to 0 in global reset mode or by writing 0 to this flag.

Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFFLL Flag

This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RSCAN0RFCCx register.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RSCAN0RFCCx register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

RFEMP Flag

This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RSCAN0RFCCx register is 0 or in global reset mode.

This flag is cleared to 0 when even a single received message has been stored in the receive FIFO buffer.

NOTE

To clear the RFMLT or RFIF flag to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

20.3.28 RSCAN0RFPCTRx — Receive FIFO Buffer Pointer Control Register (x = 0 to 7)

Access: RSCAN0RFPCTRx register is a write-only register that can be written in 32-bit units.
RSCAN0RFPCTRxL register is a write-only register that can be written in 16-bit units.
RSCAN0RFPCTRxLL register is a write-only register that can be written in 8-bit units.

Address: RSCAN0RFPCTRx: <RSCAN0_base> + $00F8_H$ + (04_H × x)

RSCAN0RFPCTRxL: <RSCAN0_base> + $00F8_H$ + (04_H × x)

RSCAN0RFPCTRxLL: <RSCAN0_base> + $00F8_H$ + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.43 RSCAN0RFPCTRx Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	RFPC[7:0]	Receive FIFO Pointer Control When these bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer.

RFPC[7:0] Bits

When the RFPC[7:0] bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[7:0] (receive FIFO unread message counter) value in the RSCAN0RFSTSx register is decremented. Read the RSCAN0RFIDx, RSCAN0RFPTRx, RSCAN0RFDF0x, and RSCAN0RFDF1x registers to read messages in the receive FIFO buffer, and then write FF_H to the RFPC[7:0] bits.

When writing FF_H to these bits, make sure that the RFE bit in the RSCAN0RFCCx register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RSCAN0RFSTSx register is 0 (the receive FIFO buffer contains unread messages).

20.3.29 RSCAN0RFIDx — Receive FIFO Buffer Access ID Register (x = 0 to 7)

Access: RSCAN0RFIDx register is a read-only register that can be read in 32-bit units.
RSCAN0RFIDxL, RSCAN0RFIDxH registers are the read-only registers that can be read in 16-bit units.
RSCAN0RFIDxLL, RSCAN0RFIDxLH, RSCAN0RFIDxHL, RSCAN0RFIDxHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0RFIDx: <RSCAN0_base> + $0E00_H$ + (10_H × x)

RSCAN0RFIDxL: <RSCAN0_base> + $0E00_H$ + (10_H × x),
RSCAN0RFIDxH: <RSCAN0_base> + $0E02_H$ + (10_H × x)

RSCAN0RFIDxLL: <RSCAN0_base> + $0E00_H$ + (10_H × x),
RSCAN0RFIDxLH: <RSCAN0_base> + $0E01_H$ + (10_H × x),
RSCAN0RFIDxHL: <RSCAN0_base> + $0E02_H$ + (10_H × x),
RSCAN0RFIDxHH: <RSCAN0_base> + $0E03_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFIDE	RFRTR	—	RFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.44 RSCAN0RFIDx Register Contents

Bit Position	Bit Name	Function
31	RFIDE	Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	RFRTR	Receive FIFO Buffer RTR 0: Data frame 1: Remote frame
29	Reserved	When read, the value after reset is returned.
28 to 0	RFID[28:0]	Receive FIFO Buffer ID Data The standard ID or extended ID of received message can be read. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

RFRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer.

RFID[28:0] Bits

These bits indicate the ID of the message stored in the receive FIFO buffer.

20.3.30 RSCAN0RFPTRx — Receive FIFO Buffer Access Pointer Register (x = 0 to 7)

Access: RSCAN0RFPTRx register is a read-only register that can be read in 32-bit units.
RSCAN0RFPTRxL, RSCAN0RFPTRxH registers are the read-only registers that can be read in 16-bit units.
RSCAN0RFPTRxLL, RSCAN0RFPTRxLH, RSCAN0RFPTRxHL, RSCAN0RFPTRxHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0RFPTRx: <RSCAN0_base> + $0E04_H$ + (10_H × x)

RSCAN0RFPTRxL: <RSCAN0_base> + $0E04_H$ + (10_H × x),
RSCAN0RFPTRxH: <RSCAN0_base> + $0E06_H$ + (10_H × x)

RSCAN0RFPTRxLL: <RSCAN0_base> + $0E04_H$ + (10_H × x),
RSCAN0RFPTRxLH: <RSCAN0_base> + $0E05_H$ + (10_H × x),
RSCAN0RFPTRxHL: <RSCAN0_base> + $0E06_H$ + (10_H × x),
RSCAN0RFPTRxHH: <RSCAN0_base> + $0E07_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDLC[3:0]				RFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.45 RSCAN0RFPTRx Register Contents

Bit Position	Bit Name	Function
31 to 28	RFDLC[3:0]	Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RFPTR[11:0]	Receive FIFO Buffer Label Data Label information of the received message can be read.
15 to 0	RFTS[15:0]	Receive FIFO Buffer Timestamp Data Timestamp value of the received message can be read.

RFDLC[3:0] Bits

These bits contain the data length of the message stored in the receive FIFO buffer.

RFPTR[11:0] Bits

These bits contain the label information of the message stored in the receive FIFO buffer.

RFTS[15:0] Bits

These bits contain the timestamp value of the message stored in the receive FIFO buffer.

20.3.31 RSCAN0RFDF0x — Receive FIFO Buffer Access Data Field 0 Register (x = 0 to 7)

Access: RSCAN0RFDF0x register is a read-only register that can be read in 32-bit units.
RSCAN0RFDF0xL, RSCAN0RFDF0xH registers are the read-only registers that can be read in 16-bit units.
RSCAN0RFDF0xLL, RSCAN0RFDF0xLH, RSCAN0RFDF0xHL, RSCAN0RFDF0xHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0RFDF0x: <RSCAN0_base> + $0E08_H$ + (10_H × x)

RSCAN0RFDF0xL: <RSCAN0_base> + $0E08_H$ + (10_H × x),
RSCAN0RFDF0xH: <RSCAN0_base> + $0E0A_H$ + (10_H × x)

RSCAN0RFDF0xLL: <RSCAN0_base> + $0E08_H$ + (10_H × x),
RSCAN0RFDF0xLH: <RSCAN0_base> + $0E09_H$ + (10_H × x),
RSCAN0RFDF0xHL: <RSCAN0_base> + $0E0A_H$ + (10_H × x),
RSCAN0RFDF0xHH: <RSCAN0_base> + $0E0B_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB3[7:0]								RFDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB1[7:0]								RFDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.46 RSCAN0RFDF0x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB3[7:0]	Receive FIFO Buffer Data Byte 3
23 to 16	RFDB2[7:0]	Receive FIFO Buffer Data Byte 2
15 to 8	RFDB1[7:0]	Receive FIFO Buffer Data Byte 1
7 to 0	RFDB0[7:0]	Receive FIFO Buffer Data Byte 0 Data for a message stored in the receive FIFO buffer can be read.

When the RFDLC[3:0] bit value in the RSCAN0RFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

20.3.32 RSCAN0RFDF1x — Receive FIFO Buffer Access Data Field 1 Register (x = 0 to 7)

Access: RSCAN0RFDF1x register is a read-only register that can be read in 32-bit units.
RSCAN0RFDF1xL, RSCAN0RFDF1xH registers are the read-only registers that can be read in 16-bit units.
RSCAN0RFDF1xLL, RSCAN0RFDF1xLH, RSCAN0RFDF1xHL, RSCAN0RFDF1xHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0RFDF1x: <RSCAN0_base> + $0E0C_H$ + (10_H × x)

RSCAN0RFDF1xL: <RSCAN0_base> + $0E0C_H$ + (10_H × x),
RSCAN0RFDF1xH: <RSCAN0_base> + $0E0E_H$ + (10_H × x)

RSCAN0RFDF1xLL: <RSCAN0_base> + $0E0C_H$ + (10_H × x),
RSCAN0RFDF1xLH: <RSCAN0_base> + $0E0D_H$ + (10_H × x),
RSCAN0RFDF1xHL: <RSCAN0_base> + $0E0E_H$ + (10_H × x),
RSCAN0RFDF1xHH: <RSCAN0_base> + $0E0F_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB7[7:0]								RFDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB5[7:0]								RFDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.47 RSCAN0RFDF1x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB7[7:0]	Receive FIFO Buffer Data Byte 7
23 to 16	RFDB6[7:0]	Receive FIFO Buffer Data Byte 6
15 to 8	RFDB5[7:0]	Receive FIFO Buffer Data Byte 5
7 to 0	RFDB4[7:0]	Receive FIFO Buffer Data Byte 4 Data for a message stored in the receive FIFO buffer can be read.

When the RFDLC[3:0] bit value in the RSCAN0RFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

20.3.33 RSCAN0CFCCk — Transmit/receive FIFO Buffer Configuration and Control Register k (k = 0 to 17)

Access: RSCAN0CFCCk register can be read or written in 32-bit units.
RSCAN0CFCCkL, RSCAN0CFCCkH registers can be read or written in 16-bit units.
RSCAN0CFCCkLL, RSCAN0CFCCkLH, RSCAN0CFCCkHL, RSCAN0CFCCkHH registers can be read or written in 8-bit units.

Address: RSCAN0CFCCk: <RSCAN0_base> + 0118_H + (04_H × k)

RSCAN0CFCCkL: <RSCAN0_base> + 0118_H + (04_H × k),
RSCAN0CFCCkH: <RSCAN0_base> + $011A_H$ + (04_H × k)

RSCAN0CFCCkLL: <RSCAN0_base> + 0118_H + (04_H × k),
RSCAN0CFCCkLH: <RSCAN0_base> + 0119_H + (04_H × k),
RSCAN0CFCCkHL: <RSCAN0_base> + $011A_H$ + (04_H × k),
RSCAN0CFCCkHH: <RSCAN0_base> + $011B_H$ + (04_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFITT[7:0]								CFTML[3:0]				CFITR	CFITSS	CFM[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFIGCV[2:0]			CFIM	—	CFDC[2:0]			—	—	—	—	—	CFTXIE	CFRXIE	CFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 20.48 RSCAN0CFCCk Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	CFITT[7:0]	Set a message transmission interval. Set Value: 00_H to FF_H
23 to 20	CFTML[3:0]	Transmit Buffer Link Configuration Set the transmit buffer number to be linked to the transmit/receive FIFO buffer.
19	CFITR	Transmit/Receive FIFO Interval Timer Resolution 0: Clock dividing pclk/2 by (ITRCP [15:0] bits) 1: Clock dividing pclk/2 by (ITRCP [15:0] bits × 10)
18	CFITSS	Transmit/Receive FIFO Interval Timer Clock Source Select 0: Interval timer clock source selected by the CFITR bit 1: Interval timer clock source is the bit time clock for the channel to which the FIFO is linked.
17, 16	CFM[1:0]	Transmit/Receive FIFO Mode Select b17 b16 0 0: Receive mode 0 1: Transmit mode 1 0: Gateway mode 1 1: Setting prohibited
15 to 13	CFIGCV[2:0]	Transmit/Receive FIFO Receive Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.

Table 20.48 RSCAN0CFCCk Register Contents (2/2)

Bit Position	Bit Name	Function
12	CFIM	Transmit/Receive FIFO Interrupt Source Select 0: • Receive mode/gateway mode When the number of received messages has met the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated. • Transmit mode/gateway mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated. 1: • Receive mode/gateway mode A FIFO receive interrupt request is generated each time a message has been received. • Transmit mode/gateway mode A FIFO transmit interrupt request is generated each time a message has been transmitted.
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 8	CFDC[2:0]	Transmit/Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	CFTXIE	Transmit/Receive FIFO Transmit Interrupt Enable 0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled.
1	CFRXIE	Transmit/Receive FIFO Receive Interrupt Enable 0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled.
0	CFE	Transmit/Receive FIFO Buffer Enable 0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used.

CFITT[7:0] Bits

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode).

Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITT[7:0] bits.

CFTML[3:0] Bits

These bits are used to set the number of transmit buffer on the channel which will be linked to transmit/receive FIFO buffer k when the CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode). There are three transmit/receive FIFO buffers per channel, so channel number m of FIFO buffer k is calculated as m = k/3 (integer division). The actual assigned transmit buffer number p linked to FIFO buffer k will be ((16 × m) + CFTML[3:0]).

See **Table 20.13** and **Table 20.14**, as for the relationship between transmit/receive FIFO buffer k and transmit buffer p.

Setting the CFDC[2:0] bits to 001_B or more enables the setting of the CFTML[3:0] bits.

Do not link to any transmit buffer which is already allocated to a transmit queue on the identical channel or to another transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFITR Bit

This bit is enabled when the CFITSS bit is 0.

When this bit is 0, the interval timer clock source is the pclk/2 clock divided by the value of the ITRCP[15:0] bits in the RSCAN0GCFG register.

When this bit is 1, the interval timer clock source is the pclk/2 clock divided by (the value of the ITRCP[15:0] bits in the RSCAN0GCFG register × 10).

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFITSS Bit

When this bit is 0, the clock selected by the CFITR bit is the count source of the interval timer.

When this bit is 1, the bit time clock of the channel to which the FIFO is linked is the count source of the interval timer.

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFM[1:0] Bits

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

CFIGCV[2:0] Bits

These bits are used to specify the number of received messages for generating a transmit/receive FIFO receive interrupt request when the CFM[1:0] bits are set to 00_B (receive mode) or 10_B (gateway mode) and the CFIM bit is set to 0 with a fraction for the total number of buffers (the setting of CFDC[2:0]).

When the CFDC[2:0] bits are set to 001_B (4 messages), set the CFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B.

Modify these bits only in global reset mode.

CFIM Bit

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

CFDC[2:0] Bits

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. When these bits are set to 000_B, do not use a transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFTXIE Bit

When this bit is set to 1 and the CFTXIF flag in the RSCAN0CFSTSk register is set to 1, a transmit/receive FIFO transmit interrupt request is generated.

Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

CFRXIE Bit

When this bit is set to 1 and the CFRXIF flag in the RSCAN0CFSTSk register is set to 1, a transmit/receive FIFO receive interrupt request is generated.

Modify this bit with the CFE bit set to 0.

CFE Bit

Setting this bit to 1 makes transmit/receive FIFO buffers available.

When this bit is set to 0 in transmit mode or gateway mode, if a message in the transmit/receive FIFO buffer is being transmitted or will be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission of that message, or upon detection of a CAN bus error, or arbitration-lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is cleared to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode or gateway mode: Channel reset mode

Modify this bit in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode or gateway mode: Channel communication mode or channel halt mode

After all other bits in the RSCANnCFCCk register have been set, set this bit to 1 by using other instructions.

20.3.34 RSCAN0CFSTSk — Transmit/receive FIFO Buffer Status Register (k = 0 to 17)

Access: RSCAN0CFSTSk register can be read or written in 32-bit units.
RSCAN0CFSTSkL register can be read or written in 16-bit units.
RSCAN0CFSTSkLL register can be read or written in 8-bit units.
RSCAN0CFSTSkLH register is a read-only register that can be read in 8-bit units.

Address: RSCAN0CFSTSk: <RSCAN0_base> + 0178_H + (04_H × k)

RSCAN0CFSTSkL: <RSCAN0_base> + 0178_H + (04_H × k)

RSCAN0CFSTSkLL: <RSCAN0_base> + 0178_H + (04_H × k),
RSCAN0CFSTSkLH: <RSCAN0_base> + 0179_H + (04_H × k)

Value after reset: $0000\ 0001_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFMC[7:0]								—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.49 RSCAN0CFSTSk Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 8	CFMC[7:0]	Transmit/Receive FIFO Message Counter The number of messages stored in the transmit/receive FIFO buffer.
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	CFTXIF	Transmit/Receive FIFO Transmit Interrupt Request Flag 0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present.
3	CFRXIF	Transmit/Receive FIFO Receive Interrupt Request Flag 0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present.
2	CFMLT	Transmit/Receive FIFO Message Lost Flag 0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost.
1	CFFLL	Transmit/Receive FIFO Buffer Full Status Flag 0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full.
0	CFEMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty).

CFMC[7:0] Bits

The CFMC[7:0] bits indicate the following values that depend on the setting of the CFM[1:0] bits in the RSCAN0CFCCk register.

- When CFM[1:0] bits are set to 01_B (transmit mode): Number of untransmitted messages in the

buffer

- When CFM[1:0] bits are set to 00_B (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] bits are set to 10_B (gateway mode): Number of untransmitted received messages in the buffer

These bits are cleared to 0 when any of the following conditions is met.

- When CFM[1:0] bits are set to 00_B: In global reset mode
- When CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

CFTXIF Flag

The CFTXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 01_B or 10_B, and the factor selected by the CFIM bit in the RSCAN0CFCCk register occurs

The CFTXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFTXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFRXIF Flag

The CFRXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B, and the factor selected by the CFIM bit in the RSCAN0CFCCk register occurs

The CFRXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFRXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFMLT Flag

The CFMLT flag is set to 1 when any of the following conditions is met.

- When an attempt is made to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

The CFMLT flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFMLT flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFFLL Flag

The CFFLL flag is set to 1 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFDC[2:0] bits in the RSCAN0CFCCk register.

The CFFLL flag is cleared to 0 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFDC[2:0] bits.
- When the CFE bit in the RSCAN0CFCCk register is 0 (no transmit/receive FIFO buffer is used): When not in the transmit abort
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

CFEMP Flag

The CFEMP flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B: All messages have been read, or in global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: All messages have been transmitted, or in channel reset mode
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not in the transmit abort

The CFEMP flag is cleared to 0 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B: At least one received message has been stored in the transmit/receive FIFO buffer.
- When the CFM[1:0] bits are set to 01_B: A value of FF_H has been written to the RSCAN0CFPCTRk register after data was written to the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers.

NOTE

To clear CFTXIF, CFRXIF, or CFMLT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

20.3.35 RSCAN0CFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register (k = 0 to 17)

Access: RSCAN0CFPCTRk register is a write-only register that can be written in 32-bit units.
RSCAN0CFPCTRkL register is a write-only register that can be written in 16-bit units.
RSCAN0CFPCTRkLL register is a write-only register that can be written in 8-bit units.

Address: RSCAN0CFPCTRk: <RSCAN0_base> + $01D8_H$ + (04_H × k)

RSCAN0CFPCTRkL: <RSCAN0_base> + $01D8_H$ + (04_H × k)

RSCAN0CFPCTRkLL: <RSCAN0_base> + $01D8_H$ + (04_H × k)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.50 RSCAN0CFPCTRk Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	CFPC[7:0]	Transmit/Receive FIFO Pointer Control • Receive mode: Writing FF_H to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. • Transmit mode: Writing FF_H to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer. • Gateway mode: Setting prohibited

CFPC[7:0] Bits

- Receive mode (CFM[1:0] bits in the RSCAN0CFCCk register are set to 00_B):
  Writing FF_H to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[7:0] bit value (transmit/receive FIFO message counter) in the RSCAN0CFSTSk register is decremented. Read the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers to read messages from the transmit/receive FIFO buffer, and then write FF_H to the CFPC[7:0] bits.
  When writing FF_H to these bits, make sure that the CFE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the RSCAN0CFSTSk register is 0 (the transmit/receive FIFO buffer contains messages).

- Transmit mode (CFM[1:0] bits in the RSCAN0CFCCk register are set to 01_B):
  Writing FF_H to the CFPC[7:0] bits stores the data written to the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[7:0] bit value is incremented. Write transmit messages to the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers before writing FF_H to the CFPC[7:0] bits.
  When writing FF_H to these bits, make sure that the CFE bit in the RSCAN0CFCCk register is set to 1 and the CFFLL flag in the RSCAN0CFSTSk register is 0 (the transmit/receive FIFO buffer is not full).
- Gateway mode (CFM[1:0] bits in the RSCAN0CFCCk register are set to 10_B):
  Setting prohibited

20.3.36 RSCAN0CFIDk — Transmit/receive FIFO Buffer Access ID Register (k = 0 to 17)

Access: RSCAN0CFIDk register can be read or written in 32-bit units.
RSCAN0CFIDkL, RSCAN0CFIDkH registers can be read or written in 16-bit units.
RSCAN0CFIDkLL, RSCAN0CFIDkLH, RSCAN0CFIDkHL, RSCAN0CFIDkHH registers can be read or written in 8-bit units.

Address: RSCAN0CFIDk: <RSCAN0_base> + $0E80_H$ + (10_H × k)

RSCAN0CFIDkL: <RSCAN0_base> + $0E80_H$ + (10_H × k),
RSCAN0CFIDkH: <RSCAN0_base> + $0E82_H$ + (10_H × k)

RSCAN0CFIDkLL: <RSCAN0_base> + $0E80_H$ + (10_H × k),
RSCAN0CFIDkLH: <RSCAN0_base> + $0E81_H$ + (10_H × k),
RSCAN0CFIDkHL: <RSCAN0_base> + $0E82_H$ + (10_H × k),
RSCAN0CFIDkHH: <RSCAN0_base> + $0E83_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.51 RSCAN0CFIDk Register Contents

Bit Position	Bit Name	Function
31	CFIDE	Transmit/Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	CFRTR	Transmit/Receive FIFO Buffer RTR 0: Data frame 1: Remote frame
29	THLEN	Transmit History Data Store Enable This bit is valid only when the CFM[1:0] bits are set to 01_B (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	CFID[28:0]	Transmit/Receive FIFO Buffer ID Data • When CFM[1:0] bits are set to 01_B (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11. • When CFM[1:0] bits are set to 00_B (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read bits 10 to 0. Bits 28 to 11 are read as 0.

This register is writable only when the CFM[1:0] bits in the RSCAN0CFCCk register are set to 01_B (transmit mode). This register is readable only when the CFM[1:0] bits are set to 00_B (receive mode). This RSCAN0CFIDk register should not be read or written when the CFM[1:0] bits are set to 10_B (gateway mode).

CFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] bits are set to 00_B. When the CFM[1:0] bits are set to 01_B, these bits are used to set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

CFRTR Bit

This bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] bits are set to 00_B. When the CFM[1:0] bits are set to 01_B, this bit is used to set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.

This bit is enabled when the CFM[1:0] bits are set to 01_B (transmit mode).

CFID[28:0] Bits

These bits contain the ID of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] bits are set to 00_B.

When the CFM[1:0] bits are set to 01_B, this bit is used to set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

20.3.37 RSCAN0CFPTRk — Transmit/receive FIFO Buffer Access Pointer Register (k = 0 to 17)

Access: RSCAN0CFPTRk register can be read or written in 32-bit units.
RSCAN0CFPTRkL, RSCAN0CFPTRkH registers can be read or written in 16-bit units.
RSCAN0CFPTRkLL, RSCAN0CFPTRkLH, RSCAN0CFPTRkHL, RSCAN0CFPTRkHH registers can be read or written in 8-bit units.

Address: RSCAN0CFPTRk: <RSCAN0_base> + $0E84_H$ + (10_H × k)

RSCAN0CFPTRkL: <RSCAN0_base> + $0E84_H$ + (10_H × k),
RSCAN0CFPTRkH: <RSCAN0_base> + $0E86_H$ + (10_H × k)

RSCAN0CFPTRkLL: <RSCAN0_base> + $0E84_H$ + (10_H × k),
RSCAN0CFPTRkLH: <RSCAN0_base> + $0E85_H$ + (10_H × k),
RSCAN0CFPTRkHL: <RSCAN0_base> + $0E86_H$ + (10_H × k),
RSCAN0CFPTRkHH: <RSCAN0_base> + $0E87_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDLC[3:0]				CFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.52 RSCAN0CFPTRk Register Contents

Bit Position	Bit Name	Function
31 to 28	CFDLC[3:0]	Transmit/Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	CFPTR[11:0]	Transmit/Receive FIFO Buffer Label Data • When CFM[1:0] bits are set to 01_B (transmit mode): Set the label information to be stored in the transmit history buffer. Only bits CFPTR[7:0] are valid. • When CFM[1:0] bits are set to 00_B (receive mode): The label information of the received message can be read.
15 to 0	CFTS[15:0]	Transmit/Receive FIFO Buffer Timestamp Data These bits are valid only when the CFM[1:0] bits are set to 00_B (receive mode). The timestamp value of the received message can be read.

This register is writable only when the CFM[1:0] bits in the RSCAN0CFCCk register are set to 01_B (transmit mode). This register is readable only when the CFM[1:0] bits are set to 00_B (receive mode). This register should not be read or written when the CFM[1:0] bits are set to 10_B (gateway mode).

CFDLC[3:0] Bits

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] bits are set to 00_B. When the CFM[1:0] bits are set to 01_B, these bits are used to set the data length of the message to be transmitted from the transmit/receive FIFO buffer. If the data length is set to 9 bytes or more, the actual transmit data defaults to 8 bytes.

CFPTR[11:0] Bits

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] bits are set to 00_B. When the CFM[1:0] bits are set to 01_B, the CFPTR[7:0] bit value is stored in the transmit history buffer when message transmission has been completed.

CFTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFM[1:0] bits are set to 00_B.

20.3.38 RSCAN0CFDF0k — Transmit/receive FIFO Buffer Access Data Field 0 Register (k = 0 to 17)

Access: RSCAN0CFDF0k register can be read or written in 32-bit units.
RSCAN0CFDF0kL, RSCAN0CFDF0kH registers can be read or written in 16-bit units.
RSCAN0CFDF0kLL, RSCAN0CFDF0kLH, RSCAN0CFDF0kHL, RSCAN0CFDF0kHH registers can be read or written in 8-bit units.

Address: RSCAN0CFDF0k: <RSCAN0_base> + $0E88_H$ + (10_H × k)

RSCAN0CFDF0kL: <RSCAN0_base> + $0E88_H$ + (10_H × k),
RSCAN0CFDF0kH: <RSCAN0_base> + $0E8A_H$ + (10_H × k)

RSCAN0CFDF0kLL: <RSCAN0_base> + $0E88_H$ + (10_H × k),
RSCAN0CFDF0kLH: <RSCAN0_base> + $0E89_H$ + (10_H × k),
RSCAN0CFDF0kHL: <RSCAN0_base> + $0E8A_H$ + (10_H × k),
RSCAN0CFDF0kHH: <RSCAN0_base> + $0E8B_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB3[7:0]								CFDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB1[7:0]								CFDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.53 RSCAN0CFDF0k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB3[7:0]	Transmit/Receive FIFO Buffer Data Byte 3 Transmit/Receive FIFO Buffer Data Byte 2 Transmit/Receive FIFO Buffer Data Byte 1 Transmit/Receive FIFO Buffer Data Byte 0 • When CFM[1:0] bits are set to 01_B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] bits are set to 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.
23 to 16	CFDB2[7:0]	
15 to 8	CFDB1[7:0]	
7 to 0	CFDB0[7:0]	

This register is writable only when the CFM[1:0] bits in the RSCAN0CFCCk register are set to 01_B (transmit mode).

This register is readable only when the CFM[1:0] bits are set to 00_B (receive mode). When the CFDLC[3:0] bit value in the RSCAN0CFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

This register should not be read or written when the CFM[1:0] bits are set to 10_B (gateway mode).

20.3.39 RSCAN0CFDF1k — Transmit/receive FIFO Buffer Access Data Field 1 Register (k = 0 to 17)

Access: RSCAN0CFDF1k register can be read or written in 32-bit units.
RSCAN0CFDF1kL, RSCAN0CFDF1kH registers can be read or written in 16-bit units.
RSCAN0CFDF1kLL, RSCAN0CFDF1kLH, RSCAN0CFDF1kHL, RSCAN0CFDF1kHH registers can be read or written in 8-bit units.

Address: RSCAN0CFDF1k: <RSCAN0_base> + $0E8C_H$ + (10_H × k)

RSCAN0CFDF1kL: <RSCAN0_base> + $0E8C_H$ + (10_H × k),
RSCAN0CFDF1kH: <RSCAN0_base> + $0E8E_H$ + (10_H × k)

RSCAN0CFDF1kLL: <RSCAN0_base> + $0E8C_H$ + (10_H × k),
RSCAN0CFDF1kLH: <RSCAN0_base> + $0E8D_H$ + (10_H × k),
RSCAN0CFDF1kHL: <RSCAN0_base> + $0E8E_H$ + (10_H × k),
RSCAN0CFDF1kHH: <RSCAN0_base> + $0E8F_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB7[7:0]								CFDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB5[7:0]								CFDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.54 RSCAN0CFDF1k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB7[7:0]	Transmit/Receive FIFO Buffer Data Byte 7 Transmit/Receive FIFO Buffer Data Byte 6 Transmit/Receive FIFO Buffer Data Byte 5 Transmit/Receive FIFO Buffer Data Byte 4 • When CFM[1:0] bits are set to 01_B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] bits are set to 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.
23 to 16	CFDB6[7:0]	
15 to 8	CFDB5[7:0]	
7 to 0	CFDB4[7:0]	

This register is writable only when the CFM[1:0] bits in the RSCAN0CFCCk register are set to 01_B (transmit mode).

This register is readable only when the CFM[1:0] bits are set to 00_B (receive mode). When the CFDLC[3:0] bit value in the RSCAN0CFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

This register should not be read or written when the CFM[1:0] bits are set to 10_B (gateway mode).

20.3.40 RSCAN0FESTS — FIFO Empty Status Register

Access: RSCAN0FESTS register is a read-only register that can be read in 32-bit units.
RSCAN0FESTSL, RSCAN0FESTSH registers are the read-only registers that can be read in 16-bit units.
RSCAN0FESTSLL, RSCAN0FESTSLH, RSCAN0FESTSHL, RSCAN0FESTSHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0FESTS: <RSCAN0_base> + 0238_H

RSCAN0FESTSL: <RSCAN0_base> + 0238_H, RSCAN0FESTSH: <RSCAN0_base> + $023A_H$

RSCAN0FESTSLL: <RSCAN0_base> + 0238_H, RSCAN0FESTSLH: <RSCAN0_base> + 0239_H,
RSCAN0FESTSHL: <RSCAN0_base> + $023A_H$, RSCAN0FESTSHH: <RSCAN0_base> + $023B_H$

Value after reset: 03FF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CF17EMP	CF16EMP	CF15EMP	CF14EMP	CF13EMP	CF12EMP	CF11EMP	CF10EMP	CF9EMP	CF8EMP
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7EMP	CF6EMP	CF5EMP	CF4EMP	CF3EMP	CF2EMP	CF1EMP	CF0EMP	RF7EMP	RF6EMP	RF5EMP	RF4EMP	RF3EMP	RF2EMP	RF1EMP	RF0EMP
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.55 RSCAN0FESTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	CF17EMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. (k = 0 to 17)
24	CF16EMP	
23	CF15EMP	
22	CF14EMP	
21	CF13EMP	
20	CF12EMP	
19	CF11EMP	
18	CF10EMP	
17	CF9EMP	
16	CF8EMP	
15	CF7EMP	
14	CF6EMP	
13	CF5EMP	
12	CF4EMP	
11	CF3EMP	
10	CF2EMP	
9	CF1EMP	
8	CF0EMP	

Table 20.55 RSCAN0FESTS Register Contents (2/2)

Bit Position	Bit Name	Function
7	RF7EMP	Receive FIFO Buffer Empty Status Flag 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message (buffer empty). (x = 0 to 7)
6	RF6EMP	
5	RF5EMP	
4	RF4EMP	
3	RF3EMP	
2	RF2EMP	
1	RF1EMP	
0	RF0EMP	

The RSCAN0FESTS register is set to 03FF $FFFF_H$ in global reset mode.

CFkEMP Flag (k = 0 to 17)

The CFkEMP flag is set to 1 when the CFEMP flag in the RSCAN0CFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)). When the CFEMP flag is cleared to 0 (the transmit/receive FIFO buffer contains messages), the CFkEMP flag is cleared to 0.

RFxEMP Flag (x = 0 to 7)

The RFxEMP flag is set to 1 when the RFEMP flag in the RSCAN0RFSTSx register is set to 1 (the receive FIFO buffer contains no unread message). When the RFEMP flag is cleared to 0 (the receive FIFO buffer contains unread messages), the RFxEMP flag is cleared to 0.

20.3.41 RSCAN0FFSTS — FIFO Full Status Register

Access: RSCAN0FFSTS register is a read-only register that can be read in 32-bit units.
RSCAN0FFSTSL, RSCAN0FFSTSH registers are the read-only registers that can be read in 16-bit units.
RSCAN0FFSTSLL, RSCAN0FFSTSLH, RSCAN0FFSTSHL, RSCAN0FFSTSHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0FFSTS: <RSCAN0_base> + $023C_H$

RSCAN0FFSTSL: <RSCAN0_base> + $023C_H$, RSCAN0FFSTSH: <RSCAN0_base> + $023E_H$

RSCAN0FFSTSLL: <RSCAN0_base> + $023C_H$, RSCAN0FFSTSLH: <RSCAN0_base> + $023D_H$,
RSCAN0FFSTSHL: <RSCAN0_base> + $023E_H$, RSCAN0FFSTSHH: <RSCAN0_base> + $023F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CF17FLL	CF16FLL	CF15FLL	CF14FLL	CF13FLL	CF12FLL	CF11FLL	CF10FLL	CF9FLL	CF8FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7FLL	CF6FLL	CF5FLL	CF4FLL	CF3FLL	CF2FLL	CF1FLL	CF0FLL	RF7FLL	RF6FLL	RF5FLL	RF4FLL	RF3FLL	RF2FLL	RF1FLL	RF0FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.56 RSCAN0FFSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	CF17FLL	Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive buffer k is not full. 1: Transmit/receive buffer k is full. (k = 0 to 17)
24	CF16FLL	
23	CF15FLL	
22	CF14FLL	
21	CF13FLL	
20	CF12FLL	
19	CF11FLL	
18	CF10FLL	
17	CF9FLL	
16	CF8FLL	
15	CF7FLL	
14	CF6FLL	
13	CF5FLL	
12	CF4FLL	
11	CF3FLL	
10	CF2FLL	
9	CF1FLL	
8	CF0FLL	

Table 20.56 RSCAN0FFSTS Register Contents (2/2)

Bit Position	Bit Name	Function
7	RF7FLL	Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full. (x = 0 to 7)
6	RF6FLL	
5	RF5FLL	
4	RF4FLL	
3	RF3FLL	
2	RF2FLL	
1	RF1FLL	
0	RF0FLL	

The RSCAN0FFSTS register is cleared to 0000 0000_H in global reset mode.

CFkFLL Flag (k = 0 to 17)

The CFkFLL flag is set to 1 when the CFFLL flag in the RSCAN0CFSTSk register is set to 1 (the transmit/receive FIFO buffer is full).
When the CFFLL flag is cleared to 0 (the transmit/receive FIFO buffer is not full), the CFkFLL flag is cleared to 0.

RFxFLL Flag (x = 0 to 7)

The RFxFLL flag is set to 1 when the RFFLL flag in the RSCAN0RFSTSx register is set to 1 (the receive FIFO buffer is full). When the RFFLL flag is cleared to 0 (the receive FIFO buffer is not full), the RFxFLL flag is cleared to 0.

20.3.42 RSCAN0FMSTS — FIFO Message Lost Status Register

Access: RSCAN0FMSTS register is a read-only register that can be read in 32-bit units.
RSCAN0FMSTSL, RSCAN0FMSTSH registers are the read-only registers that can be read in 16-bit units.
RSCAN0FMSTSLL, RSCAN0FMSTSLH, RSCAN0FMSTSHL, RSCAN0FMSTSHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0FMSTS: <RSCAN0_base> + 0240_H

RSCAN0FMSTSL: <RSCAN0_base> + 0240_H, RSCAN0FMSTSH: <RSCAN0_base> + 0242_H

RSCAN0FMSTSLL: <RSCAN0_base> + 0240_H, RSCAN0FMSTSLH: <RSCAN0_base> + 0241_H,
RSCAN0FMSTSHL: <RSCAN0_base> + 0242_H, RSCAN0FMSTSHH: <RSCAN0_base> + 0243_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CF17MLT	CF16MLT	CF15MLT	CF14MLT	CF13MLT	CF12MLT	CF11MLT	CF10MLT	CF9MLT	CF8MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7MLT	CF6MLT	CF5MLT	CF4MLT	CF3MLT	CF2MLT	CF1MLT	CF0MLT	RF7MLT	RF6MLT	RF5MLT	RF4MLT	RF3MLT	RF2MLT	RF1MLT	RF0MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.57 RSCAN0FMSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	CF17MLT	Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. (k = 0 to 17)
24	CF16MLT	
23	CF15MLT	
22	CF14MLT	
21	CF13MLT	
20	CF12MLT	
19	CF11MLT	
18	CF10MLT	
17	CF9MLT	
16	CF8MLT	
15	CF7MLT	
14	CF6MLT	
13	CF5MLT	
12	CF4MLT	
11	CF3MLT	
10	CF2MLT	
9	CF1MLT	
8	CF0MLT	

Table 20.57 RSCAN0FMSTS Register Contents (2/2)

Bit Position	Bit Name	Function
7	RF7MLT	Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. (x = 0 to 7)
6	RF6MLT	
5	RF5MLT	
4	RF4MLT	
3	RF3MLT	
2	RF2MLT	
1	RF1MLT	
0	RF0MLT	

The RSCAN0FMSTS register is cleared to 0000 0000_H in global reset mode.

CFkMLT Flag (k = 0 to 17)

The CFkMLT flag is set to 1 when the CFMLT flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CFkMLT flag is cleared to 0.

RFxMLT Flag (x = 0 to 7)

The RFxMLT flag is set to 1 when the RFMLT flag in the RSCAN0RFSTSx register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFxMLT flag is cleared to 0.

20.3.43 RSCAN0RFISTS — Receive FIFO Buffer Interrupt Flag Status Register

Access: RSCAN0RFISTS register is a read-only register that can be read in 32-bit units.
RSCAN0RFISTSL register is a read-only register that can be read in 16-bit units.
RSCAN0RFISTSLL register is a read-only register that can be read in 8-bit units.

Address: RSCAN0RFISTS: <RSCAN0_base> + 0244_H

RSCAN0RFISTSL: <RSCAN0_base> + 0244_H

RSCAN0RFISTSLL: <RSCAN0_base> + 0244_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RF7IF	RF6IF	RF5IF	RF4IF	RF3IF	RF2IF	RF1IF	RF0IF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.58 RSCAN0RFISTS Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	RF7IF	Receive FIFO Buffer Interrupt Request Status Flag 0: No receive FIFO buffer x interrupt request is present. 1: A receive FIFO buffer x interrupt request is present. (x = 0 to 7)
6	RF6IF	
5	RF5IF	
4	RF4IF	
3	RF3IF	
2	RF2IF	
1	RF1IF	
0	RF0IF	

The RSCAN0RFISTS register is cleared to $0000\ 0000_H$ in global reset mode.

RFxIF Flag (x = 0 to 7)

The RFxIF flag is set to 1 when the RFIF flag in the RSCAN0RFSTSx register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RFxIF flag is cleared to 0.

20.3.44 RSCAN0CFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register

Access: RSCAN0CFRISTS register is a read-only register that can be read in 32-bit units.
RSCAN0CFRISTSL, RSCAN0CFRISTSH registers are the read-only registers that can be read in 16-bit units.
RSCAN0CFRISTSLL, RSCAN0CFRISTSLH, RSCAN0CFRISTSHL registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0CFRISTS: <RSCAN0_base> + 0248_H

RSCAN0CFRISTSL: <RSCAN0_base> + 0248_H, RSCAN0CFRISTSH: <RSCAN0_base> + $024A_H$

RSCAN0CFRISTSLL: <RSCAN0_base> + 0248_H, RSCAN0CFRISTSLH: <RSCAN0_base> + 0249_H, RSCAN0CFRISTSHL: <RSCAN0_base> + $024A_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF17RXIF	CF16RXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF15RXIF	CF14RXIF	CF13RXIF	CF12RXIF	CF11RXIF	CF10RXIF	CF9RXIF	CF8RXIF	CF7RXIF	CF6RXIF	CF5RXIF	CF4RXIF	CF3RXIF	CF2RXIF	CF1RXIF	CF0RXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.59 RSCAN0CFRISTS Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	CF17RXIF	Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k receive interrupt request is present. 1: A transmit/receive FIFO buffer k receive interrupt request is present. (k = 0 to 17)
16	CF16RXIF	
15	CF15RXIF	
14	CF14RXIF	
13	CF13RXIF	
12	CF12RXIF	
11	CF11RXIF	
10	CF10RXIF	
9	CF9RXIF	
8	CF8RXIF	
7	CF7RXIF	
6	CF6RXIF	
5	CF5RXIF	
4	CF4RXIF	
3	CF3RXIF	
2	CF2RXIF	
1	CF1RXIF	
0	CF0RXIF	

The RSCAN0CFRISTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkRXIF Flag (k = 0 to 17)

The CFkRXIF flag is set to 1 when the CFRXIF flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CFkRXIF flag is cleared to 0.

20.3.45 RSCAN0CFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register

Access: RSCAN0CFTISTS register is a read-only register that can be read in 32-bit units.
RSCAN0CFTISTSL, RSCAN0CFTISTSH registers are the read-only registers that can be read in 16-bit units.
RSCAN0CFTISTSLL, RSCAN0CFTISTSLH, RSCAN0CFTISTSHL registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0CFTISTS: <RSCAN0_base> + $024C_H$

RSCAN0CFTISTSL: <RSCAN0_base> + $024C_H$, RSCAN0CFTISTSH: <RSCAN0_base> + $024E_H$

RSCAN0CFTISTSLL: <RSCAN0_base> + $024C_H$, RSCAN0CFTISTSLH: <RSCAN0_base> + $024D_H$, RSCAN0CFTISTSHL: <RSCAN0_base> + $024E_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF17TXIF	CF16TXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF15TXIF	CF14TXIF	CF13TXIF	CF12TXIF	CF11TXIF	CF10TXIF	CF9TXIF	CF8TXIF	CF7TXIF	CF6TXIF	CF5TXIF	CF4TXIF	CF3TXIF	CF2TXIF	CF1TXIF	CF0TXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.60 RSCAN0CFTISTS Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned.
17	CF17TXIF	Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k transmit interrupt request is present. 1: A transmit/receive FIFO buffer k transmit interrupt request is present. (k = 0 to 17)
16	CF16TXIF	
15	CF15TXIF	
14	CF14TXIF	
13	CF13TXIF	
12	CF12TXIF	
11	CF11TXIF	
10	CF10TXIF	
9	CF9TXIF	
8	CF8TXIF	
7	CF7TXIF	
6	CF6TXIF	
5	CF5TXIF	
4	CF4TXIF	
3	CF3TXIF	
2	CF2TXIF	
1	CF1TXIF	
0	CF0TXIF	

The RSCAN0CFTISTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkTXIF Flag (k = 0 to 17)

The CFkTXIF flag is set to 1 when the CFTXIF flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO transmit interrupt request is present). When the CFTXIF flag is cleared to 0, the CFkTXIF flag is cleared to 0.

20.3.46 RSCAN0TMCp — Transmit Buffer Control Register (p = 0 to 95)

Access: RSCAN0TMCp register can be read or written in 8-bit units.

Address: RSCAN0TMCp: <RSCAN0_base> + 0250_H + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W*1	R/W*1

Note 1. The only effective value for writing to this bit is 1, which sets the bit. Otherwise writing to the bit results in retention of its state.

Table 20.61 RSCAN0TMCp Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TMOM	One-Shot Transmission Enable 0: One-shot transmission is disabled. 1: One-shot transmission is enabled.
1	TMTAR	Transmit Abort Request 0: Transmit abort is not requested. 1: Transmit abort is requested.
0	TMTR	Transmit Request 0: Transmission is not requested. 1: Transmission is requested.

When the RSCAN0TMCp register meets any of the following conditions, set it to 00_H.

- The RSCAN0TMCp register corresponds to the transmit buffer number selected by the CFTML[3:0] bits in the RSCAN0CFCCk register (p = m × 16 + the value of CFTML[3:0] bits).
- The RSCAN0TMCp register corresponds to the transmit buffer allocated to the transmit queue by the TXQDC[3:0] bits in the RSCAN0TXQCCm (m = 0 to 5) register (p = (m × 16 + 15) to (m × 16 + 15 – the value of TXQDC[3:0] bits)).

Bits in the RSCAN0TMCp register are all cleared to 0 in channel reset mode. Modify the RSCAN0TMCp register in channel communication mode or channel halt mode.

TMOM Bit

Setting this bit to 1 enables one-shot transmission. When transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMTRM flag in the RSCAN0TMSTSp register is set to 0. Set the TMOM bit to 1 together with the TMTR bit.

TMTAR Bit

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or one that will be transmitted next cannot be aborted.

The TMTAR bit can be set to 1 when TMTR bit is 1.

The TMTAR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration lost has been detected.

If this bit becomes 0 at the same time as the program writes 1 to this bit, this bit becomes 0.

TMTR Bit

Setting this bit to 1 transmits the message stored in the transmit buffer.

The TMTR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed after the TMTAR bit was set to 1.
- An error or arbitration-lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the value of TMTRF[1:0] flags in the RSCAN0TMSTSp register is 00_B.

20.3.47 RSCAN0TMSTSp — Transmit Buffer Status Register (p = 0 to 95)

Access: RSCAN0TMSTSp register can be read or written in 8-bit units.

Address: RSCAN0TMSTSp: <RSCAN0_base> + $02D0_H$ + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R

Table 20.62 RSCAN0TMSTSp Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	TMTARM	Transmit Buffer Transmit Abort Request Status Flag 0: No transmit abort request is present. 1: A transmit abort request is present.
3	TMTRM	Transmit Buffer Transmit Request Status Flag 0: No transmit request is present. 1: A transmit request is present.
2, 1	TMTRF[1:0]	Transmit Buffer Transmit Result Status Flag b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request).
0	TMTSTS	Transmit Buffer Transmit Status Flag 0: Transmission is not in progress. 1: Transmission is in progress.

The RSCAN0TMSTSp register is cleared to all 0 in channel reset mode.

TMTARM Flag

The TMTARM flag is set to 1 when the TMTAR bit in the RSCAN0TMCp register is set to 1.

The TMTARM flag is set to 0 when the TMTAR bit in the RSCAN0TMCp register is set to 0.

TMTRM Flag

The TMTRM flag is set to 1 when the TMTR bit in the RSCAN0TMCp register is set to 1.

The TMTRM flag is set to 0 when the TMTR bit in the RSCAN0TMCp register is set to 0.

TMTRF[1:0] Flag

These flags indicate the result of transmission from the transmit buffer.

00_B: Transmission is in progress or no transmit request is present.

01_B: Transmission from the transmit buffer was aborted.

10_B: Transmission has been completed with the TMTAR bit in the RSCAN0TMCp register set to 0 (transmit abort is not requested).

11_B: Transmission has been completed with the TMTAR bit in the RSCAN0TMCp register set to 1 (transmit abort is requested).

Write 00_B to the TMTRF[1:0] flags in channel communication mode or channel halt mode. Do not write any value other than 00_B to these flags.

TMTSTS Flag

This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

20.3.48 RSCAN0TMTRSTSy — Transmit Buffer Transmit Request Status Register (y = 0 to 2)

Access: RSCAN0TMTRSTSy register is a read-only register that can be read in 32-bit units.
RSCAN0TMTRSTSyL, RSCAN0TMTRSTSyH registers are the read-only registers that can be read in 16-bit units.
RSCAN0TMTRSTSyLL, RSCAN0TMTRSTSyLH, RSCAN0TMTRSTSyHL, RSCAN0TMTRSTSyHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0TMTRSTSy: <RSCAN0_base> + 0350_H + (04_H × y)

RSCAN0TMTRSTSyL: <RSCAN0_base> + 0350_H + (04_H × y),
RSCAN0TMTRSTSyH: <RSCAN0_base> + 0352_H + (04_H × y)

RSCAN0TMTRSTSyLL: <RSCAN0_base> + 0350_H + (04_H × y),
RSCAN0TMTRSTSyLH: <RSCAN0_base> + 0351_H + (04_H × y),
RSCAN0TMTRSTSyHL: <RSCAN0_base> + 0352_H + (04_H × y),
RSCAN0TMTRSTSyHH: <RSCAN0_base> + 0353_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTRSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTRSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.63 RSCAN0TMTRSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit request is present. 1: A transmit request is present.
15 to 0	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit request is present. 1: A transmit request is present.

TMTRSTSp Flags (p = 0 to 95)

These flags indicate the status of the TMTR bit in the RSCAN0TMCp register.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag is set to 1.

The corresponding TMTRSTSp flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Table 20.64 shows the bit assignment.

Table 20.64 TMTRSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15
48	3	0
. .	. .	. .
62	3	14
63	3	15
64	4	0
65	4	1
. .	. .	. .
78	4	14
79	4	15
80	5	0
81	5	1
. .	. .	. .
94	5	14
95	5	15

20.3.49 RSCAN0TMTARSTSy — Transmit Buffer Transmit Abort Request Status Register (y = 0 to 2)

Access: RSCAN0TMTARSTSy register is a read-only register that can be read in 32-bit units.
RSCAN0TMTARSTSyL, RSCAN0TMTARSTSyH registers are the read-only registers that can be read in 16-bit units.
RSCAN0TMTARSTSyLL, RSCAN0TMTARSTSyLH, RSCAN0TMTARSTSyHL, RSCAN0TMTARSTSyHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0TMTARSTSy: <RSCAN0_base> + 0360_H + (04_H × y)

RSCAN0TMTARSTSyL: <RSCAN0_base> + 0360_H + (04_H × y),
RSCAN0TMTARSTSyH: <RSCAN0_base> + 0362_H + (04_H × y)

RSCAN0TMTARSTSyLL: <RSCAN0_base> + 0360_H + (04_H × y),
RSCAN0TMTARSTSyLH: <RSCAN0_base> + 0361_H + (04_H × y),
RSCAN0TMTARSTSyHL: <RSCAN0_base> + 0362_H + (04_H × y),
RSCAN0TMTARSTSyHH: <RSCAN0_base> + 0363_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTARSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTARSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.65 RSCAN0TMTARSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit abort request is present. 1: A transmit abort request is present.
15 to 0	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit abort request is present. 1: A transmit abort request is present.

TMTARSTSp Flags (p = 0 to 95)

These flags indicate the status of the TMTAR bit in the RSCAN0TMCp register.

When the TMTAR bit is set to 1 (transmit abort is requested), the corresponding TMTARSTSp flag is set to 1.

The corresponding TMTARSTSp flag is cleared to 0 when the TMTAR bit is set to 0 (transmit abort is not requested) or in channel reset mode.

Table 20.66 shows the bit assignment.

Table 20.66 TMTARSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15
48	3	0
. .	. .	. .
62	3	14
63	3	15
64	4	0
65	4	1
. .	. .	. .
78	4	14
79	4	15
80	5	0
81	5	1
. .	. .	. .
94	5	14
95	5	15

20.3.50 RSCAN0TMTCSTSy — Transmit Buffer Transmit Complete Status Register (y = 0 to 2)

Access: RSCAN0TMTCSTSy register is a read-only register that can be read in 32-bit units.
RSCAN0TMTCSTSyL, RSCAN0TMTCSTSyH registers are the read-only registers that can be read in 16-bit units.
RSCAN0TMTCSTSyLL, RSCAN0TMTCSTSyLH, RSCAN0TMTCSTSyHL, RSCAN0TMTCSTSyHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0TMTCSTSy: <RSCAN0_base> + 0370_H + (04_H × y)

RSCAN0TMTCSTSyL: <RSCAN0_base> + 0370_H + (04_H × y),
RSCAN0TMTCSTSyH: <RSCAN0_base> + 0372_H + (04_H × y)

RSCAN0TMTCSTSyLL: <RSCAN0_base> + 0370_H + (04_H × y),
RSCAN0TMTCSTSyLH: <RSCAN0_base> + 0371_H + (04_H × y),
RSCAN0TMTCSTSyHL: <RSCAN0_base> + 0372_H + (04_H × y),
RSCAN0TMTCSTSyHH: <RSCAN0_base> + 0373_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTCSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTCSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.67 RSCAN0TMTCSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission has not been completed. 1: Transmission has been completed.
15 to 0	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission has not been completed. 1: Transmission has been completed.

TMTCSTSp Flags (p = 0 to 95)

When the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)), the corresponding TMTCSTSp flag is set to 1.

To clear the TMTCSTSp flag to 0, set the corresponding TMTRF[1:0] flags to 00_B. These flags are cleared to 0 in channel reset mode.

Table 20.68 shows the bit assignment.

Table 20.68 TMTCSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15
48	3	0
. .	. .	. .
62	3	14
63	3	15
64	4	0
65	4	1
. .	. .	. .
78	4	14
79	4	15
80	5	0
81	5	1
. .	. .	. .
94	5	14
95	5	15

20.3.51 RSCAN0TMTASTSy — Transmit Buffer Transmit Abort Status Register (y = 0 to 2)

Access: RSCAN0TMTASTSy register is a read-only register that can be read in 32-bit units.
RSCAN0TMTASTSyL, RSCAN0TMTASTSyH registers are the read-only registers that can be read in 16-bit units.
RSCAN0TMTASTSyLL, RSCAN0TMTASTSyLH, RSCAN0TMTASTSyHL, RSCAN0TMTASTSyHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0TMTASTSy: <RSCAN0_base> + 0380_H + (04_H × y)

RSCAN0TMTASTSyL: <RSCAN0_base> + 0380_H + (04_H × y),
RSCAN0TMTASTSyH: <RSCAN0_base> + 0382_H + (04_H × y)

RSCAN0TMTASTSyLL: <RSCAN0_base> + 0380_H + (04_H × y),
RSCAN0TMTASTSyLH: <RSCAN0_base> + 0381_H + (04_H × y),
RSCAN0TMTASTSyHL: <RSCAN0_base> + 0382_H + (04_H × y),
RSCAN0TMTASTSyHH: <RSCAN0_base> + 0383_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTASTSp (p = y × 32 + 31 to y × 32 × 16 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTASTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.69 RSCAN0TMTASTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission is not aborted 1: Transmission is aborted
15 to 0	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission is not aborted. 1: Transmission is aborted.

TMTASTSp Flags (p = 0 to 95)

When the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to 01_B (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

To clear the TMTASTSp flag to 0, set the corresponding TMTRF[1:0] flags to 00_B. These flags are cleared to 0 in channel reset mode.

Table 20.70 shows the bit assignment.

Table 20.70 TMTASTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15
48	3	0
. .	. .	. .
62	3	14
63	3	15
64	4	0
65	4	1
. .	. .	. .
78	4	14
79	4	15
80	5	0
81	5	1
. .	. .	. .
94	5	14
95	5	15

20.3.52 RSCAN0TMIECy — Transmit Buffer Interrupt Enable Configuration Register (y = 0 to 2)

Access: RSCAN0TMIECy register can be read or written in 32-bit units.
RSCAN0TMIECyL, RSCAN0TMIECyH registers can be read or written in 16-bit units.
RSCAN0TMIECyLL, RSCAN0TMIECyLH, RSCAN0TMIECyHL, RSCAN0TMIECyHH registers can be read or written in 8-bit units.

Address: RSCAN0TMIECy: <RSCAN0_base> + 0390_H + (04_H × y)

RSCAN0TMIECyL: <RSCAN0_base> + 0390_H + (04_H × y),
RSCAN0TMIECyH: <RSCAN0_base> + 0392_H + (04_H × y)

RSCAN0TMIECyLL: <RSCAN0_base> + 0390_H + (04_H × y),
RSCAN0TMIECyLH: <RSCAN0_base> + 0391_H + (04_H × y),
RSCAN0TMIECyHL: <RSCAN0_base> + 0392_H + (04_H × y),
RSCAN0TMIECyHH: <RSCAN0_base> + 0393_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIEp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMIEp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1, 2))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.71 RSCAN0TMIECy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMIEp	Transmit Buffer Interrupt Enable p (p = y × 32 + 31 to y × 32 + 16) 0: Transmit buffer interrupt is disabled 1: Transmit buffer interrupt is enabled
15 to 0	TMIEp	Transmit Buffer Interrupt Enable p (p = y × 32 + 15 to y × 32 + 0) 0: Transmit buffer interrupt is disabled. 1: Transmit buffer interrupt is enabled.

TMIEp Bits (p = 0 to 95)

When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify these bits when the TMTRM flag in the corresponding RSCAN0TMSTSp register is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers or transmit buffers allocated to the transmit queue.

Table 20.72 shows the bit assignment.

Table 20.72 TMIEp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15
48	3	0
. .	. .	. .
62	3	14
63	3	15
64	4	0
65	4	1
. .	. .	. .
78	4	14
79	4	15
80	5	0
81	5	1
. .	. .	. .
94	5	14
95	5	15

20.3.53 RSCAN0TMIDp — Transmit Buffer ID Register (p = 0 to 95)

Access: RSCAN0TMIDp register can be read or written in 32-bit units.
RSCAN0TMIDpL, RSCAN0TMIDpH registers can be read or written in 16-bit units.
RSCAN0TMIDpLL, RSCAN0TMIDpLH, RSCAN0TMIDpHL, RSCAN0TMIDpHH registers can be read or written in 8-bit units.

Address: RSCAN0TMIDp: <RSCAN0_base> + 1000_H + (10_H × p)

RSCAN0TMIDpL: <RSCAN0_base> + 1000_H + (10_H × p),
RSCAN0TMIDpH: <RSCAN0_base> + 1002_H + (10_H × p)

RSCAN0TMIDpLL: <RSCAN0_base> + 1000_H + (10_H × p),
RSCAN0TMIDpLH: <RSCAN0_base> + 1001_H + (10_H × p),
RSCAN0TMIDpHL: <RSCAN0_base> + 1002_H + (10_H × p),
RSCAN0TMIDpHH: <RSCAN0_base> + 1003_H + (10_H × p)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.73 RSCAN0TMIDp Register Contents

Bit Position	Bit Name	Function
31	TMIDE	Transmit Buffer IDE 0: Standard ID 1: Extended ID
30	TMRTR	Transmit Buffer RTR 0: Data frame 1: Remote frame
29	THLEN	Transmit History Data Store Enable 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	TMID[28:0]	Transmit Buffer ID Data Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write data to this register. If this register is allocated to the transmit queue, only write data to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMIDE Bit

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

TMRTR Bit

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

THLEN Bit

With this bit set to 1, the transmit history data of the message transmitted (label information and the number and type) are stored in the transmit history buffer after transmission is completed.

TMID[28:0] Bits

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

20.3.54 RSCAN0TMPTRp — Transmit Buffer Pointer Register (p= 0 to 95)

Access: RSCAN0TMPTRp register can be read or written in 32-bit units.
RSCAN0TMPTRpH register can be read or written in 16-bit units.
RSCAN0TMPTRpHL, RSCAN0TMPTRpHH registers can be read or written in 8-bit units.

Address: RSCAN0TMPTRp: <RSCAN0_base> + 1004_H + (10_H × p)

RSCAN0TMPTRpH: <RSCAN0_base> + 1006_H + (10_H × p)

RSCAN0TMPTRpHL: <RSCAN0_base> + 1006_H + (10_H × p),
RSCAN0TMPTRpHH: <RSCAN0_base> + 1007_H + (10_H × p)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDLC[3:0]				—	—	—	—	TMPTR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.74 RSCAN0TMPTRp Register Contents

Bit Position	Bit Name	Function
31 to 28	TMDLC[3:0]	Transmit Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 24	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
23 to 16	TMPTR[7:0]	Transmit Buffer Label Data Set the label information to be stored in the transmit history buffer.
15 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMDLC[3:0] Bits

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the RSCAN0TMIDp register is set to 0 (data frame). If the data length is set to 9 bytes or more, the transmit data is 8 bytes long.

When the TMRTR bit is set to 1 (remote frame), set the data length of messages to be requested.

TMPTR[7:0] Bits

When message transmission has been completed, the TMPTR[7:0] bit value is stored in the transmit history buffer.

20.3.55 RSCAN0TMDF0p — Transmit Buffer Data Field 0 Register (p = 0 to 95)

Access: RSCAN0TMDF0p register can be read or written in 32-bit units.
RSCAN0TMDF0pL, RSCAN0TMDF0pH registers can be read or written in 16-bit units.
RSCAN0TMDF0pLL, RSCAN0TMDF0pLH, RSCAN0TMDF0pHL, RSCAN0TMDF0pHH registers can be read or written in 8-bit units.

Address: RSCAN0TMDF0p: <RSCAN0_base> + 1008_H + (10_H × p)

RSCAN0TMDF0pL: <RSCAN0_base> + 1008_H + (10_H × p),
RSCAN0TMDF0pH: <RSCAN0_base> + $100A_H$ + (10_H × p)

RSCAN0TMDF0pLL: <RSCAN0_base> + 1008_H + (10_H × p),
RSCAN0TMDF0pLH: <RSCAN0_base> + 1009_H + (10_H × p),
RSCAN0TMDF0pHL: <RSCAN0_base> + $100A_H$ + (10_H × p),
RSCAN0TMDF0pHH: <RSCAN0_base> + $100B_H$ + (10_H × p)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB3[7:0]								TMDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB1[7:0]								TMDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.75 RSCAN0TMDF0p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB3[7:0]	Transmit Buffer Data Byte 3
23 to 16	TMDB2[7:0]	Transmit Buffer Data Byte 2
15 to 8	TMDB1[7:0]	Transmit Buffer Data Byte 1
7 to 0	TMDB0[7:0]	Transmit Buffer Data Byte 0 Set the transmit buffer data.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

20.3.56 RSCAN0TMDF1p — Transmit Buffer Data Field 1 Register (p = 0 to 95)

Access: RSCAN0TMDF1p register can be read or written in 32-bit units.
RSCAN0TMDF1pL, RSCAN0TMDF1pH registers can be read or written in 16-bit units.
RSCAN0TMDF1pLL, RSCAN0TMDF1pLH, RSCAN0TMDF1pHL, RSCAN0TMDF1pHH registers can be read or written in 8-bit units.

Address: RSCAN0TMDF1p: <RSCAN0_base> + $100C_H$ + (10_H × p)

RSCAN0TMDF1pL: <RSCAN0_base> + $100C_H$ + (10_H × p),
RSCAN0TMDF1pH: <RSCAN0_base> + $100E_H$ + (10_H × p)

RSCAN0TMDF1pLL: <RSCAN0_base> + $100C_H$ + (10_H × p),
RSCAN0TMDF1pLH: <RSCAN0_base> + $100D_H$ + (10_H × p),
RSCAN0TMDF1pHL: <RSCAN0_base> + $100E_H$ + (10_H × p),
RSCAN0TMDF1pHH: <RSCAN0_base> + $100F_H$ + (10_H × p)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB7[7:0]								TMDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB5[7:0]								TMDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.76 RSCAN0TMDF1p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB7[7:0]	Transmit Buffer Data Byte 7
23 to 16	TMDB6[7:0]	Transmit Buffer Data Byte 6
15 to 8	TMDB5[7:0]	Transmit Buffer Data Byte 5
7 to 0	TMDB4[7:0]	Transmit Buffer Data Byte 4 Set the transmit buffer data.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

20.3.57 RSCAN0TXQCCm — Transmit Queue Configuration and Control Register (m = 0 to 5)

Access: RSCAN0TXQCCm register can be read or written in 32-bit units.
RSCAN0TXQCCmL register can be read or written in 16-bit units.
RSCAN0TXQCCmLL, RSCAN0TXQCCmLH registers can be read or written in 8-bit units.

Address: RSCAN0TXQCCm: <RSCAN0_base> + $03A0_H$ + (04_H × m)

RSCAN0TXQCCmL: <RSCAN0_base> + $03A0_H$ + (04_H × m)

RSCAN0TXQCCmLL: <RSCAN0_base> + $03A0_H$ + (04_H × m),
RSCAN0TXQCCmLH: <RSCAN0_base> + $03A1_H$ + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TXQIM	TXQIE	TXQDC[3:0]				—	—	—	—	—	—	—	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 20.77 RSCAN0TXQCCm Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13	TXQIM	Transmit Queue Interrupt Source Select 0: When the transmit queue becomes empty upon completion of message transmission, a transmit queue interrupt request is generated. 1: A transmit queue interrupt request is generated each time a message has been transmitted.
12	TXQIE	Transmit Queue Interrupt Enable 0: Transmit queue interrupt is disabled. 1: Transmit queue interrupt is enabled.
11 to 8	TXQDC[3:0]	Transmit Queue Depth Configuration Setting these bits to g (g = 2 to 15) makes the (g + 1)-buffer transmit queue available. Setting these bits to 0 disables the transmit queue. Setting these bits to 1 is prohibited.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TXQE	Transmit Queue Enable 0: The transmit queue is not used. 1: The transmit queue is used.

TXQIM Bit

This bit is used to select a transmit queue interrupt source. Modify this bit in channel reset mode.

TXQIE Bit

When the TXQIE bit is set to 1 and the source selected by the TXQIM bit occurs, an interrupt request is generated.

Set the TXQE bit to 0 before modifying the TXQIE bit.

TXQDC[3:0] Bits

These bits are used to specify the number of transmit buffers to be allocated to the transmit queues. Transmit buffers are allocated to transmit queues in descending order of buffer number, that is, from (m × 16 + 15) to (m × 16 + 0). For examples of how buffer allocation is done, see **Figure 20.9**. Modify these bits only in channel reset mode.

TXQE Bit

Setting this bit to 1 makes the transmit queue available. Modify this bit in channel communication mode or channel halt mode. This bit is cleared to 0 in channel reset mode.

Before setting the TXQE bit to 1, set the TXQDC[3:0] bit value to 0010_B or more.

20.3.58 RSCAN0TXQSTSm — Transmit Queue Status Register (m = 0 to 5)

Access: RSCAN0TXQSTSm register can be read or written in 32-bit units.
RSCAN0TXQSTSmL register can be read or written in 16-bit units.
RSCAN0TXQSTSmLL register can be read or written in 8-bit units.

Address: RSCAN0TXQSTSm: <RSCAN0_base> + $03C0_H$ + (04_H × m)

RSCAN0TXQSTSmL: <RSCAN0_base> + $03C0_H$ + (04_H × m)

RSCAN0TXQSTSmLL: <RSCAN0_base> + $03C0_H$ + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQIF	TXQFLL	TXQEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.78 RSCAN0TXQSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	Reserved	When read, the undefined value is returned. When writing, write “0”
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	TXQIF	Transmit Queue Interrupt Request Flag 0: No transmit queue interrupt request is present. 1: A transmit queue interrupt request is present.
1	TXQFLL	Transmit Queue Full Status Flag 0: The transmit queue is not full. 1: The transmit queue is full.
0	TXQEMP	Transmit Queue Empty Status Flag 0: The transmit queue contains messages. 1: The transmit queue contains no message (transmit queue empty).

TXQIF Flag

The TXQIF flag is set to 1 when the event specified by the TXQIM bit in the RSCAN0TXQCCm register has occurred.

The TXQIF flag is cleared to 0 in channel reset mode or by writing 0 to this flag. This flag is not cleared to 0 by setting the TXQE bit in the RSCAN0TXQCCm register to 0 (the transmit queue is not used).

TXQFLL Flag

The TXQFLL flag is set to 1 when the number of messages set for the transmit queue matches the transmit queue depth set by the TXQDC[3:0] bits in the RSCAN0TXQCCm register.

This flag is cleared to 0 in any of the following cases.

- The number of messages set for the transmit queue is smaller than the transmit queue depth set by the TXQDC[3:0] bits.
- In channel reset mode

TXQEMP Flag

The TXQEMP flag is cleared to 0 when even a single message is set for the transmit queue.

This flag is set to 1 in any of the following cases.

- The TXQE bit is set to 0 (the transmit queue is not used).
- The transmit queue becomes empty.
- In channel reset mode

20.3.59 RSCAN0TXQPCTRm — Transmit Queue Pointer Control Register (m = 0 to 5)

Access: RSCAN0TXQPCTRm register is a write-only register that can be written in 32-bit units.
RSCAN0TXQPCTRmL register is a write-only register that can be written in 16-bit units.
RSCAN0TXQPCTRmLL register is a write-only register that can be written in 8-bit units.

Address: RSCAN0TXQPCTRm: <RSCAN0_base> + $03E0_H$ + (04_H × m)

RSCAN0TXQPCTRmL: <RSCAN0_base> + $03E0_H$ + (04_H × m)

RSCAN0TXQPCTRmLL: <RSCAN0_base> + $03E0_H$ + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.79 RSCAN0TXQPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	TXQPC[7:0]	Transmit Queue Pointer Control Writing FF_H to these bits moves the write pointer of the transmit queue to the next queue buffer.

TXQPC[7:0] Bits

Writing FF_H to the TXQPC[7:0] bits moves the write pointer to the next transmit queue buffer and generates a transmit request of the message. Write transmit messages to the RSCAN0TMIDp, RSCAN0TMPTRp, RSCAN0TMDF0p, and RSCAN0TMDF1p registers (p = 15, 31, 47, 63, 79, and 95) before writing FF_H to the TXQPC[7:0] bits.

When writing FF_H to these bits, make sure that the TXQE bit in the RSCAN0TXQCCm register is set to 1 (the transmit queue is used) and the TXQFLL flag in the RSCAN0TXQSTSm register is 0 (the transmit queue is not full).

20.3.60 RSCAN0THLCCm — Transmit History Configuration and Control Register (m = 0 to 5)

Access: RSCAN0THLCCm register can be read or written in 32-bit units.
RSCAN0THLCCmL register can be read or written in 16-bit units.
RSCAN0THLCCmLL, RSCAN0THLCCmLH registers can be read or written in 8-bit units.

Address: RSCAN0THLCCm: <RSCAN0_base> + 0400_H + (04_H × m)

RSCAN0THLCCmL: <RSCAN0_base> + 0400_H + (04_H × m)

RSCAN0THLCCmLL: <RSCAN0_base> + 0400_H + (04_H × m),
RSCAN0THLCCmLH: <RSCAN0_base> + 0401_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 20.80 RSCAN0THLCCm Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	THLDTE	Transmit History Target Buffer Select 0: Entry from transmit/receive FIFO buffers and transmit queue 1: Entry from transmit buffers, transmit/receive FIFO buffers, and transmit queue
9	THLIM	Transmit History Interrupt Source Select 0: When 12 sets of data have been stored in the transmit history buffer 1: When a single set of transmit history data has been stored
8	THLIE	Transmit History Interrupt Enable 0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled.
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	THLE	Transmit History Buffer Enable 0: Transmit history buffer is not used. 1: Transmit history buffer is used.

THLDTE Bit

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers and the transmit queue is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers, transmit/receive FIFO buffers, and the transmit queue is stored in the transmit history buffer.

Modify this bit only in channel reset mode.

THLIM Bit

This bit is used to select a transmit history interrupt source.

Modify this bit only in channel reset mode.

THLIE Bit

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit only when the THLE bit set to 0.

THLE Bit

Setting this bit to 1 makes the transmit history buffer available. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer.

Modify this bit in channel communication mode or channel halt mode.

20.3.61 RSCAN0THLSTSm — Transmit History Status Register (m = 0 to 5)

Access: RSCAN0THLSTSm register can be read or written in 32-bit units.
RSCAN0THLSTSmL register can be read or written in 16-bit units.
RSCAN0THLSTSmLL register can be read or written in 8-bit units.
RSCAN0THLSTSmLH register is a read-only register that can be read in 8-bit units.

Address: RSCAN0THLSTSm: <RSCAN0_base> + 0420_H + (04_H × m)

RSCAN0THLSTSmL: <RSCAN0_base> + 0420_H + (04_H × m)

RSCAN0THLSTSmLL: <RSCAN0_base> + 0420_H + (04_H × m),
RSCAN0THLSTSmLH: <RSCAN0_base> + 0421_H + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THLMC[4:0]					—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 20.81 RSCAN0THLSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	THLMC[4:0]	Transmit History Buffer Unread Data Counter These bits indicate the number of unread data sets stored in the transmit history buffer.
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	THLIF	Transmit History Interrupt Request Flag 0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present.
2	THLELT	Transmit History Buffer Overflow Flag 0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred.
1	THLFLL	Transmit history Buffer Full Status Flag 0: Transmit history buffer is not full. 1: Transmit history buffer is full.
0	THLEMP	Transmit History Buffer Empty Status Flag 0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty).

THLMC[4:0] Bits

These bits indicate the number of unread data sets stored in the transmit history buffer.

THLIF Flag

The THLIF flag is set to 1 when the interrupt source specified with the THLIM bit in the RSCAN0THLCCm register occurs.

This flag is cleared to 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLELT Flag

The THLELT flag is set to 1 when an attempt is made to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLFLL Flag

The THLFLL flag is set to 1 when 16 data sets have been stored in the transmit history buffer, and is cleared to 0 when the number of data sets stored in the transmit history buffer has decreased to less than 16. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the RSCAN0THLCCm register is set to 0 (transmit history buffer is not used).

THLEMP Flag

The THLEMP flag is cleared to 0 when even a single set of transmit history data has been stored in the transmit history buffer.

This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the RSCAN0THLCCm register is set to 0 (transmit history buffer is not used).

NOTE

To clear THLIF or THLELT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

20.3.62 RSCAN0THLACCm — Transmit History Access Register (m = 0 to 5)

Access: RSCAN0THLACCm register is a read-only register that can be read in 32-bit units.
RSCAN0THLACCmL register is a read-only register that can be read in 16-bit units.
RSCAN0THLACCmLL, RSCAN0THLACCmLH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0THLACCm: <RSCAN0_base> + 1800_H + (04_H × m)

RSCAN0THLACCmL: <RSCAN0_base> + 1800_H + (04_H × m)

RSCAN0THLACCmLL: <RSCAN0_base> + 1800_H + (04_H × m),
RSCAN0THLACCmLH: <RSCAN0_base> + 1801_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TID[7:0]								—	BN[3:0]				BT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.82 RSCAN0THLACCm Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 8	TID[7:0]	Label Data The label information of stored data can be read.
7	Reserved	When read, the value after reset is returned.
6 to 3	BN[3:0]	Buffer Number Data The buffer number of transmit source (transmit buffer, transmit/receive FIFO or transmit queue) can be read.
2 to 0	BT[2:0]	Buffer Type Data b2 b1 b0 0 0 1: Transmit buffer 0 1 0: Transmit/receive FIFO buffer 1 0 0: Transmit queue

TID[7:0] Bits

These bits indicate the label information of transmit history data stored in the transmit history buffer.

BN[3:0] Bits

These bits indicate the transmit source buffer number in the transmit history data stored in the transmit history buffer.

BT[2:0] Bits

These bits indicate the type of the transmit source buffer in the transmit history data stored in the transmit history buffer.

20.3.63 RSCAN0THLPCTRm — Transmit History Pointer Control Register (m = 0 to 5)

Access: RSCAN0THLPCTRm register is a write-only register that can be written in 32-bit units.
RSCAN0THLPCTRmL register is a write-only register that can be written in 16-bit units.
RSCAN0THLPCTRmLL register is a write-only register that can be written in 8-bit units.

Address: RSCAN0THLPCTRm: <RSCAN0_base> + 0440_H + (04_H × m)

RSCAN0THLPCTRmL: <RSCAN0_base> + 0440_H + (04_H × m)

RSCAN0THLPCTRmLL: <RSCAN0_base> + 0440_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 20.83 RSCAN0THLPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing, write the value after reset.
7 to 0	THLPC[7:0]	Transmit History List Pointer Control Writing FF_H to these bits moves the read pointer to the next unread data in the transmit history buffer.

THLPC[7:0] Bits

When the THLPC[7:0] bits are set to FF_H, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[4:0] (transmit history buffer unread data counter) value in the RSCAN0THLSTSm register is decremented. Write FF_H to the THLPC[7:0] bits after reading from the RSCAN0THLACCm register.

When writing FF_H to these bits, make sure that the THLE bit in the RSCAN0THLCCm register is set to 1 (transmit history buffer is used) and the THLEMP flag in the RSCAN0THLSTSm register is 0.

20.3.64 RSCAN0GTSTCFG — Global Test Configuration Register

Access: RSCAN0GTSTCFG register can be read or written in 32-bit units.
RSCAN0GTSTCFGL, RSCAN0GTSTCFGH registers can be read or written in 16-bit units.
RSCAN0GTSTCFGLL, RSCAN0GTSTCFGHL registers can be read or written in 8-bit units.

Address: RSCAN0GTSTCFG: <RSCAN0_base> + 0468_H

RSCAN0GTSTCFGL: <RSCAN0_base> + 0468_H, RSCAN0GTSTCFGH: <RSCAN0_base> + $046A_H$

RSCAN0GTSTCFGLL: <RSCAN0_base> + 0468_H, RSCAN0GTSTCFGHL: <RSCAN0_base> + $046A_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	RTMPS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	C5ICBCE	C4ICBCE	C3ICBCE	C2ICBCE	C1ICBCE	C0ICBCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.84 RSCAN0GTSTCFG Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	RTMPS[6:0]	RAM Test Page Configuration Set a value within a range of page 0 (00_H) to page 56 (38_H).
15 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	C5ICBCE	CAN5 Inter-channel Communication Test Enable 0: CAN5 inter-channel communication test is disabled 1: CAN5 inter-channel communication test is enabled.
4	C4ICBCE	CAN4 Inter-channel Communication Test Enable 0: CAN4 inter-channel communication test is disabled 1: CAN4 inter-channel communication test is enabled.
3	C3ICBCE	CAN3 Inter-channel Communication Test Enable 0: CAN3 inter-channel communication test is disabled 1: CAN3 inter-channel communication test is enabled.
2	C2ICBCE	CAN2 Inter-channel Communication Test Enable 0: CAN2 inter-channel communication test is disabled 1: CAN2 inter-channel communication test is enabled.
1	C1ICBCE	CAN1 Inter-Channel Communication Test Enable 0: CAN1 inter-channel communication test is disabled. 1: CAN1 inter-channel communication test is enabled.
0	C0ICBCE	CAN0 Inter-Channel Communication Test Enable 0: CAN0 inter-channel communication test is disabled. 1: CAN0 inter-channel communication test is enabled.

Modify the RSCAN0GTSTCFG register only in global test mode.

RTMPS[6:0] Bits

These bits are used to set the RAM test target page number for RAM test. Set a value in the range of 00_H to 38_H, inclusive.

C5ICBCE Bit

Setting this bit to 1 enables the channel 5 inter-channel communication test.

C4ICBCE Bit

Setting this bit to 1 enables the channel 4 inter-channel communication test.

C3ICBCE Bit

Setting this bit to 1 enables the channel 3 inter-channel communication test.

C2ICBCE Bit

Setting this bit to 1 enables the channel 2 inter-channel communication test.

C1ICBCE Bit

Setting this bit to 1 enables the channel 1 inter-channel communication test.

C0ICBCE Bit

Setting this bit to 1 enables the channel 0 inter-channel communication test.

20.3.65 RSCAN0GTSTCTR — Global Test Control Register

Access: RSCAN0GTSTCTR register can be read or written in 32-bit units.
RSCAN0GTSTCTRL register can be read or written in 16-bit units.
RSCAN0GTSTCTRLL register can be read or written in 8-bit units.

Address: RSCAN0GTSTCTR: <RSCAN0_base> + $046C_H$

RSCAN0GTSTCTRL: <RSCAN0_base> + $046C_H$

RSCAN0GTSTCTRLL: <RSCAN0_base> + $046C_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBCTME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 20.85 RSCAN0GTSTCTR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	RTME	RAM Test Enable 0: RAM test is disabled. 1: RAM test is enabled.
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ICBCTME	Communication Test between Channels Enable 0: Communication test between channels disabled 1: Communication test between channels enabled

RTME Bit

Setting this bit to 1 enables the RAM test. Modify this bit only in global test mode.

1. Set the GMDC[1:0] bits in the RSCAN0GCTR register to 10_B (Global test mode).
2. Set the RTME bit to 1.
3. Check that the RTME bit is set to 1.

ICBCTME Bit

When this bit is set to 1, a communication test is enabled between the channels for which the CmICBCE bit (m = 0 to 5) in the RSCAN0GTSTCFG register has been set to 1. Modify the ICBCTME bit only in global test mode.

20.3.66 RSCAN0GLOCKK — Global Lock Key Register

Access: RSCAN0GLOCKK register is a write-only register that can be written in 32-bit units.
RSCAN0GLOCKKL register is a write-only register that can be written in 16-bit units.

Address: RSCAN0GLOCKK: <RSCAN0_base> + $047C_H$

RSCAN0GLOCKKL: <RSCAN0_base> + $047C_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1

Note 1. Writing to these bits is effective only when the RS-CAN module is in global test mode.

Table 20.86 RSCAN0GLOCKK Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing, write the value after reset.
15 to 0	LOCK[15:0]	Lock Key These bits are key bits to release protection of test mode.

The RSCAN0GLOCKK register releases protection of special test bits and is write only.

For the protection release data, see **Section 20.10.4.2, Procedure for Releasing the Protection**.

LOCK[15:0] Bits

Writing the protection release data to the LOCK[15:0] bits in succession enables writing 1 to the RTME bit in the RSCAN0GTSTCTR register.

After the protection has been released, writing to the I/O register area (<RSCAN0_base> + 0000_H to <RSCAN0_base> + $04FF_H$) of the CAN (except the RAM) enables the protection again.

Reading from the I/O register area of the CAN or reading from/writing to other areas does not enable the protection.

20.3.67 RSCAN0RPGACCr — RAM Test Page Access Register (r = 0 to 63)

Access: RSCAN0RPGACCr register can be read or written in 32-bit units.
RSCAN0RPGACCrL, RSCAN0RPGACCrH registers can be read or written in 16-bit units.
RSCAN0RPGACCrLL, RSCAN0RPGACCrLH, RSCAN0RPGACCrHL, RSCAN0RPGACCrHH registers can be read or written in 8-bit units.

Address: RSCAN0RPGACCr: <RSCAN0_base> + 1900_H + (04_H × r)

RSCAN0RPGACCrL: <RSCAN0_base> + 1900_H + (04_H × r),
RSCAN0RPGACCrH: <RSCAN0_base> + 1902_H + (04_H × r)

RSCAN0RPGACCrLL: <RSCAN0_base> + 1900_H + (04_H × r),
RSCAN0RPGACCrLH: <RSCAN0_base> + 1901_H + (04_H × r),
RSCAN0RPGACCrHL: <RSCAN0_base> + 1902_H + (04_H × r),
RSCAN0RPGACCrHH: <RSCAN0_base> + 1903_H + (04_H × r)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.87 RSCAN0RPGACCr Register Contents

Bit Position	Bit Name	Function
31 to 0	RDTA[31:0]	RAM Data Test Access Data can be read and written in RSCAN RAM.

Modify the RSCAN0RPGACCr register in global test mode with the RTME bit in the RSCAN0GTSTCTR register set to 1 (RAM test is enabled).

The RSCAN0RPGACCr register is readable and writable when the RTME bit is set to 1.

20.4 Interrupt Sources

The RS-CAN module has 20 interrupts that are grouped into global interrupts and channel interrupts.

- Global interrupts (2 sources):
  Receive FIFO interrupt
  Global error interrupt
- Channel interrupts (3 sources/channel):
  CANm transmit interrupt (m = 0 to 5)
  - CANm transmit complete interrupt
  - CANm transmit abort interrupt
  - CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode)
  - CANm transmit history interrupt
  - CANm transmit queue Interrupt

  CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode)
  CANm error interrupt

When an interrupt request is generated, the corresponding interrupt request flag is set to 1 (interrupt request present). In that case, when the interrupt enable bit is set to 1 (enabling interrupts), an interrupt request is output from the RS-CAN module. (Generation of interrupts also depends on the interrupt control register settings of the interrupt controller.)

Setting the interrupt request flag to 0 (no interrupt request present) or setting the interrupt enable bit to 0 (disabling interrupts) clears the current interrupt request. The current interrupt request is still output until the interrupt request flag is cleared.

Table 20.88 lists the CAN interrupt sources. **Figure 20.2** shows the CAN global interrupt block diagram. **Figure 20.3** shows the CAN channel interrupt block diagram.

Table 20.88 List of CAN Interrupt Sources

Interrupt Source			Corresponding Interrupt Request Flag	Corresponding Interrupt Enable Bit
Global interrupts	Receive FIFO	Receive FIFO 0	RFIF flag in the RSCAN0RFSTS0 register	RFIE bit in the RSCAN0RFCC0 register
		Receive FIFO 1	RFIF flag in the RSCAN0RFSTS1 register	RFIE bit in the RSCAN0RFCC1 register
		Receive FIFO 2	RFIF flag in the RSCAN0RFSTS2 register	RFIE bit in the RSCAN0RFCC2 register
		Receive FIFO 3	RFIF flag in the RSCAN0RFSTS3 register	RFIE bit in the RSCAN0RFCC3 register
		Receive FIFO 4	RFIF flag in the RSCAN0RFSTS4 register	RFIE bit in the RSCAN0RFCC4 register
		Receive FIFO 5	RFIF flag in the RSCAN0RFSTS5 register	RFIE bit in the RSCAN0RFCC5 register
		Receive FIFO 6	RFIF flag in the RSCAN0RFSTS6 register	RFIE bit in the RSCAN0RFCC6 register
		Receive FIFO 7	RFIF flag in the RSCAN0RFSTS7 register	RFIE bit in the RSCAN0RFCC7 register
	Global error		• DEF flag in the RSCAN0GERFL register • MES flag in the RSCAN0GERFL register • THLES flag in the RSCAN0GERFL register	• DEIE bit in the RSCAN0GCTR register • MEIE bit in the RSCAN0GCTR register • THLEIE bit in the RSCAN0GCTR register
Channel interrupts (m = 0 to 5)	CANm transmit	CANm transmit complete	TMTRF[1:0] flag in the RSCAN0TMSTSp register	TMIE bit in the RSCAN0TMIECy register
		CANm transmit abort	TMTRF[1:0] flag in the RSCAN0TMSTSp register	TAIE bit in the RSCAN0CmCTR register
		CANm transmit/receive FIFO transmit complete	CFTXIF flag in the RSCAN0CFSTSk register	CFTXIE bit in the RSCAN0CFCCk register
		CANm transmit queue	TXQIF flag in the RSCAN0TXQSTSm register	TXQIE bit in the RSCAN0TXQCCm register
		CANm transmit history	THLIF flag in the RSCAN0THLSTSm register	THLIE bit in the RSCAN0THLCCm register
	CANm transmit/receive FIFO receive complete		CFRXIF flag in the RSCAN0CFSTSk register	CFRXIE bit in the RSCAN0CFCCk register
	CANm error		• BEF flag in the RSCAN0CmERFL register • ALF flag in the RSCAN0CmERFL register • BLF flag in the RSCAN0CmERFL register • OVLF flag in the RSCAN0CmERFL register • BORF flag in the RSCAN0CmERFL register • BOEF flag in the RSCAN0CmERFL register • EPF flag in the RSCAN0CmERFL register • EWF flag in the RSCAN0CmERFL register	• BEIE bit in the RSCAN0CmCTR register • ALIE bit in the RSCAN0CmCTR register • BLIE bit in the RSCAN0CmCTR register • OLIE bit in the RSCAN0CmCTR register • BORIE bit in the RSCAN0CmCTR register • BOEIE bit in the RSCAN0CmCTR register • EPIE bit in the RSCAN0CmCTR register • EWIE bit in the RSCAN0CmCTR register

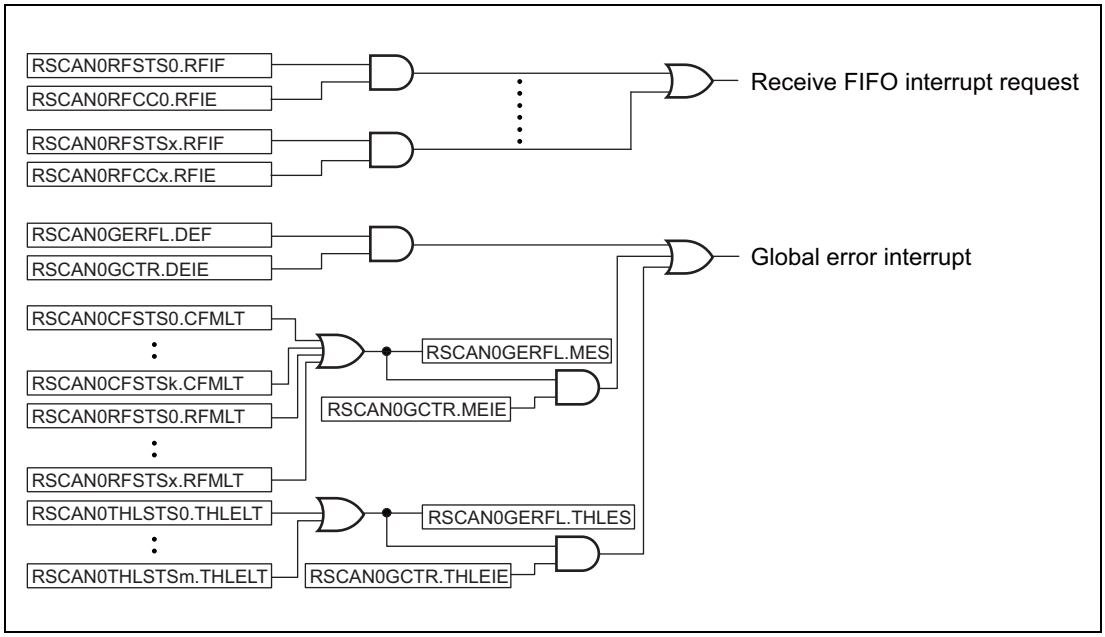


Figure 20.2 CAN Global Interrupt Block Diagram

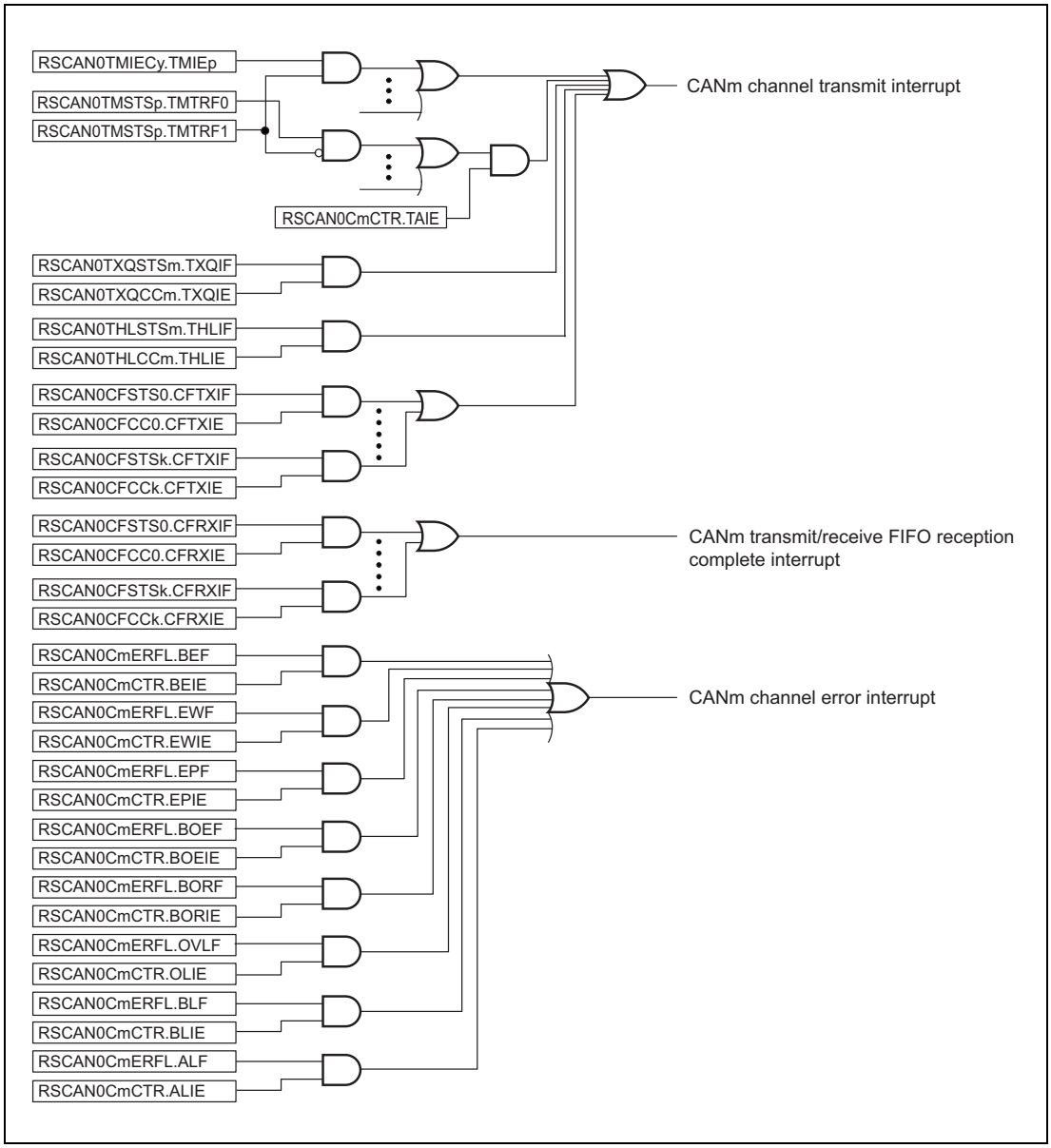


Figure 20.3 CAN Channel Interrupt Block Diagram

20.5 CAN Modes

The RS-CAN module has four global modes to control the entire RS-CAN module status and four channel modes to control individual channel status. Details of global modes are described in **Section 20.5.1, Global Modes**, and details of channel modes are described in **Section 20.5.2, Channel Modes**.

- Global stop mode: Stops the clocks of the entire module to achieve low power consumption.
- Global reset mode: Performs initial settings for the entire module.
- Global test mode: Performs test settings and performs the RAM test.
- Global operating mode: Makes the entire module operable.
- Channel stop mode: Stops the channel clock.
- Channel reset mode: Performs initial settings for the channels.
- Channel halt mode: Stops CAN communication and allows channel testing.
- Channel communication mode: Performs CAN communication.

20.5.1 Global Modes

Figure 20.4 shows the transitions of global modes.

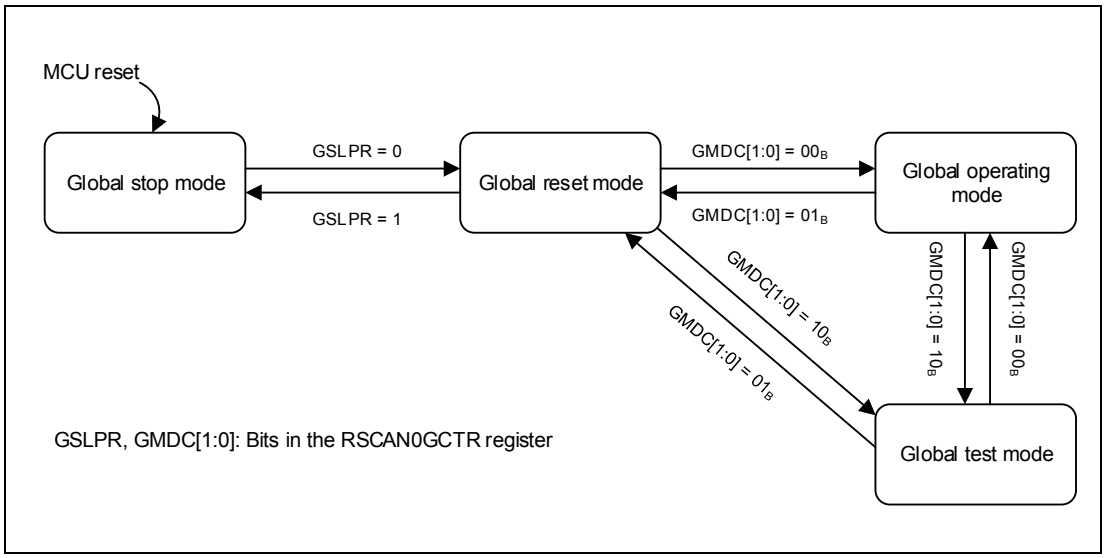


Figure 20.4 Transitions of Global Modes

In some cases, global mode transitions also force channel mode transitions. **Table 20.89** shows the channel mode transitions depending on the global mode setting dictated by the GMDC[1:0] bits and the GSLPR bit.

Table 20.89 Transitions of Channel Modes Depending on Global Mode Setting (GMDC[1:0] and GSLPR Bits)

Channel Mode before Setting	Channel Mode after Setting			
	GMDC[1:0] = 00_B GSLPR = 0 (Global Operation)	**GMDC[1:0] = 10_B GSLPR = 0 (Global Test)**	**GMDC[1:0] = 01_B GSLPR = 0 (Global Reset)**	**GMDC[1:0] = 01_B GSLPR = 1 (Global Stop)**
Channel communication	Channel communication	Channel halt	Channel reset	Transition prohibited
Channel halt	Channel halt	Channel halt	Channel reset	Transition prohibited
Channel reset	Channel reset	Channel reset	Channel reset	Channel stop
Channel stop	Channel stop	Channel stop	Channel stop	Channel stop

Note: GMDC[1:0], GSLPR: Bits in the RSCAN0GCTR register

Table 20.90 shows the global mode transition time.

Table 20.90 Global Mode Transition Time

Mode before Transition	Mode after Transition	Maximum Transition Time
Global stop	Global reset	Three pclk cycles
Global reset	Global stop	Three pclk cycles
Global reset	Global test	Ten pclk cycles
Global reset	Global operating	Ten pclk cycles
Global test	Global reset	Two CANm bit times
Global test	Global operating	Three pclk cycles
Global operating	Global reset	Two CANm bit times
Global operating	Global test	Two CAN frames*1

Note 1. CAN frame time of the lowest communication speed of the channels in use

20.5.1.1 Global Stop Mode

In global stop mode, clocks of the CAN do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained. Only the clock used by the CPU for writing to the GSLPR bit runs in this mode.

After the MCU is reset, the CAN module transitions to global stop mode. Setting the GSLPR bit in the RSCAN0GCTR register to 1 (in global stop mode) in global reset mode sets the CSLPR bit in each of the RSCAN0CmCTR register to 1 (channel stop mode). Afterwards, if all channels are forced to transition to channel stop mode, the CAN module transitions to global stop mode. The GSLPR bit should not be modified in global operating mode or global test mode.

20.5.1.2 Global Reset Mode

In global reset mode, RS-CAN module settings are performed. When the RS-CAN module transitions to global reset mode, some registers are initialized. **Table 20.93** and **Table 20.94** list the registers to be initialized.

Setting the GMDC[1:0] bits in the RSCAN0GCTR register to 01_B sets the CHMDC[1:0] bits in each of the RSCAN0CmCTR registers (m = 0 to 5) to 01_B (channel reset mode). If all channels are forced to transition to channel reset mode, the CAN module transitions to global reset mode. Channels that are already in channel reset mode or channel stop mode do not transition (because the CHMDC[1:0] bits have already been set to 01_B).

20.5.1.3 Global Test Mode

In global test mode, settings for test-related registers are performed. When the CAN module transitions to global test mode, all CAN communications are disabled.

Setting the GMDC[1:0] bits in the RSCAN0GCTR register to 10_B sets the CHMDC[1:0] bits in each of the RSCAN0CmCTR register to 10_B (channel halt mode). If all channels are forced to transition to channel halt mode, the CAN module transitions to global test mode. Channels that are in channel stop mode, channel reset mode, or channel halt mode do not transition.

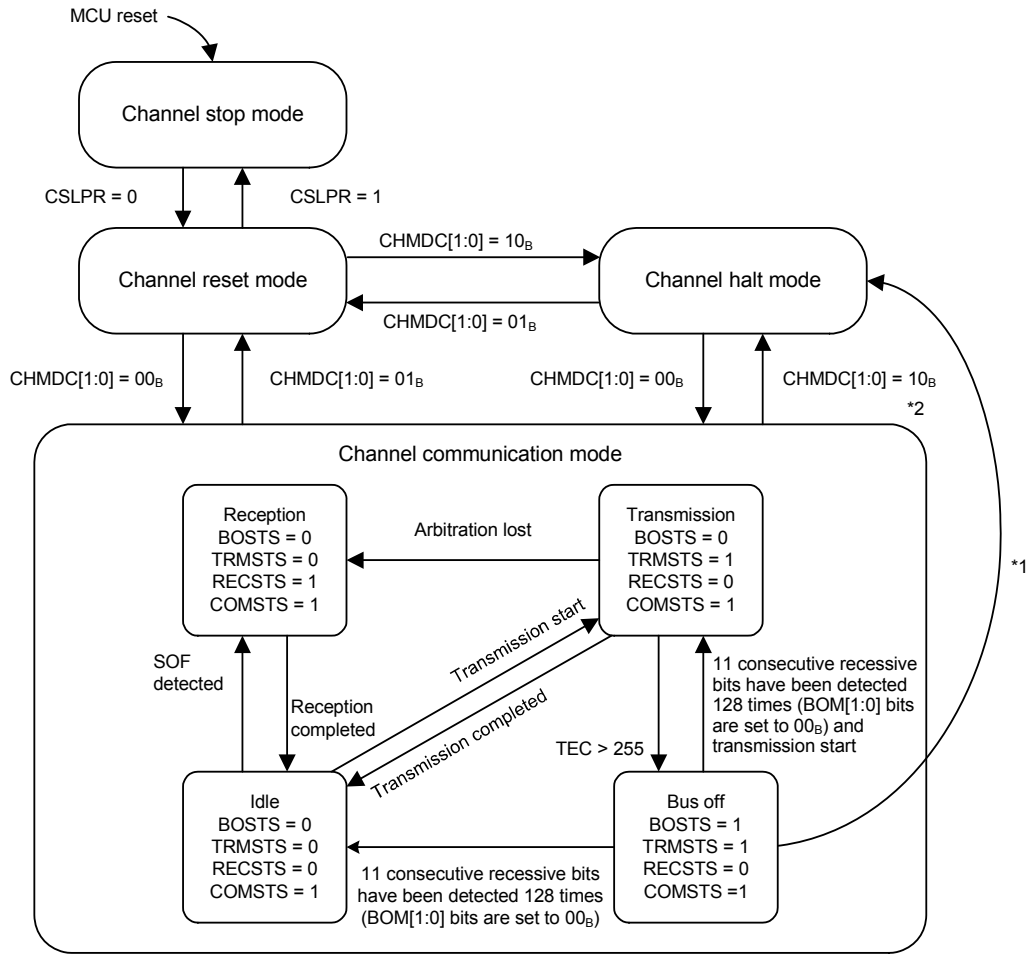
20.5.1.4 Global Operating Mode

The RS-CAN module operates in global operating mode.

When the GMDC[1:0] bits in the RSCAN0GCTR register are set to 00_B, the RS-CAN module transitions to global operating mode.

20.5.2 Channel Modes

Figure 20.5 shows a channel mode state transition chart. **Table 20.91** shows the channel mode transition time.



Note: CHMDC[1:0], CSLPR, BOM[1:0]: Bits in the RSCAN0CmCTR register (m = 0 to 5)
BOSTS, TRMSTS, RECSTS, COMSTS: Bits in the RSCAN0CmSTS register

Note 1. Timing of transition from bus off state to channel halt mode
- When BOM[1:0] = 01_B: Transition to channel halt mode when TEC exceeds 255
- When BOM[1:0] = 10_B: Transition to channel halt mode when 11 consecutive recessive bits have been detected 128 times
- When BOM[1:0] = 11_B: Transition to channel halt mode when the CHMDC[1:0] bits are set to 10_B

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode.

Figure 20.5 Channel Mode State Transition Chart

Table 20.91 Channel Mode Transition Time (1/2)

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel stop	Channel reset	Three pclk cycles
Channel reset	Channel stop	Three pclk cycles
Channel reset	Channel halt	Three CANm bit times
Channel reset	Channel communication	Three CANm bit times

Table 20.91 Channel Mode Transition Time (2/2)

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel halt	Channel reset	Two CANm bit times
Channel halt	Channel communication	Four CANm bit times
Channel communication	Channel reset	Two CANm bit times
Channel communication	Channel halt	Two CANm frames

20.5.2.1 Channel Stop Mode

In channel stop mode, clocks are not supplied to channels and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained.

Each channel enters channel stop mode after the MCU is reset. Channels also transition to channel stop mode when the CSLPR bit in the RSCAN0CmCTR register (m = 0 to 5) is set to 1 (channel stop mode) in channel reset mode. The CSLPR bit should not be modified in channel communication mode and channel halt mode.

20.5.2.2 Channel Reset Mode

In channel reset mode, channel settings are performed. When a channel transitions to channel reset mode, some channel-related registers are initialized. **Table 20.93** lists the registers to be initialized.

When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 01_B (channel reset mode) during CAN communication, communication is terminated before it is completed and the channel transitions to channel reset mode. **Table 20.92** shows the operation when the CHMDC[1:0] bits are set to 01_B (channel reset mode) during CAN communication.

20.5.2.3 Channel Halt Mode

In channel halt mode, settings for test-related registers of channels are performed. When a channel transitions to channel halt mode, CAN communication of the channel stops.

Table 20.92 shows operation when the CHMDC[1:0] bits are set to 10_B (channel halt mode) during CAN communication.

Table 20.92 Operation a Channel Transitions to Channel Reset Mode/Channel Halt Mode

Mode	During Reception	During Transmission	Bus Off State
Channel reset (CHMDC[1:0] = 01_B)	Transitions to channel reset mode before reception is completed.*1	Transitions to channel reset mode before transmission is completed.*1	Transitions to channel reset mode before bus off recovery.
Channel halt*3 (CHMDC[1:0] = 10_B)	Transitions to channel halt mode after reception is completed.*2	Transitions to channel halt mode after transmission is completed.	[When BOM[1:0] bits = 00_B] Transitions to channel halt mode (CHMDC[1:0] = 10_B) only after bus off recovery. [When BOM[1:0] bits = 01_B] Transitions to channel halt mode automatically when the condition for transition to bus off state is met. [When BOM[1:0] bits = 10_B] Transitions to channel halt mode automatically after bus off recovery. [When BOM[1:0] bits = 11_B] Transitions to channel halt mode immediately after the CHMDC[1:0] bits are set to 10_B before bus off recovery.

Note 1. To allow transition to channel reset mode after communication is completed, set the CHMDC[1:0] bits to

10_B and confirm that communication has been completed and transition to channel halt mode has been made, and then set the CHMDC[1:0] bits to 01_B.

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode. The CAN bus status can be confirmed with the BLF flag of the RSCAN0CmERFL register that becomes 1 when dominant lock is detected.

Note 3. When the transition from channel reset mode to channel halt mode is to be made, set the RSCAN0CmCFG register in channel reset mode and then shift to channel halt mode.

20.5.2.4 Channel Communication Mode

In channel communication mode, CAN communication is performed. Each channel has the following communication states during CAN communication.

- Idle: Neither reception nor transmission is in progress.
- Reception: Receiving a message sent from another node.
- Transmission: Transmitting a message.
- Bus off: Isolated from CAN communication.

When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 00_B, the channel transitions to channel communication mode. After that, once 11 consecutive recessive bits have been detected, the COMSTS flag in the RSCAN0CmSTS register (m = 0 to 5) is set to 1 (communication is ready) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

20.5.2.5 Bus Off State

A channel transitions to the bus off state according to the transmit/receive error counter increment/ decrement rules of the CAN specifications.

The conditions for returning from the bus off state are determined by the BOM[1:0] bits in the RSCAN0CmCTR register.

- When BOM[1:0] bits = 00_B:
  Bus off recovery is compliant with the CAN specifications. After 11 consecutive recessive bits have been detected 128 times, a channel returns from the bus off state to the CAN communication ready state (error active state). At that time, the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register are initialized to 00_H, the BORF flag in the RSCAN0CmERFL register is set to 1 (bus off recovery is detected), and a bus off recovery interrupt request is generated. When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 10_B (channel halt mode) in the bus off state, the channel transitions to channel halt mode after bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times).
- When BOM[1:0] bits = 01_B:
  When a channel transitions to the bus off state, the CHMDC[1:0] bits are set to 10_B and the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H. The BORF flag is not set to 1, and bus off recovery interrupt request is not generated.
- When BOM[1:0] bits = 10_B:
  When a channel has transitioned to the bus off state, the CHMDC[1:0] bits are set to 10_B. After bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times), the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H, the BORF flag is set to 1, and a bus off recovery interrupt request is generated.

- When BOM[1:0] bits = 11_B:
  When the CHMDC[1:0] bits are set to 10_B in the bus off state, the channel transitions to channel halt mode before bus off recovery is completed. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H, but the BORF flag is not set to 1. Also, a bus off recovery interrupt is not generated.
  However, the BORF flag becomes 1 and a bus off recovery interrupt request is generated if a CAN module transitions to error active state (by detecting 128 times of 11 consecutive recessive bits) before CHMDC[1:0] bits are set to 10_B.

If the RS-CAN module causes the channel to transition to channel halt mode simultaneously with a program write to the CHMDC[1:0] bits, the program write takes precedence. An automatic transition to channel halt mode when the BOM[1:0] bits are set to 01_B or 10_B is made only when the CHMDC[1:0] bits are 00_B (channel communication mode).

Furthermore, setting the RTBO bit in the RSCAN0CmCTR register to 1 allows a forced return from the bus off state. As soon as the RTBO bit is set to 1, the state changes to the error active state. After 11 consecutive recessive bits have been detected, the CAN module becomes ready for communication. In this case, the BORF flag is not set to 1 and the TEC[7:0] and REC[7:0] bits are initialized to 00_H. Write 1 to the RTBO bit only when the BOM[1:0] bits are set to 00_B. Writing the RTBO bit to 1 in a state other than the bus off state is ignored, and the RTBO bit is immediately set to 0.

Table 20.93 Registers Initialized in Global Reset Mode or Channel Reset Mode

Register	Bit / Flag
RSCAN0CmCTR register	CTMS[1:0], CTME, CHMDC[1:0]
RSCAN0CmSTS register	CHLTSTS, EPSTS, BOSTS, TRMSTS, RECSTS, COMSTS, REC[7:0], TEC[7:0]
RSCAN0CmERFL register	CRCREG[14:0], ADERR, B0ERR, B1ERR, CERR, AERR, FERR, SERR, ALF, BLF, OVLF, BORF, BOEF, EPF, EWF, BEF
RSCAN0CFCCk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFE
RSCAN0CFSTSk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFMC[7:0], CFFLL, CFEMP, CFMLT, CFRXIF, CFTXIF
RSCAN0CFTISTS register	CFkTXIF
RSCAN0TMCp register	TMOM, TMTAR, TMTR
RSCAN0TMSTSp register	TMTARM, TMTRM, TMTRF[1:0], TMTSTS
RSCAN0TMTRSTSy register	TMTRSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMTARSTSy register	TMTARSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMTCSTSy register	TMTCSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMTASTSy register	TMTASTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TXQCCm register	TXQE
RSCAN0TXQSTSm register	TXQIF, TXQFLL, TXQEMP
RSCAN0THLCCm register	THLE
RSCAN0THLSTSm register	THLMC[4:0], THLIF, THLELT, THLFLL, THLEMP
RSCAN0GTINTSTS0 register	TSIFm, TAIFm, TQIFm, CFTIFm, THIFm (m = 0 to 3)
RSCAN0GTINTSTS1 register	TSIFm, TAIFm, TQIFm, CFTIFm, THIFm (m = 4, 5)

Table 20.94 Registers Initialized Only in Global Reset Mode

Register	Bit / Flag
RSCAN0GSTS register	GHLTSTS
RSCAN0GERFL register	THLES, MES, DEF
RSCAN0GTSC register	TS[15:0]
RSCAN0RMNDy register	RMNSq
RSCAN0RFCCx register	RFE
RSCAN0RFSTSx register	RFMC[7:0], RFIF, RFMLT, RFFLL, RFEMP
RSCAN0CFCCk register	When transmit/receive FIFO buffer is in receive mode: CFE
RSCAN0CFSTSk register	When transmit/receive FIFO buffer is in receive mode: CFMC[7:0], CFFLL, CFEMP, CFTXIF, CFRXIF, CFMLT
RSCAN0FESTS register	CFkEMP, RFxEMP
RSCAN0FFSTS register	CFkFLL, RFxFLL
RSCAN0FMSTS register	CFkMLT, RFxMLT
RSCAN0RFISTS register	RFxIF
RSCAN0CFRISTS register	CFkRXIF
RSCAN0GTSTCFG register	RTMPS[6:0], C0ICBCE, C1ICBCE, C2ICBCE, C3ICBCE, C4ICBCE, C5ICBCE
RSCAN0GTSTCTR register	RTME, ICBCTME

20.6 Reception Function

There are two reception types.

- Reception by receive buffers:
  Zero to 96 receive buffers can be shared by all channels. Since messages stored in receive buffers are overwritten at each reception, the latest receive data can always be read.
- Reception by receive FIFO buffers and transmit/receive FIFO buffers (receive mode):
  Eight receive FIFO buffers can be shared by all channels and three dedicated transmit/receive FIFO buffers are provided for each channel. Messages of up to the number of buffer stages specified with the RFDC[2:0] and CFDC[2:0] bits can be stored in FIFO buffers and can be read sequentially from the oldest.

20.6.1 Data Processing Using the Receive Rule Table

Data processing using the receive rule table allows dispatching of selected messages to the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.

Up to 128 receive rules can be registered per channel and up to (64 × number of channels) total receive rules can be registered in the entire module. (Up to 384 receive rules can be registered in this module that has six channels.) Set receive rules for each channel. Receive rules cannot be shared with other channels. If receive rules are not set, no messages can be received. **Figure 20.6** illustrates how receive rules are registered.

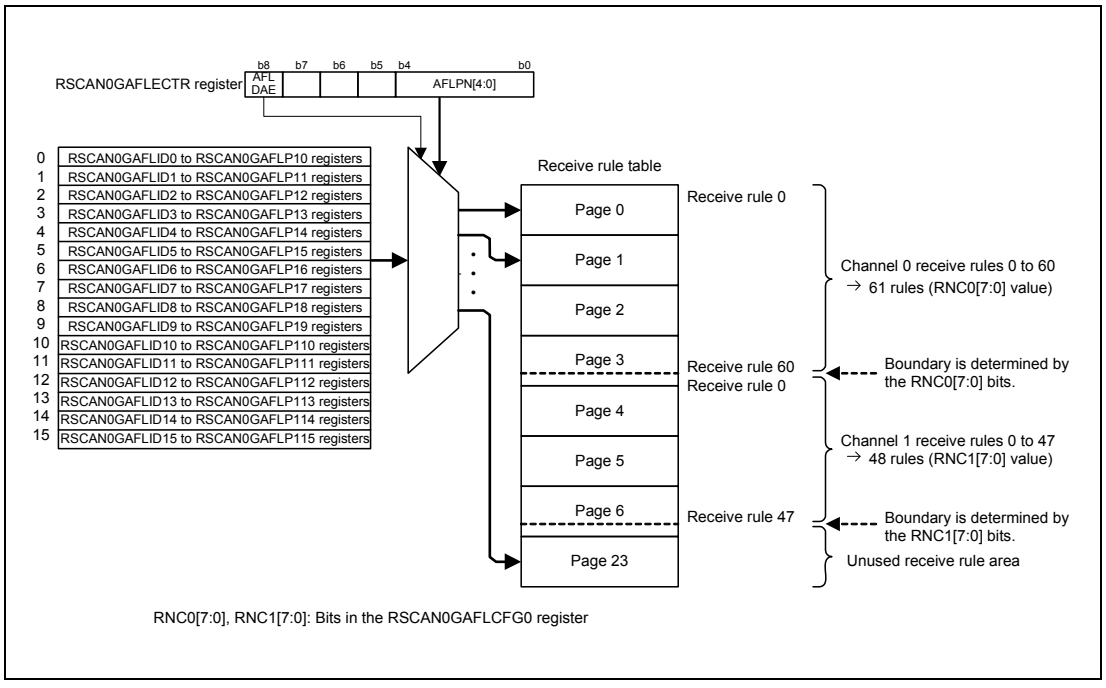


Figure 20.6 Entry of Receive Rules (for Setting Channel 0 and 1)

CAUTION

Receive rules for each channel must be set in contiguous blocks.
Channel 1 rules and channel 0 rules must be set separately.

Each receive rule consists of 16 bytes in the RSCAN0GAFLIDj, RSCAN0GAFLMj, RSCAN0GAFLP0j, and RSCAN0GAFLP1j registers (j = 0 to 15). The RSCAN0GAFLIDj register is used to set GAFLID, GAFLIDE bit, GAFLRTR bit, and the mirror function, the RSCAN0GAFLMj register is used to set mask, the RSCAN0GAFLP0j register is used to set label information to be added, DLC value, and storage receive buffer, and the RSCAN0GAFLP1j register is used to set storage FIFO buffer. Up to 16 receive rules can be set per page.

20.6.1.1 Acceptance Filter Processing

In the acceptance filter processing, the ID data, IDE bit, and RTR bit in a received message are compared with the ID data, IDE bit, and RTR bit set in the receive rule of the corresponding channel. When all these bits match, the message passes through the acceptance filter processing. The ID data, IDE bit, and RTR bit in the received message which correspond to the bits set to 0 (bits are not compared) in the RSCAN0GAFLMj register are not compared and are regarded as matched.

Check begins with the receive rule of the minimum number for the corresponding channel. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. If there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.

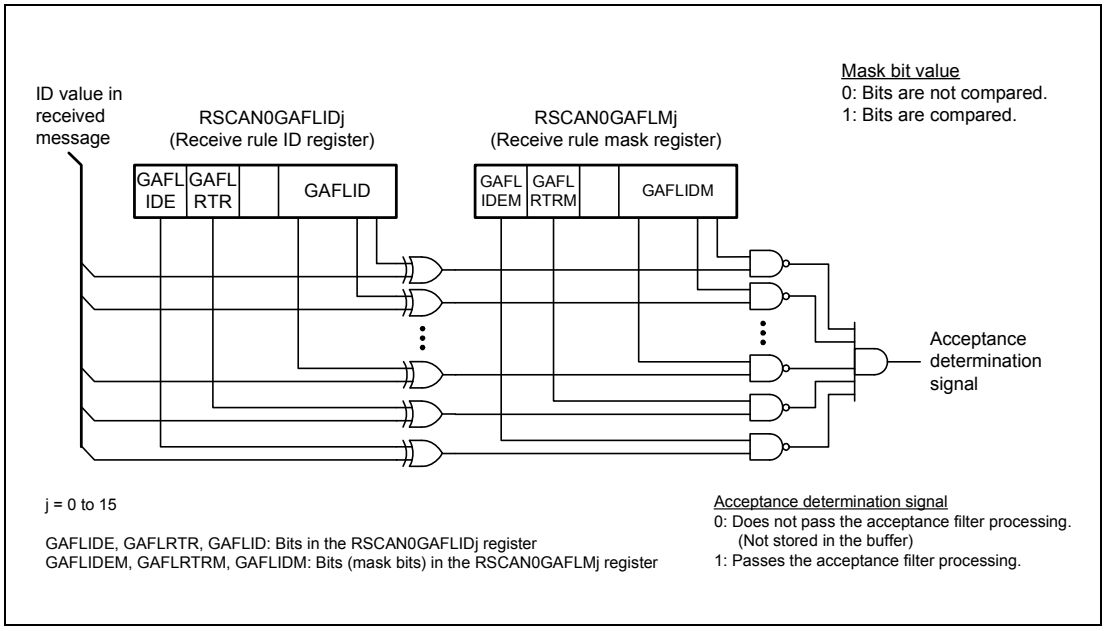


Figure 20.7 Acceptance Filter Function

20.6.1.2 DLC Filter Processing

When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), DLC filter processing is added to messages that passed through the acceptance filter processing. When the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing.

When a message has passed through the DLC filter processing with the DRE bit in the RSCAN0GCFG register set to 0 (DLC replacement is disabled), the DLC value in the received message is stored in the buffer. In this case, all the data bytes in the received message are stored in the buffer.

When a message has passed through the DLC filter processing with the DRE bit in the RSCAN0GCFG register set to 1 (DLC replacement is enabled), the DLC value in the receive rule is stored in the buffer instead of the DLC value in the received message. In this case, a value of 00_H is stored in each data byte beyond the number of bytes which is indicated by the DLC value in the receive rule.

When the DLC value in the received message is smaller than that in the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and the DEF flag in the RSCAN0GERFL register is set to 1 (a DLC error is present).

20.6.1.3 Routing Processing

Messages that passed through the acceptance filter processing and the DLC filter processing are stored in receive buffers, receive FIFO buffers, or transmit/receive FIFO buffers (set to receive mode or gateway mode). Message storage destination is set by the GAFLRMV and GAFLRMDP[6:0] bits in the RSCAN0GAFLP0j register (j = 0 to 15) and by the RSCAN0GAFLP1j register. Messages that passed through the acceptance filter processing and the DLC filter processing can be stored in up to eight buffers.

20.6.1.4 Label Addition Processing

It is possible to add 12-bit label information to messages that passed through the filter processing and store them in buffers. This label information is set in the GAFLPTR[11:0] bits in the RSCAN0GAFLP0j register.

20.6.1.5 Mirror Function Processing

The mirror function allows the CAN node to receive its own transmitted messages. The mirror function is made available by setting the MME bit in the RSCAN0GCFG register to 1 (mirror function is enabled).

When the mirror function is in use, receive rules for which the GAFLLB bit in the RSCAN0GAFLIDj register is set to 0 are used for data processing when receiving messages transmitted from other CAN nodes. When the CAN node is receiving its own transmitted messages, receive rules for which the GAFLLB bit is set to 1 are used for data processing.

20.6.1.6 Timestamp

The timestamp counter is a 16-bit free-running counter used for recording message receive time. The timestamp counter value is fetched at the start-of-frame (SOF) timing of a message and is then stored in a receive buffer or a FIFO buffer together with the message ID and data. Either pclk/2 or the CANm bit time clock (m = 0 to 5) may be selected as a timestamp counter clock source using the TSBTCS[2:0] and TSSS bits in the RSCAN0GCFG register. The timestamp counter count source is obtained by dividing the selected clock source by the TSP[3:0] bits in the RSCAN0GCFG register.

When the CANm bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When the pclk/2 is used as a clock source, the timestamp function is not affected by channel mode.

The timestamp counter value is reset to 0000_H by setting the TSRST bit in the RSCAN0GCTR register to 1.

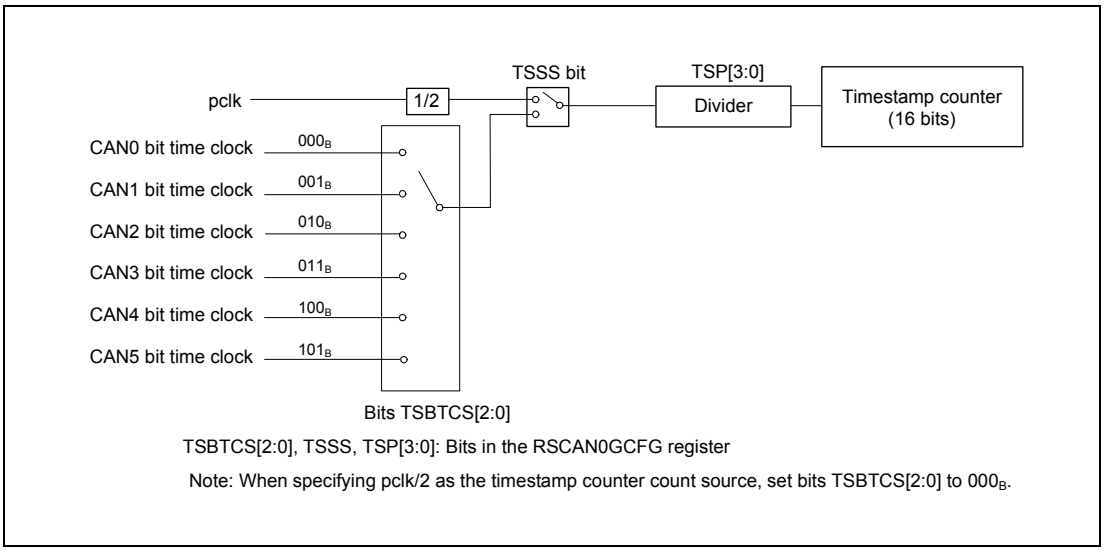


Figure 20.8 Timestamp Function Block Diagram

20.7 Transmission Functions

There are three types of transmission.

- Transmission using transmit buffers:
  Each channel has 16 buffers.
- Transmission using transmit/receive FIFO buffers (transmit mode):
  Each channel has three FIFO buffers. Up to 128 messages can be contained in a single FIFO buffer. Each FIFO buffer is used with a link to a transmit buffer. Only the message to be transmitted next in a FIFO buffer becomes the target of transmit priority determination. Messages are transmitted sequentially on a first-in, first-out basis.
- Transmission using transmit queues:
  Up to 16 transmit buffers per channel can be allocated to the transmit queues. Transmit buffer ((16 × m) + 15) is used as an access window of a corresponding channel. Transmit buffers are allocated to transmit queues in descending order of buffer number. All messages in transmit queues, which are targets of priority determination, are transmitted in the order of ID number.

Figure 20.9 shows the allocation of transmit queues and transmit/receive FIFO buffer link.

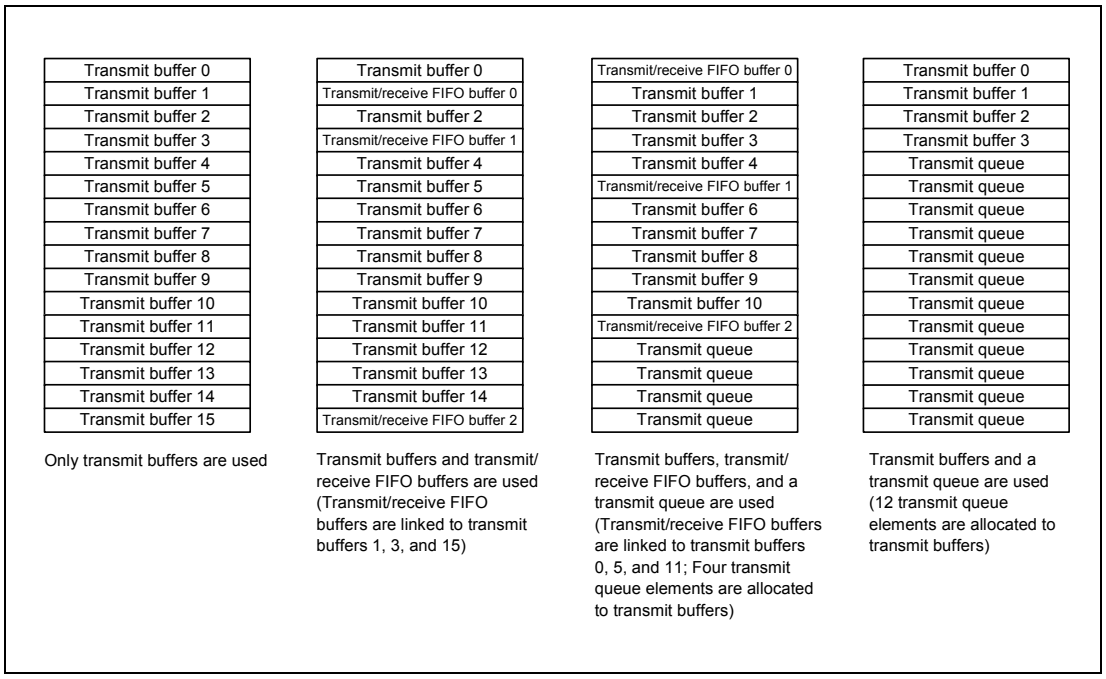


Figure 20.9 Allocation of Transmit Queues and Transmit/Receive FIFO Buffer Links

20.7.1 Transmit Priority Determination

If transmit requests are issued from multiple buffers or from the queue on the same channel, transmit priority is determined using one of the following methods.

The priority is determined by using one of the following methods.

- ID priority (TPRI bit = 0)
- Transmit buffer number priority (TPRI bit = 1)

All CAN channels use the setting of the TPRI bit in the RSCAN0GCFG register.

When the TPRI bit is set to 0, messages are transmitted according to the priority of stored message IDs. ID priority conforms to the CAN bus arbitration specification defined in the CAN specifications. All IDs of pending transmit messages are targets of priority determination, regardless of whether they are stored in transmit buffers, transmit/receive FIFO buffers (set to transmit mode or gateway mode), or the transmit queue. If even a single transmit queue is used, select ID priority. When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes the target of priority determination. When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the FIFO buffer becomes the target of priority determination. When a transmit queue is used, all messages in the transmit queue are targets of priority determination. If the same ID is set for two or more buffers, the buffer with the smaller buffer number takes precedence.

When the TPRI bit is set to 1, the message in the transmit buffer with the minimum buffer number among all buffers with a transmit request is transmitted first. When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to linked transmit buffer numbers.

When messages are retransmitted due to an arbitration-lost or an error, transmit priority determination is made again according to the TPRI bit.

20.7.2 Transmission Using Transmit Buffers

Setting the transmit request bit (TMTR bit in the RSCAN0TMCp register) in a transmit buffer to 1 (transmission is requested) allows transmission of data frames or remote frames.

The transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCAN0TMSTSp register (p = 0 to 95). When transmit completes successfully, the TMTRF[1:0] flag is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)).

20.7.2.1 Transmit Abort Function

With respect to transmit buffers for which the TMTRM bit in the RSCAN0TMSTSp register is set to 1 (a transmit request is present), when the TMTAR bit in the RSCAN0TMCp register is set to 1 (transmit abort is requested), the transmit request is canceled. When transmit abort is completed, the TMTRF[1:0] flag in the RSCAN0TMSTSp register is set to 01_B (transmit abort has been completed) and the transmit request is canceled (clearing the TMTRM bit to 0).

A message that is being transmitted or a message to be transmitted next according to the transmit priority determination cannot be aborted. However, when an arbitration-lost or an error occurs during transmission of a message for which the TMTAR bit is set to 1, retransmission is not performed.

20.7.2.2 One-Shot Transmission Function (Retransmission Disabling Function)

When the TMOM bit in the RSCAN0TMCp register is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration-lost or an error occurs, retransmission is not performed.

The one-shot transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCAN0TMSTSp register. When one-shot transmission completes successfully, the TMTRF[1:0] flag is set to 10_B or 11_B. When an arbitration-lost or an error occurs, the TMTRF[1:0] flag is set to 01_B (transmit abort has been completed).

20.7.3 Transmission Using FIFO Buffers

Multiple messages can be stored in a single transmit/receive FIFO buffers, up to the number specified by the FIFO buffer depth, which is set by the CFDC[2:0] bits in the RSCAN0CFCCk register (k = 0 to 17). Messages are transmitted sequentially on a first-in, first-out basis.

Each transmit/receive FIFO buffer is linked to a transmit buffer selected by the CFTML[3:0] bits in the RSCAN0CFCCk register. When the CFE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO buffers are used), transmit/receive FIFO buffers become targets of transmit priority determination. Priority of only the next transmit message is determined in the FIFO buffer.

When the CFE bit is set to 0 (no transmit/receive FIFO buffer is used), the CFEMP flag is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately if the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration-lost in the case that a message in it is being transmitted or to be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages cannot be stored in FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

20.7.3.1 Interval Transmission Function

A message transmission interval time can be set to space the transmission of messages from the same FIFO buffer when using a transmit/receive FIFO buffer set to transmit mode or gateway mode.

Immediately after the first message has been transmitted successfully from the FIFO buffer with the CFE bit in the RSCAN0CFCCk register set to 1, the interval timer starts counting (after EOF7 of the CAN protocol). After that, when the interval time has passed, the next message is transmitted. The interval timer stops in channel reset mode or by clearing the CFE bit to 0.

The interval time is set by the CFITT[7:0] bits in the RSCAN0CFCCk register. When the interval timer is not used, set the CFITT[7:0] bits to 00_H.

Select an interval timer count source using the CFITR and CFITSS bits in the RSCAN0CFCCk register. When the CFITR and CFITSS bits are set to 00_B, the count source is obtained by dividing pclk/2 by the value of the ITRCP[15:0] bits. When the CFITR and CFITSS bits are set to 10_B, the count source is obtained by dividing pclk/2 by (the value of the ITRCP[15:0] bits in the RSCAN0GCFG register × 10). When the CFITR and CFITSS bits are set to $x1_B$, the CANm bit time clock is used as a count source.

The interval time is calculated by the following equations where M is the set value of ITRCP[15:0] bits and N is the value of CFITT[7:0]] bits.

- When CFITR and CFITSS bits = 00_B (fPBA is the frequency of pclk):

$$\frac{1}{f_{PBA}} \times 2 \times M \times N$$

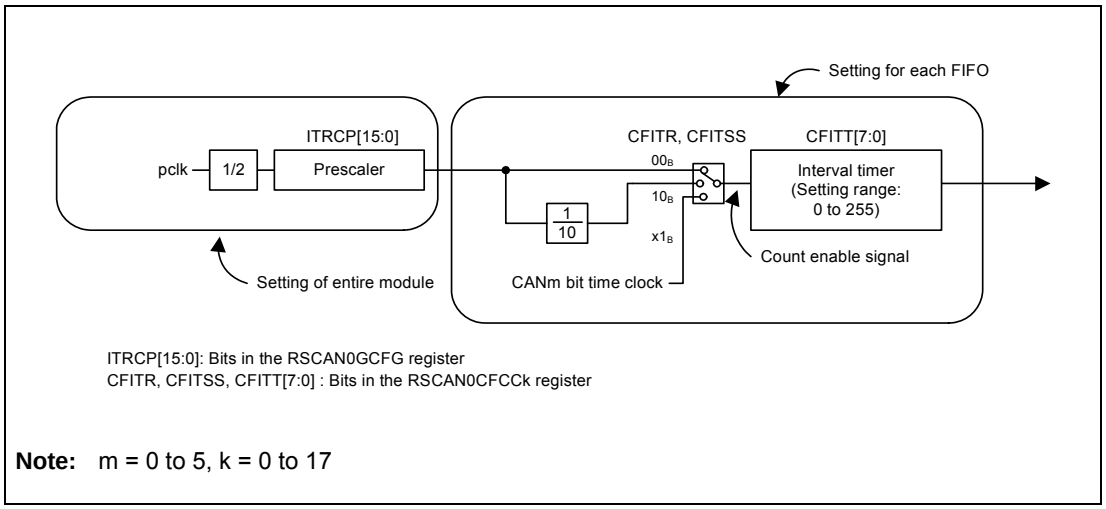
- When CFITR and CFITSS bits = 10_B:

$$\frac{1}{f_{PBA}} \times 2 \times M \times 10 \times N$$

- When CFITR and CFITSS bits = $x1_B$ (fCANBIT is the frequency of CANm bit time clock):

$$\frac{1}{f_{CANBIT}} \times N$$

Figure 20.10 shows the interval timer block diagram.



Note: m = 0 to 5, k = 0 to 17

Figure 20.10 Interval Timer Block Diagram

Figure 20.11 shows the interval timer timing diagram.

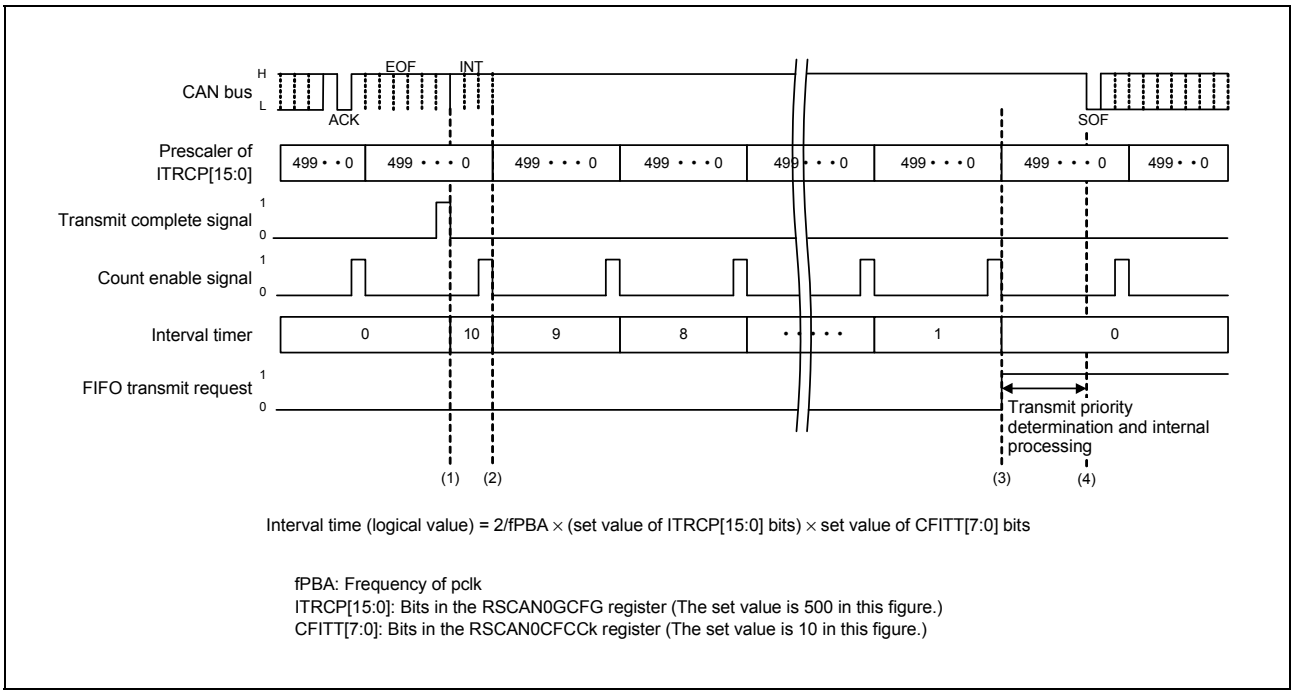


Figure 20.11 Interval Timer Timing Chart

(1) The interval timer starts counting upon completion of transmission. Since the prescaler is not initialized at the time of transmission completion, the first interval time contains an error of up to one count of the interval timer.

(2) The interval timer is decremented by the next count enable signal.

(3) When the interval timer has decreased to 0, the transmit/receive FIFO buffer issues a transmit request.

(4) The transmit/receive FIFO buffer is determined for the next transmission by the priority determination, it starts transmitting data. Transmission starts usually with a delay of three CANm bit time clock cycles or less from the issue of transmit request. If multiple internal processes (such as receive filter processing, message routing, and transmit priority determination) take place in all channels, a delay of up to 582 cycles of the pclk may be generated.

20.7.4 Transmission Using Transmit Queues

Three to sixteen buffers are allocated to a transmit queue for each channel, and transmit buffer ((16 × m) + 15) is used as an access window of a corresponding channel.

All messages in a transmit queue are targets of transmit priority determination and are transmitted in the ID priority order regardless of storage sequence. If two messages having the same ID are stored in a transmit queue, these messages are not always transmitted in the order of their storage in the transmit queue.

Setting the TXQE bit in the RSCAN0TXQCCm register to 0 disables transmit queues. When the TXQE bit is set to 0, the TXQEMP flag in the RSCAN0TXQSTSm register is set to 1 (the transmit queue contains no messages (transmit queue empty)) at the timing below.

- The transmit queue becomes empty immediately when no message in it is being transmitted or will be transmitted next.
- The transmit queue becomes empty after transmission completion, CAN bus error detection, or arbitration-lost when a message in it is being transmitted or will be transmitted next.

When the TXQE bit is cleared to 0, all messages in transmit queues are lost and messages cannot be stored in transmit queues. Confirm that the TXQEMP flag is set to 1 before setting the TXQE bit to 1 again.

20.7.5 Transmit History Function

Information about transmission-completed messages can be stored in the transmit history buffer. Each channel has a single transmit history buffer that can contain 16 sets of transmit history data.

A message transmit source buffer type can be selected by the THLDTE bit in the RSCAN0THLCCm register. The THLEN bit in the RSCAN0CFIDk register (k = 0 to 17) determines whether transmit history data is stored for each message.

The following information on a transmitted message will be stored in the transmission history buffer after the successful completion of transmission.

Storage of the transmission history data after the successful completion of transmission may take up to 152 cycles of pclk.

- Buffer type — 001_B: Transmit buffer
010_B: Transmit/receive FIFO buffer
100_B: Transmit queue
- Buffer number — Number of source transmit buffer, transmit queue, or transmit/receive FIFO buffer. This number depends on buffer types. See **Table 20.95**.
- Label data — Label information of the transmit message

Table 20.95 Transmit History Data Buffer Numbers

Buffer No. / Buffer type	001_B	010_B	100_B
0000_B	Transmit buffer 16 × m + 0	Buffer numbers of the transmit buffer linked to the transmit/receive FIFO buffer by the CFTML[3:0] bits in the RSCAN0CFCCk register (k = 0 to 17)	Buffer numbers of the transmit buffer allocated to the transmit queue that performed transmission
0001_B	Transmit buffer 16 × m + 1		
0010_B	Transmit buffer 16 × m + 2		
0011_B	Transmit buffer 16 × m + 3		
0100_B	Transmit buffer 16 × m + 4		
0101_B	Transmit buffer 16 × m + 5		
0110_B	Transmit buffer 16 × m + 6		
0111_B	Transmit buffer 16 × m + 7		
1000_B	Transmit buffer 16 × m + 8		
1001_B	Transmit buffer 16 × m + 9		
1010_B	Transmit buffer 16 × m + 10		
1011_B	Transmit buffer 16 × m + 11		
1100_B	Transmit buffer 16 × m + 12		
1101_B	Transmit buffer 16 × m + 13		
1110_B	Transmit buffer 16 × m + 14		
1111_B	Transmit buffer 16 × m + 15		

Label data is used to identify each message. Unique label data can be added to each message transmitted from a transmit buffer, transmit queue, or transmit/receive FIFO buffer.

Transmit history data can be read from the RSCAN0THLACCm register. If an attempt is made to store new transmit history data while the buffer is full, the buffer overflows and the new data is discarded.

20.8 Gateway Function

When a transmit/receive FIFO buffer is set to gateway mode, receive massages can be transmitted from an arbitrary channel without CPU intervention.

When the CFM[1:0] bits in the RSCAN0CFCCk register are set to 10_B (gateway mode) and the transmit/receive FIFO buffer of a channel transmitting a message is selected in RSCAN0GAFLP1j register, messages that pass through filter processing according to the reception rule are stored in the specified transmit/receive FIFO buffer and are automatically transmitted from the buffer.

Messages stored in a transmit/receive FIFO buffer are transmitted sequentially on a first-in, first-out basis. Only the message to be transmitted next becomes the target of transmit priority determination.

Transmit/receive FIFO buffers in the gateway mode are disabled by setting the CFE bit in the RSCAN0CFCCk register to 0 and the CFEMP flag becomes 1 according to the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the message in it is not being transmitted and will not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration-lost when the message in it is being transmitted or will be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages can no longer be stored in transmit/receive FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

20.9 Test Function

The test function is classified into communication tests and global tests.

- Communication tests: Performed for each channel.
  - Standard test mode
  - Listen-only mode
  - Self-test mode 0 (external loopback mode)
  - Self-test mode 1 (internal loopback mode)
- Global tests: Performed for the entire module
  - RAM test (read/write test)
  - Inter-channel communication test

20.9.1 Standard Test Mode

Standard test mode allows CRC test.

20.9.2 Listen-Only Mode

Listen-only mode allows reception of data frames and remote frames. Only recessive bits are transmitted on the CAN bus, and the ACK bit, overload flag, and active error flag are not transmitted.

Listen-only mode is available for detecting the communication speed.

Do not make a transmit request from any buffer or queue in listen-only mode.

Figure 20.12 shows the connection when listen-only mode is selected.

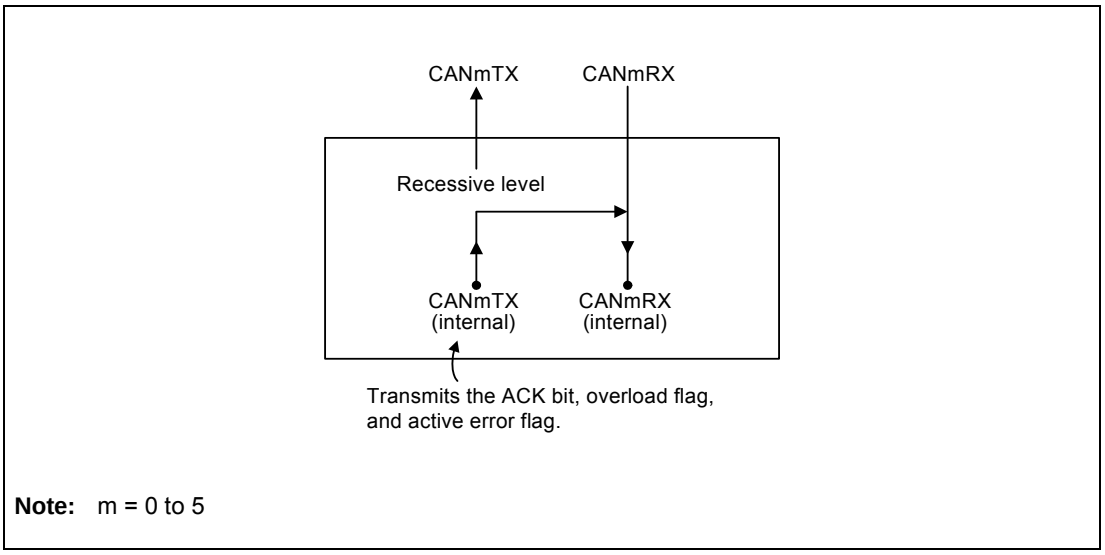


Figure 20.12 Connection when Listen-Only Mode is Selected

20.9.3 Self-Test Mode (Loopback Mode)

In self-test mode, transmitted messages are compared with the receive rule of the own channel and the messages are stored in a buffer if they have passed through the filter processing. Messages transmitted from other CAN nodes are compared only with the receive rule for which the GAFLLB bit in the RSCAN0GAFLIDj register (j = 0 to 15) is set to 0 (when a message transmitted from another CAN node is received).

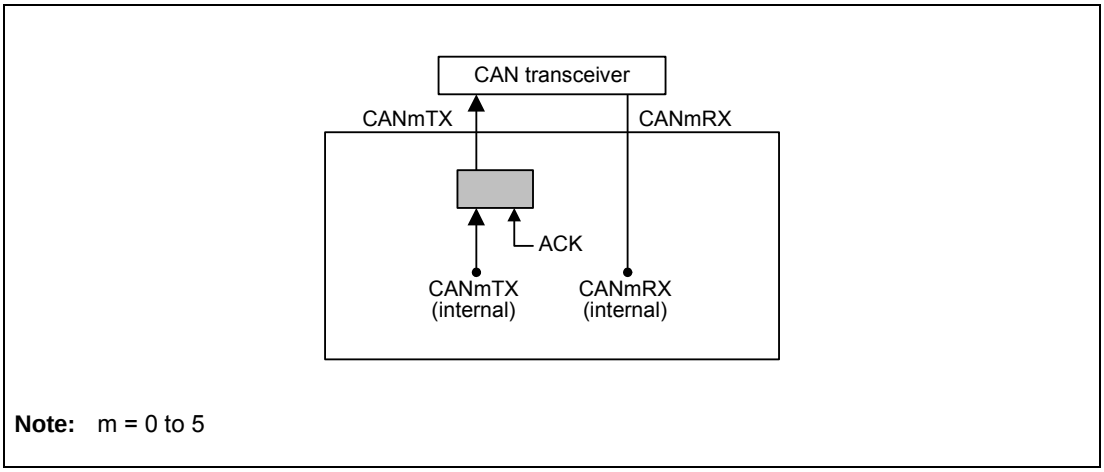
If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

20.9.3.1 Self-Test Mode 0 (External Loopback Mode)

Self-test mode 0 is used to perform a loopback test within a channel including the CAN transceiver.

In self-test mode 0, transmitted messages are handled as messages received through the CAN transceiver and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

Figure 20.13 shows the connection when self-test mode 0 is selected.



Note: m = 0 to 5

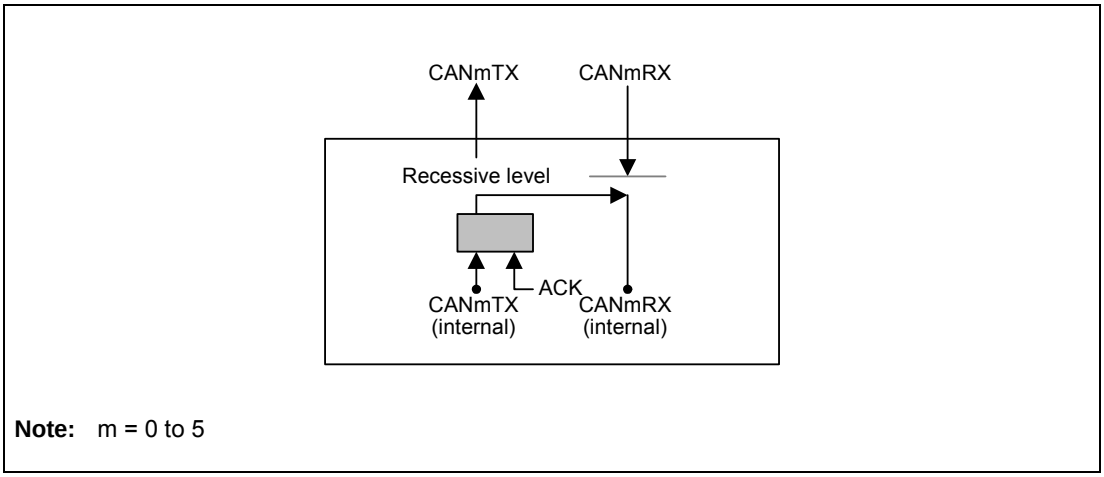
Figure 20.13 Connection when Self-Test Mode 0 is Selected

20.9.3.2 Self-Test Mode 1 (Internal Loopback Mode)

In self-test mode 1, transmitted messages are handled as received messages and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

In self-test mode 1, internal feedback from the internal CANmTX pin (m = 0 to 5) to the internal CANmRX pin is performed. The external CANmRX pin input is isolated. The external CANmTX pin outputs only recessive bits.

Figure 20.14 shows the connection when self-test mode 1 is selected.



Note: m = 0 to 5

Figure 20.14 Connection when Self-Test Mode 1 is Selected

20.9.4 RAM Test

The RAM test function allows accesses to all CAN RAM addresses.

When the RAM test function is used, the RAM is divided into pages of 256 bytes each. RAM test page is set by the RTMPS[6:0] bits in the RSCAN0GTSTCFG register. Data in the set page can be read from and written to the RSCAN0RPGACCr register (r = 0 to 63). The available total RAM size is shown below.

– RSCAN0: 15168 bytes ($3B40_H$).

20.9.5 Inter-Channel Communication Test

The inter-channel communication test function allows communication test by internally connecting CAN channels to each other. During this test, channels are isolated from the external CAN bus.

Before starting data transmission/reception in channel communication mode, make transmission/ reception settings for each channel.

Figure 20.15 shows the connection for inter-channel communication test.

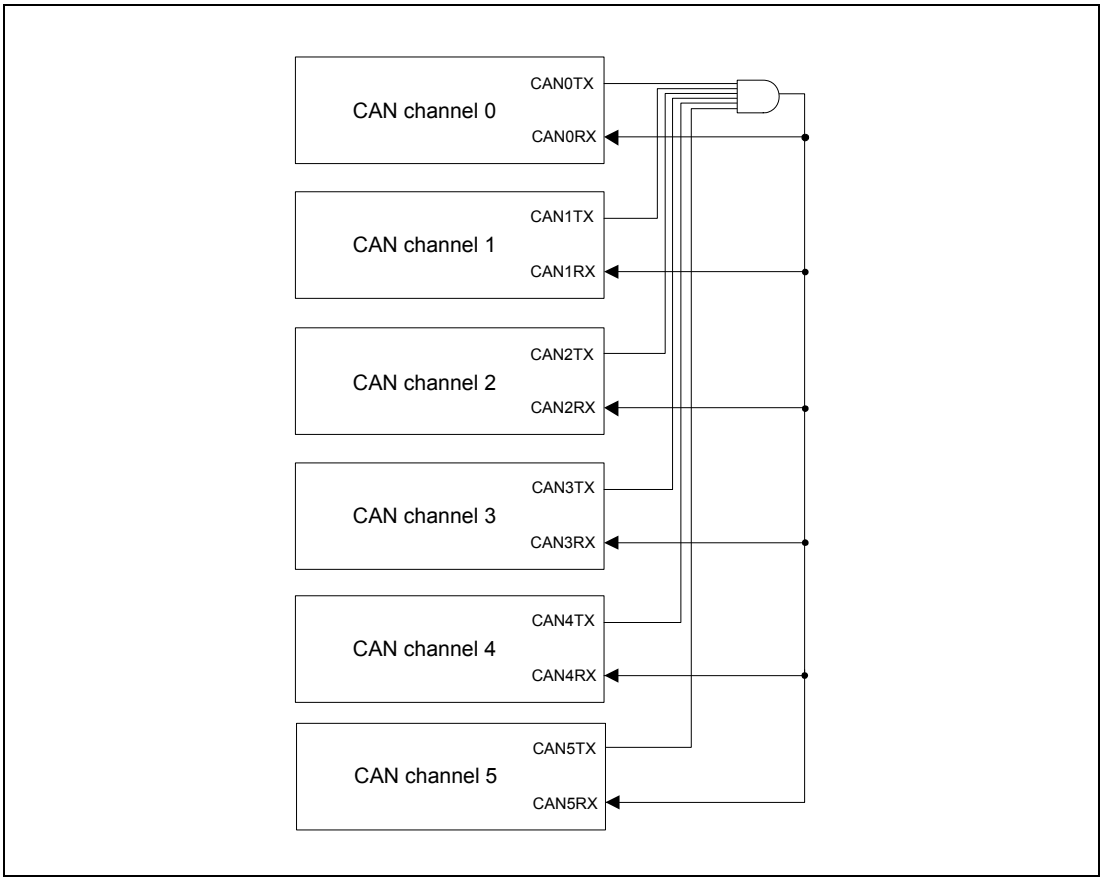


Figure 20.15 Connection for Inter-Channel Communication Test

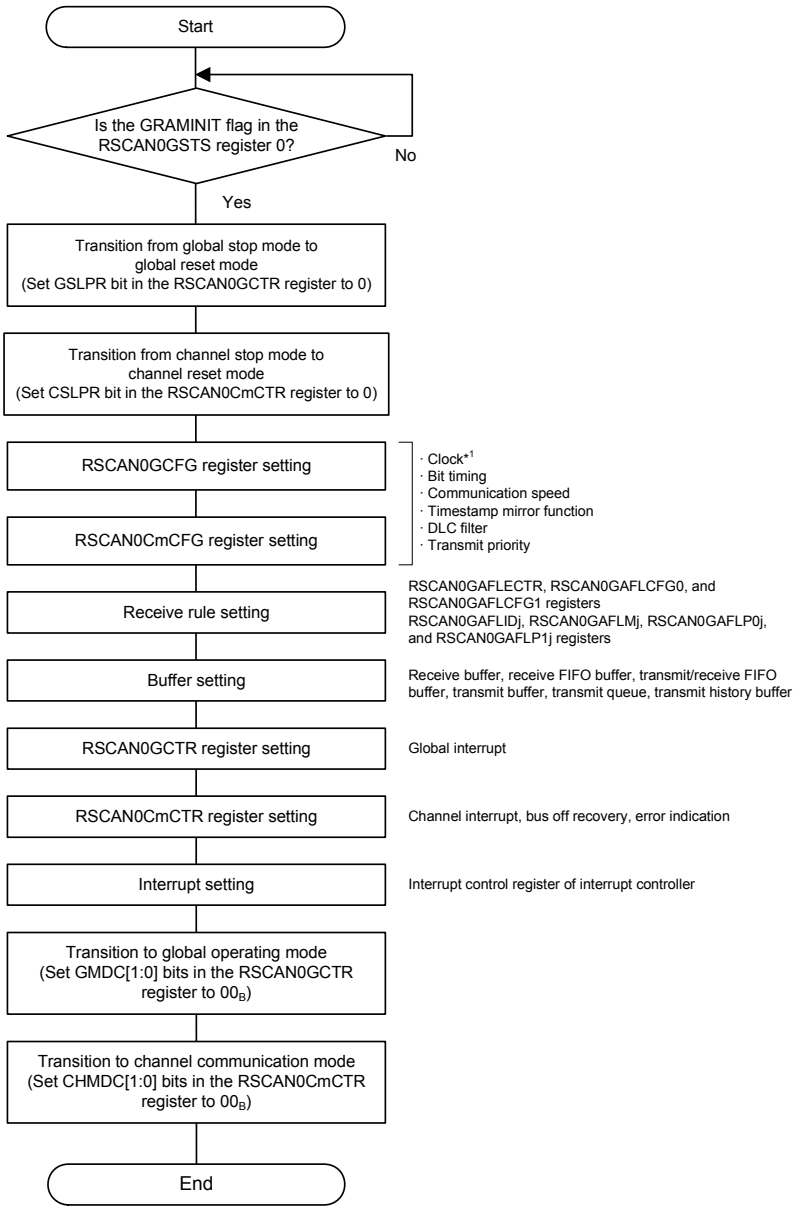
20.10 RS-CAN Setting Procedure

20.10.1 Initial Settings

The RS-CAN module initializes the CAN RAM after the MCU is reset. The RAM initialization time is shown below.

– RSCAN0: 7586 cycles of the pclk.

The GRAMINIT flag in the RSCAN0GSTS register is set to 1 (CAN RAM initialization is ongoing) during the RAM initialization and is cleared to 0 (CAN RAM initialization is finished) when the initialization is completed. Make CAN settings after the GRAMINIT flag is cleared to 0. **Figure 20.16** shows the CAN setting procedure after the MCU is reset.



Note 1. For the setting of CAN clock frequency, see the **Table 20.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1M**.

Note: m = 0 to 5, j = 0 to 15

Figure 20.16 CAN Setting Procedure after the MCU is Reset

20.10.1.1 Clock Setting

Set the CAN clock (fCAN) as a clock source of the RS-CAN module. Select the clk_xincan or clkc using the DCS bit in the RSCAN0GCFG register.

20.10.1.2 Bit Timing Setting

In the CAN protocol, one bit of a communication frame consists of three segments SS, TSEG1, and TSEG2, of which two segments TSEG1 and TSEG2 can be set by the RSCAN0CmCFG register for each channel. Sample point timing can be determined by setting these two segments. This timing can be adjusted in units of 1 Time Quantum (hereafter referred to as Tq). 1 Tq is equal to one CANmTq clock cycle. The CANmTq clock is obtained by selecting the clock source with the DCS bit in the RSCAN0GCFG register and selecting the clock division ratio with the BRP[9:0] bits in the RSCAN0CmCFG register.

Figure 20.17 shows the bit timing chart. **Table 20.96** shows an example of bit timing setting.

SS = 1 Tq fixed
Set TSEG1 to a range of 4 Tq to 16 Tq
Set TSEG2 to a range of 2 Tq to 8 Tq
Set SJW to a range of 1 Tq to 4 Tq
Set SS + TSEG1 + TSEG2 to a range of 8 Tq to 25 Tq
TSEG1 > TSEG2 ≥ SJW

- SS (synchronization segment):
  The SS is a segment that performs synchronization by monitoring the edge from recessive to dominant bits in the Interframe Space.
  Interframe Space consists of Intermission, Suspend Transmission, and Bus Idle. All nodes can start transmission during Bus Idle.
- TSEG1 (Time segment 1):
  TSEG1 is a segment that absorbs physical delay on the CAN network. The physical delay on the CAN network is twice the total of the delay on the CAN bus, the delay in the input comparator, and the delay in the output driver.
- TSEG2 (Time segment 2):
  TSEG2 is a segment that compensates phase error due to an error in frequency.
- SJW (Resynchronization jump width):
  The SJW is a length to extend or reduce the time segment to compensate for an error in phase due to phase error.

Figure 20.17 Bit Timing Chart

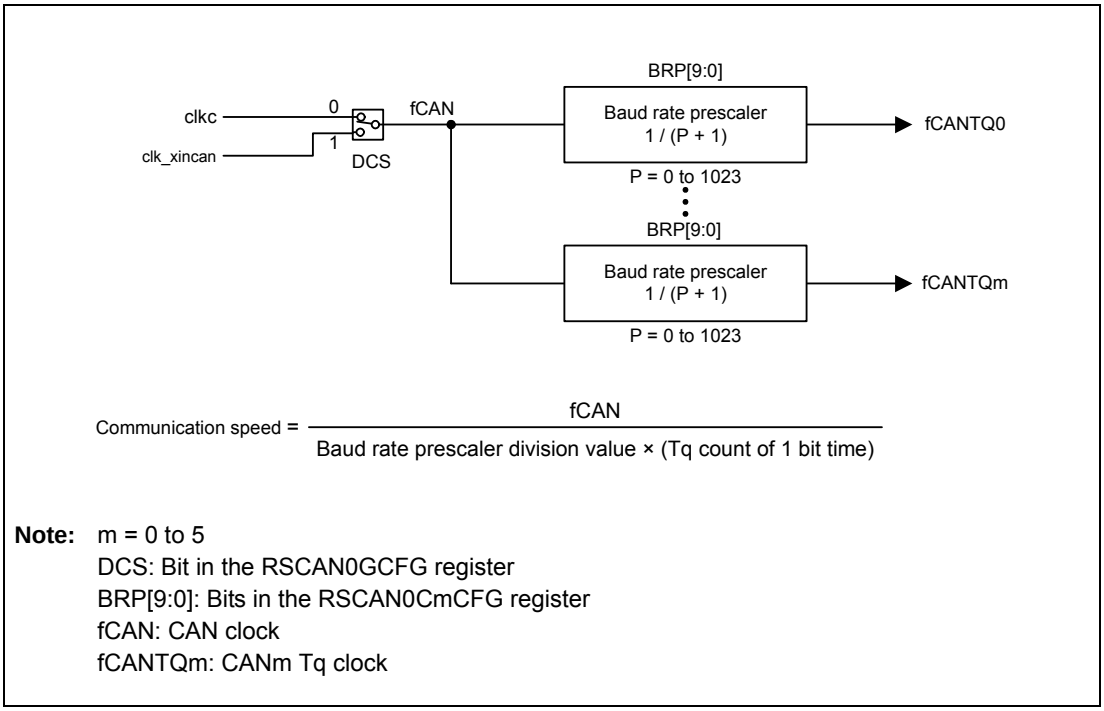
Table 20.96 Example of Bit Timing Setting

1 Bit	Set Value (Tq)				Sample Point (%) Note: See Figure 20.17.
	SS	TSEG1	TSEG2	SJW	
8 Tq	1	4	3	1	62.50
	1	5	2	1	75.00
10 Tq	1	6	3	1	70.00
	1	7	2	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00

20.10.1.3 Communication Speed Setting

Set the CAN communication speed for each channel using the fCAN, baud rate prescaler division value (BRP[9:0] bits in the RSCAN0CmCFG register), and Tq count per bit time.

Figure 20.18 shows the CAN clock control block diagram, and **Table 20.97** shows an example of the communication speed setting.



Note: m = 0 to 5
DCS: Bit in the RSCAN0GCFG register
BRP[9:0]: Bits in the RSCAN0CmCFG register
fCAN: CAN clock
fCANTQm: CANm Tq clock

Figure 20.18 CAN Clock Control Block Diagram

Table 20.97 Example of Communication Speed Setting

fCAN / Communication speed	40 MHz	32 MHz	24 MHz	16 MHz	8 MHz
1 Mbps	8 Tq (5) 20 Tq (2)	8 Tq (4) 16 Tq (2)	8 Tq (3) 12 Tq (2) 24 Tq (1)	8 Tq (2) 16 Tq (1)	8 Tq (1)
500 Kbps	8 Tq (10) 20 Tq (4)	8 Tq (8) 16 Tq (4)	8 Tq (6) 12 Tq (4) 24 Tq (2)	8 Tq (4) 16 Tq (2)	8 Tq (2) 16 Tq (1)
250 Kbps	8 Tq (20) 20 Tq (8)	8 Tq (16) 16 Tq (8)	8 Tq (12) 12 Tq (8) 24 Tq (4)	8 Tq (8) 16 Tq (4)	8 Tq (4) 16 Tq (2)
125 Kbps	8 Tq (40) 20 Tq (16)	8 Tq (32) 16 Tq (16)	8 Tq (24) 12 Tq (16) 24 Tq (8)	8 Tq (16) 16 Tq (8)	8 Tq (8) 16 Tq (4)

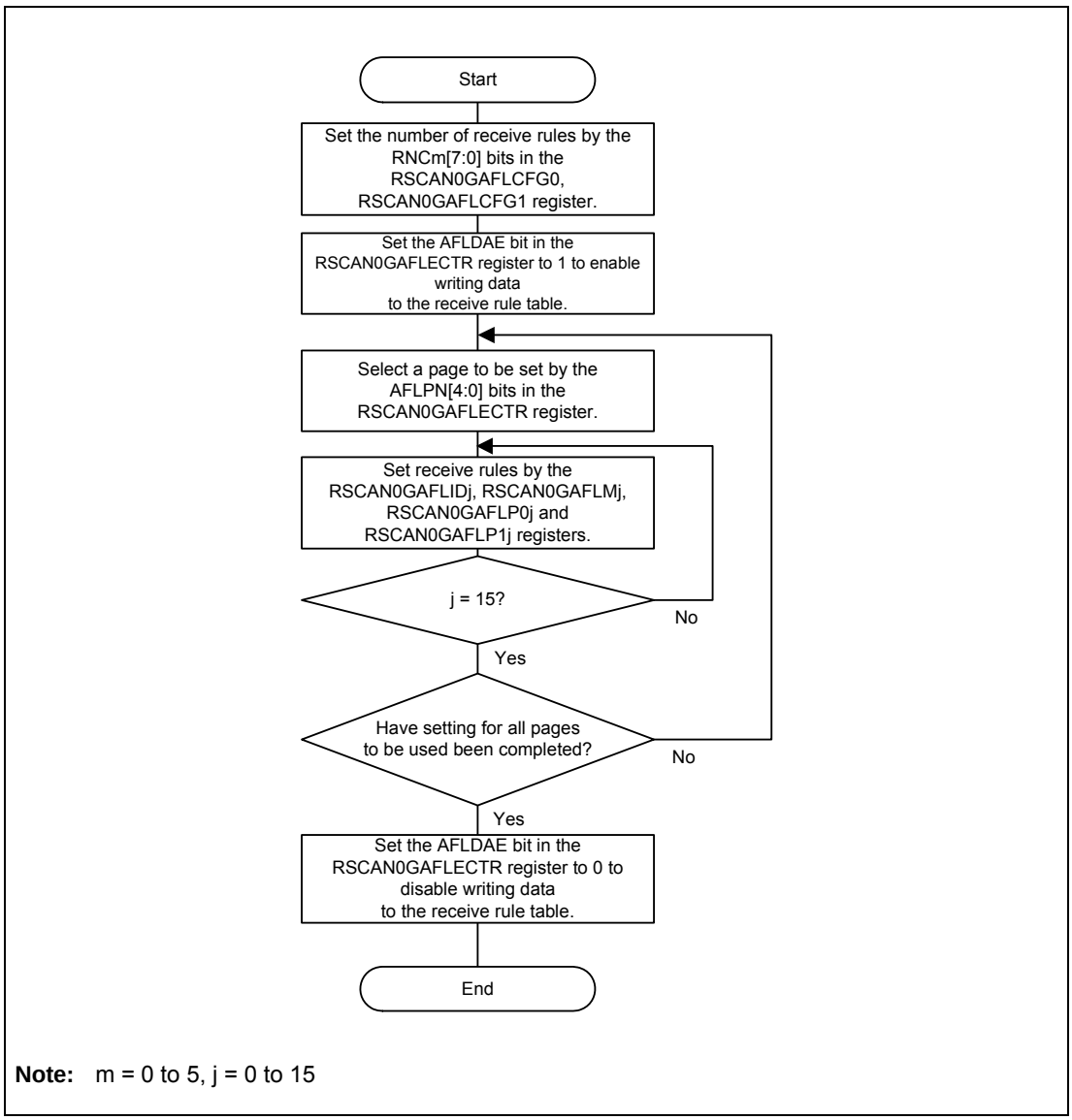
Note: Values in ( ) are baud rate prescaler division values.

20.10.1.4 Receive Rule Setting

Receive rules can be set using receive rule-related registers.

Up to 16 receive rules can be registered per page. Specify pages 0 to 23 by the AFLPN[4:0] bits in the RSCAN0GAFLECTR register. Set receive rule table write enable/disable using the AFLDAE bit.

Figure 20.19 shows the receive rule setting procedure.



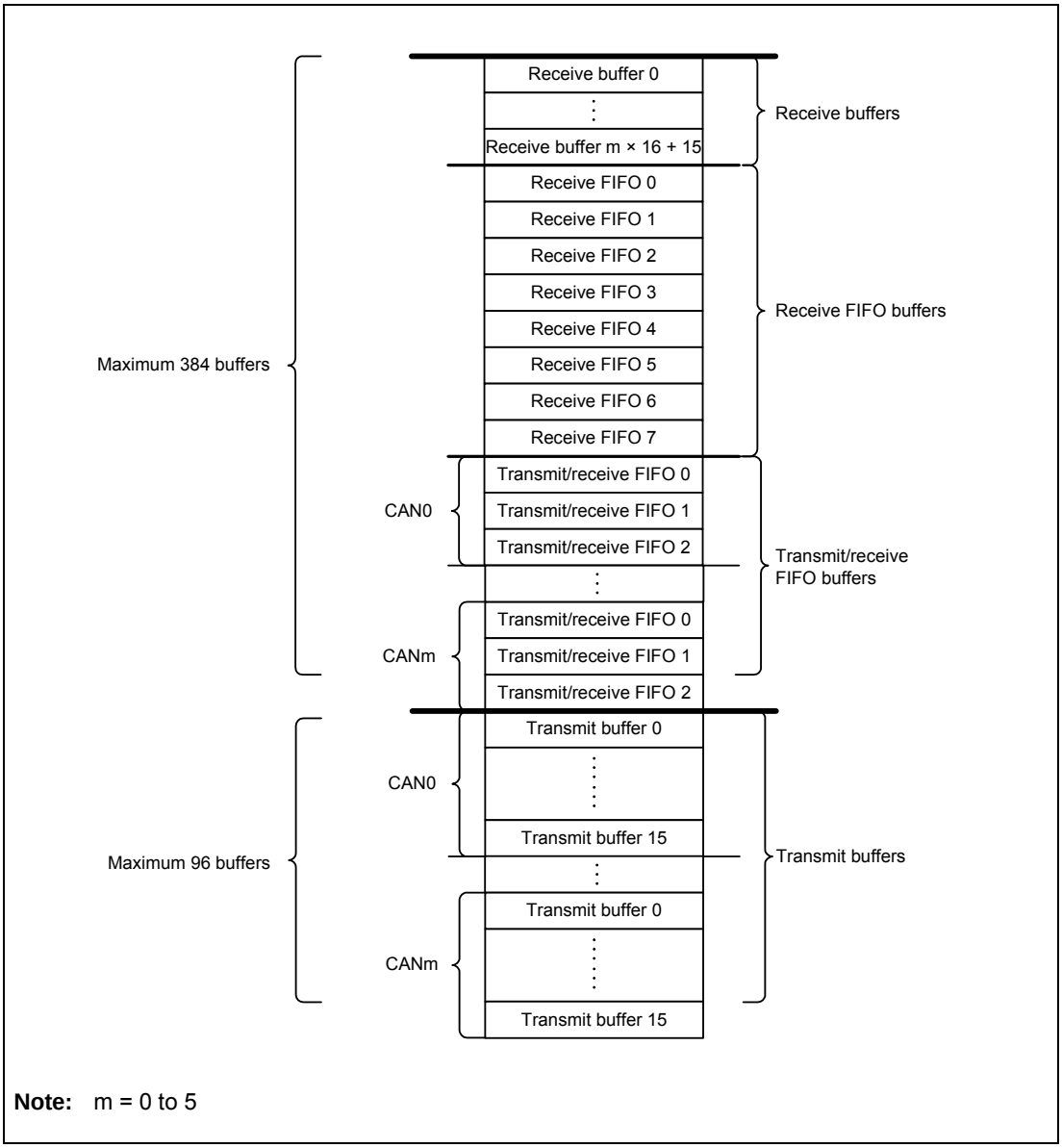
Note: m = 0 to 5, j = 0 to 15

Figure 20.19 Receive Rule Setting Procedure

20.10.1.5 Buffer Setting

Set sizes and interrupt sources of buffers. For transmit/receive FIFO buffers that are set to transmit mode, set transmit buffers to be linked.

Figure 20.20 shows the buffer configuration. **Figure 20.21** shows the buffer setting procedure.



Note: m = 0 to 5

Figure 20.20 Buffer Configuration

CAUTION

Receive buffers, receive FIFO buffers, transmit/receive FIFO buffers, and transmit buffers are located in succession.

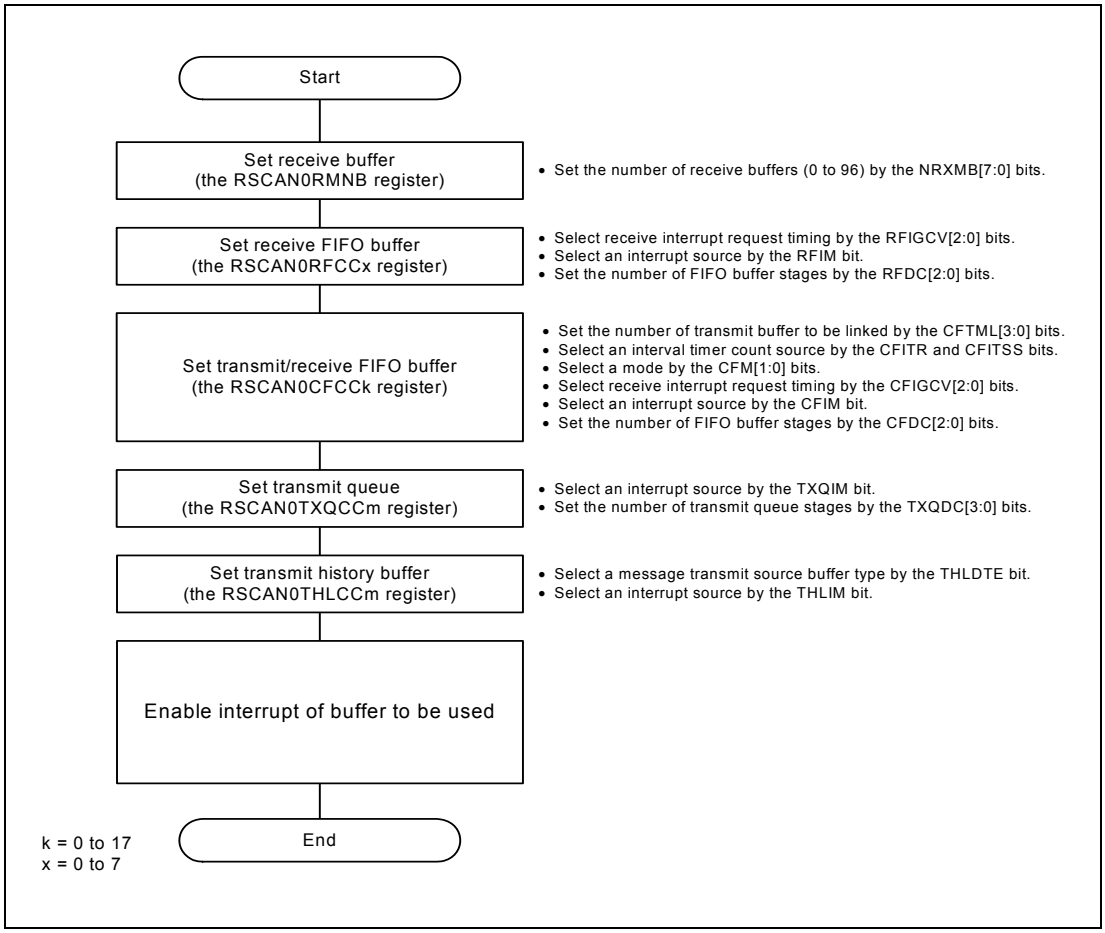
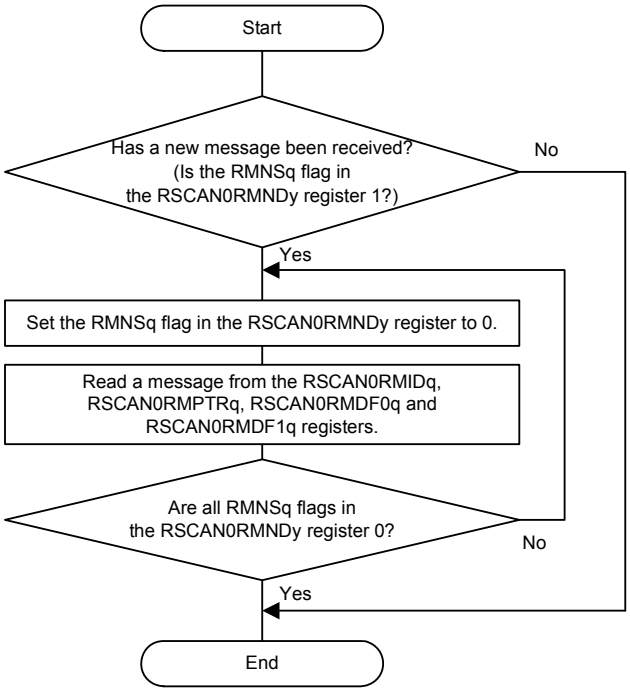


Figure 20.21 Buffer Setting Procedure

20.10.2 Reception Procedure

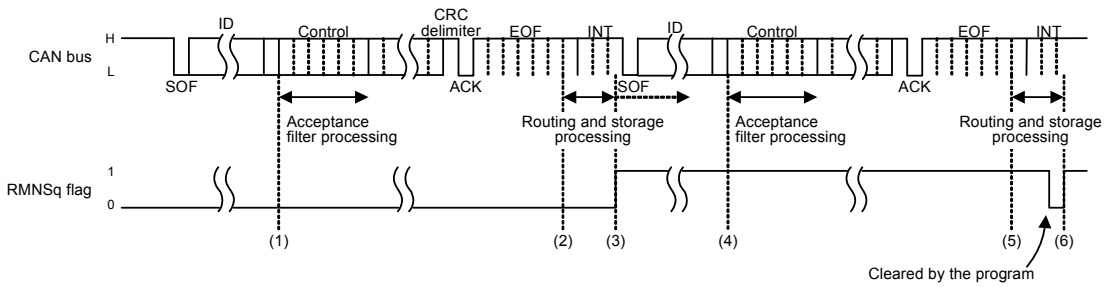
20.10.2.1 Receive Buffer Reading Procedure

When the processing to store received messages in a receive buffer starts, the RMNSq flag in the RSCAN0RMNDy register (y = 0 to 2, q = 0 to 95) is set to 1 (receive buffer q contains a new message). Messages can be read from the RSCAN0RMIDq, RSCAN0RMPTRq, RSCAN0RMDF0q, and RSCAN0RMDF1q registers. If the next message has been received before the current message is read from the receive buffer, the message is overwritten. **Figure 20.22** shows the receive buffer reading procedure.



Note: y = 0 to 2, q = 0 to 95

Figure 20.22 Receive Buffer Reading Procedure



Note: y = 0 to 2, q = 0 to 95
RMNSq: Flag in the RSCAN0RMNDy register

Figure 20.23 Receive Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing, the processing to store the message in the specified receive buffer starts.
When the message storage processing starts, the RMNSq flag in the corresponding RSCAN0RMNDy register is set to 1 (the receive buffer contains a new message). If other channels are performing filter processing or transmit priority determination processing, the routing processing and the storage processing may be delayed.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(6) When the corresponding RMNSq flag is cleared to 0 (the receive buffer contains no new message), this flag is set to 1 again when the message storage processing starts. Even if the RMNSq flag remains 1, a new message is overwritten to the receive buffer. The RMNSq flag should not be cleared to 0 during storage of messages.

20.10.2.2 FIFO Buffer Reading Procedure

When received messages have been stored in one or more receive FIFO buffers or a transmit/receive FIFO buffer that is set to receive mode or gateway mode, the corresponding message count display counter (RFMC[7:0] bits in the RSCAN0RFSTSx register (x = 0 to 7) or CFMC[7:0] bits in the RSCAN0CFSTSk register (k = 0 to 17)) is incremented. At this time, when the RFIE bit (receive FIFO interrupt is enabled) in the RSCAN0RFCCx register or the CFRXIE bit (transmit/receive FIFO receive interrupt is enabled) in the RSCAN0CFCCk register is set to 1, an interrupt request is generated. Received messages can be read from the RSCAN0RFIDx, RSCAN0RFPTRx, RSCAN0RFDF0x, and RSCAN0RFDF1x registers for receive FIFO buffers, or from the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers for transmit/receive FIFO buffers. Messages in FIFO buffers can be read sequentially on a first-in, first-out basis.

When the message count display counter value matches the FIFO buffer depth (a value set by the RFDC[2:0] bits in the RSCAN0RFCCx register or the CFDC[2:0] bits in the RSCAN0CFCCk register), the RFFLL or CFFLL flag is set to 1 (the receive FIFO buffer is full).

When all messages have been read out of the FIFO buffer, the RFEMP flag in the RSCAN0RFSTSx register or the CFEMP flag in the RSCAN0CFSTSk register is set to 1 (the receive FIFO buffer contains no unread message (buffer empty)).

If the RFE bit or the CFE bit is cleared to 0 (no receive FIFO buffer is used) with the interrupt request flag (RFIF flag in the RSCAN0RFSTSx register or CFRXIF flag in the RSCAN0CFSTSk register) set to 1 (a receive FIFO interrupt request is present), the interrupt request flag is not automatically cleared to 0. The program must clear the interrupt request flag to 0.

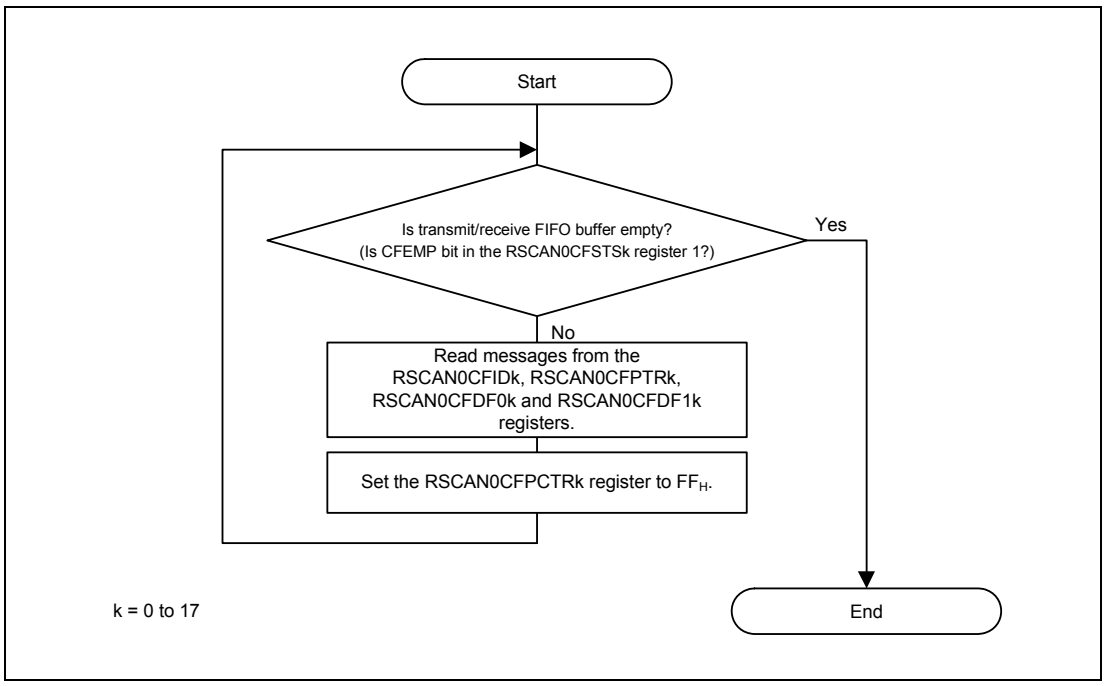


Figure 20.24 Transmit/Receive FIFO Buffer Reading Procedure

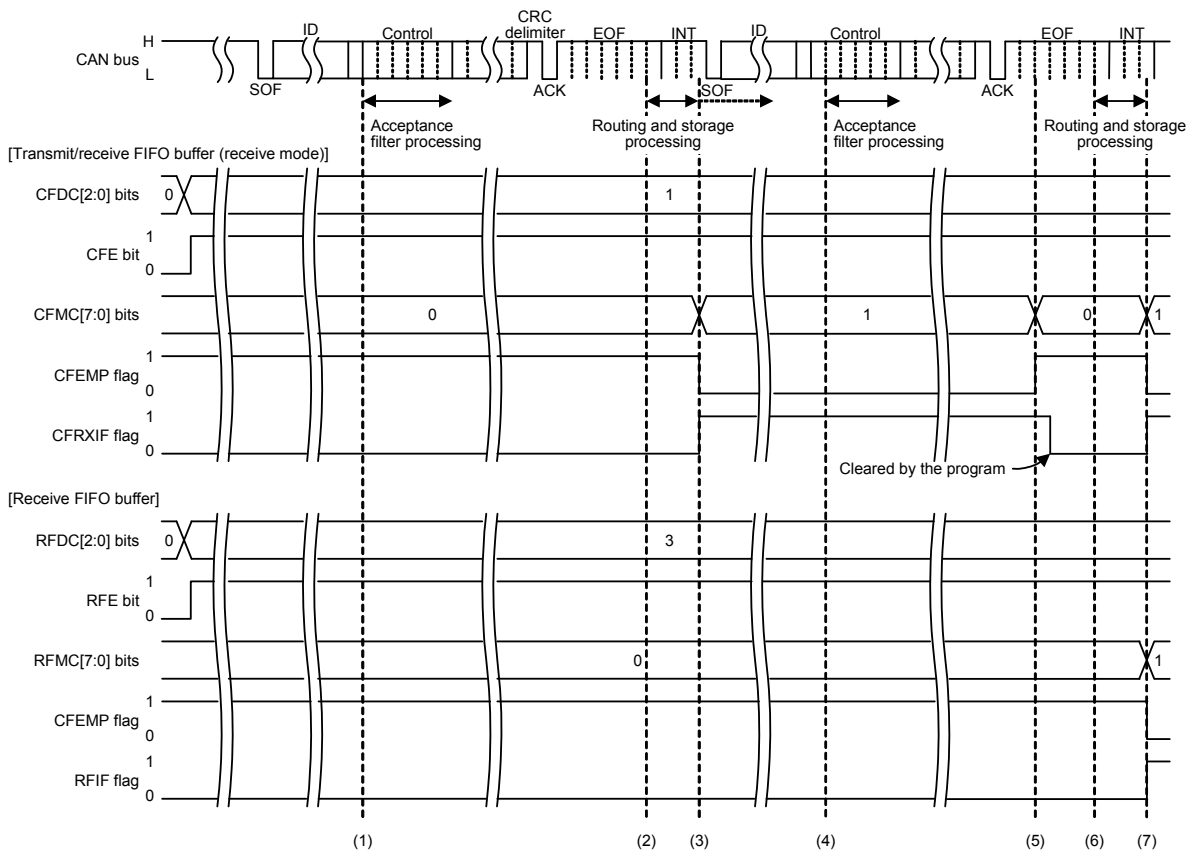


Figure 20.25 FIFO Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing and the CFE bit in the RSCAN0CFCCk register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] bit value in the RSCAN0CFCCk register is 001_B or more, the message is stored in the transmit/receive FIFO buffer that is set to receive mode. The CFMC[7:0] bits in the RSCAN0CFSTSk register are incremented and becomes 01_H. When the CFIM bit in the RSCAN0CFCCk register is set to 1 (a FIFO receive interrupt request is generated each time a message has been received), the CFRXIF flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). The CFRXIF flag can be reset to 0 by the program.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) Read received messages from the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers and write FF_H to the RSCAN0CFPCTRk register. This causes the

CFMC[7:0] bits in the RSCAN0CFSTSk register to be decremented. When CFMC[7:0] bits become 00_H, the CFEMP flag in the RSCAN0CFSTSk register becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

(6) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(7) The message is stored in the transmit/receive FIFO buffer set in receive mode when the message has passed through the DLC filter process if the CFE bit is set to 1 (transmit/receive FIFO buffers are used), and the CFDC[2:0] bit value is set to 001_B or more. The CFMC[7:0] bit value is incremented by 1 to be 01_H. When the CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFRXIF flag is set to 1 (a transmit/receive FIFO receive interrupt request is present).
The message is stored in the receive FIFO buffer if the RFE bit in the RSCAN0RFCCx register is set to 1 (receive FIFO buffers are used), and the RFDC[2:0] bit value in the RSCAN0RFCCx register is set to 001_B or more. The RFMC[7:0] bit value in the RSCAN0RFSTSx register is set to 01_H by being incremented by 1. When the RFIM bit in the RSCAN0RFCCx register is set to 1 (an interrupt occurs each time a message has been received), the RFIF flag in the RSCAN0RFSTSx register is set to 1 (a receive FIFO interrupt request is present).

20.10.3 Transmission Procedure

20.10.3.1 Procedure for Transmission from Transmit Buffers

Figure 20.26 shows the procedure for transmission from transmit buffers.

Figure 20.27 shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmission has been successfully completed. **Figure 20.28** shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmit abort has been completed.

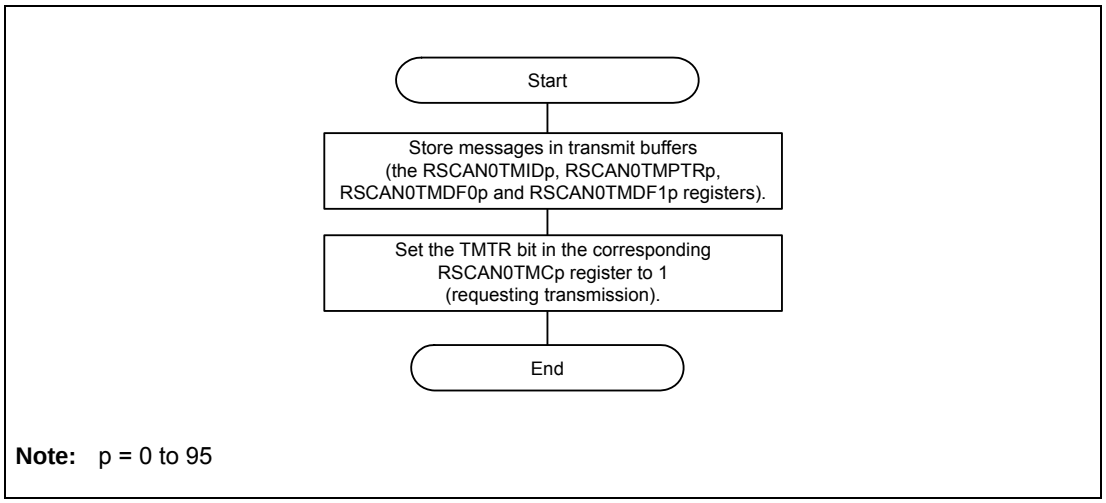
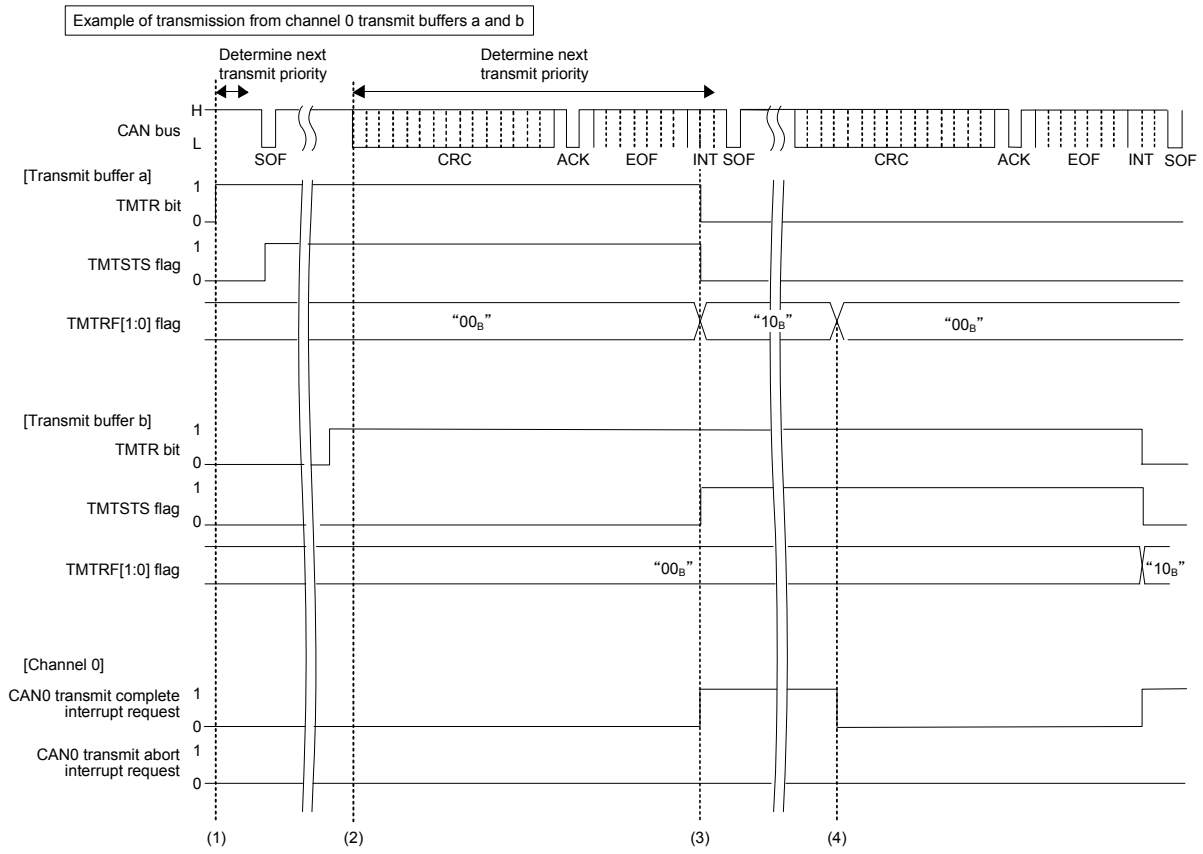


Figure 20.26 Procedure for Transmission from Transmit Buffers



Note: a = 0 to 95, b = 0 to 95
TMTR: Bit in the RSCAN0TMCp register
TMTSTS, TMTRF[1:0]: Flags in the RSCAN0TMSTSp registers

Figure 20.27 Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)

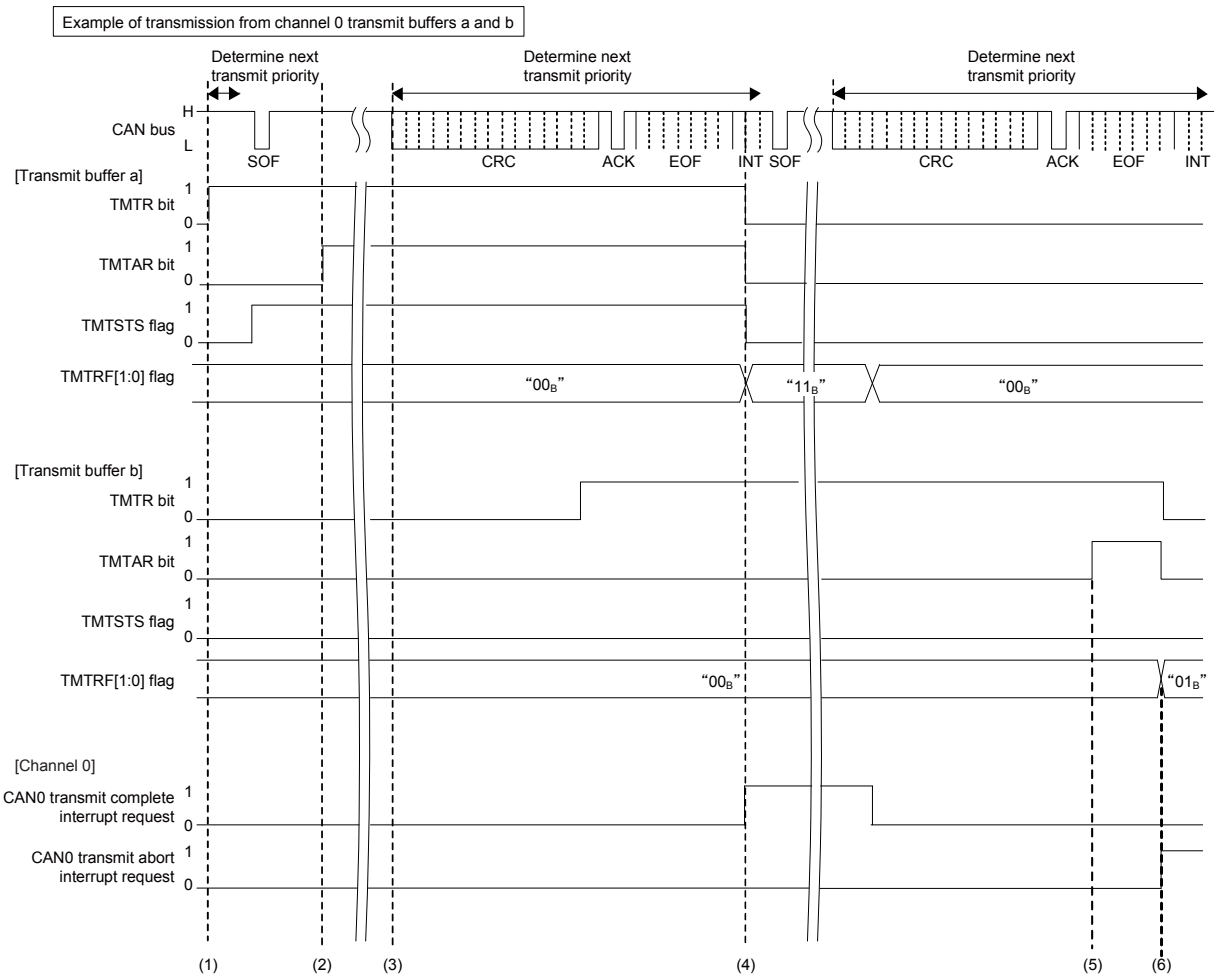
(1) When the TMTR bit in the RSCAN0TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCAN0TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When a transmit request from a buffer is present, the priority determination starts at the first bit of CRC field for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(3) When transmission completes successfully, the TMTRF[1:0] flag in the RSCAN0TMSTSa register is set to 10_B (transmission has been completed (without transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCAN0TMCa register are cleared to 0. When the TMIEa bit in the RSCAN0TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

(4) Before starting the next transmission, set the TMTRF[1:0] flag to 00_B. Write the next message to the transmit buffer, and then set the TMTR bit to 1 (transmission is requested). The TMTR bit can be set to 1 only when the TMTRF[1:0] flag is set to 00_B.

If an arbitration-lost has occurred after transmission is started, the TMTSTS flag is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC field to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.



Note: a = 0 to 95, b = 0 to 95
TMTR, TMTAR: Bits in the RSCAN0TMCp register
TMTSTS, TMTRF[1:0]: Flags in the RSCAN0TMSTSp register

Figure 20.28 Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) When the TMTR bit in the RSCAN0TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCAN0TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When it is determined that the transmit buffer is used for the next transmission or transmission is in progress, message transmission is not aborted unless an error or arbitration loss occurs even if the TMTAR bit is set to 1 (transmit abort is requested).

(3) The priority determination starts at the first bit of the CRC field for the next transmission. In this timing chart, buffer b is not selected as the next transmit buffer. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(4) When transmission completes successfully, the TMTRF[1:0] flag in the RSCAN0TMSTSa register is set to 11_B (transmission has been completed (with transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCAN0TMCa register are cleared to 0. When the TMIEa value in the RSCAN0TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

(5) While another CAN node is transmitting data on the CAN bus (TMTSTS flag = 0), if the TMTAR bit is set to 1 while the corresponding channel is determining transmit priority, the TMTR bit cannot be cleared to 0.

(6) After the internal processing time has passed, the transmission is terminated and the TMTRF[1:0] flag is set to 01_B. When the transmit buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not being made, an abort request is immediately accepted and the TMTRF[1:0] flag is set to 01_B. At this time, the TMTR and TMTAR bits are cleared to 0. When transmit abort is completed with the TAIE bit in the RSCAN0CmCTR register set to 1 (transmit abort interrupt is enabled), an interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flags to 00_B.

If an arbitration loss has occurred after the CAN channel started transmission, the TMTSTS bit is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC field to find the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

20.10.3.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

Figure 20.29 shows the procedure for transmission from transmit/receive FIFO buffers.

Figure 20.30 shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmission has been successfully completed. **Figure 20.31** shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmit abort has been completed.

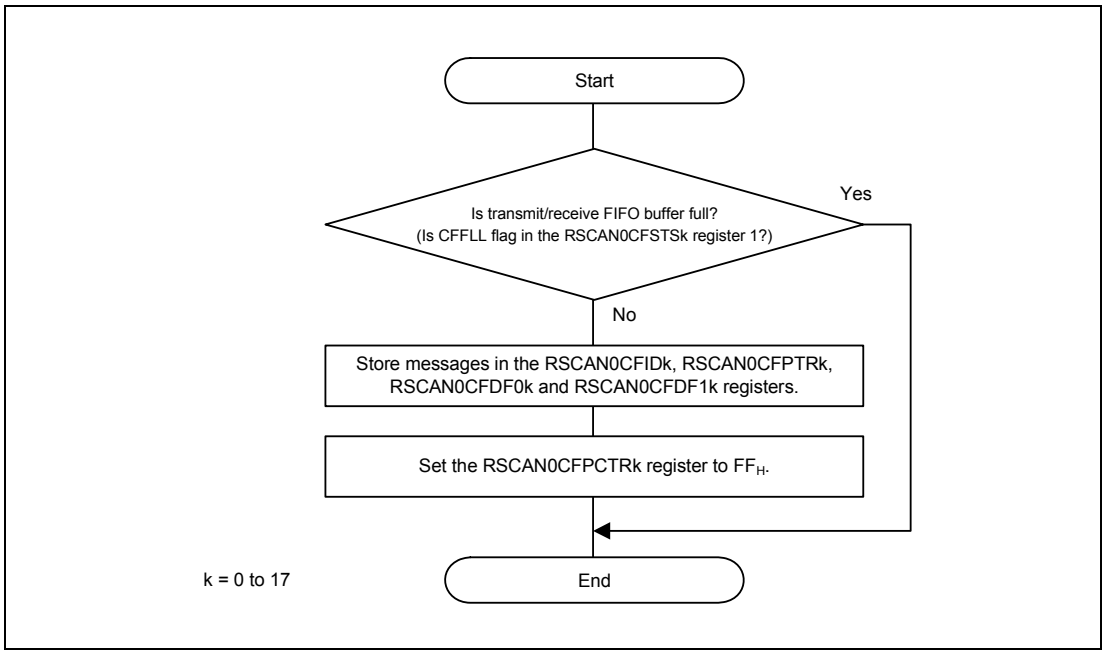


Figure 20.29 Procedure for Transmission from Transmit/Receive FIFO Buffers

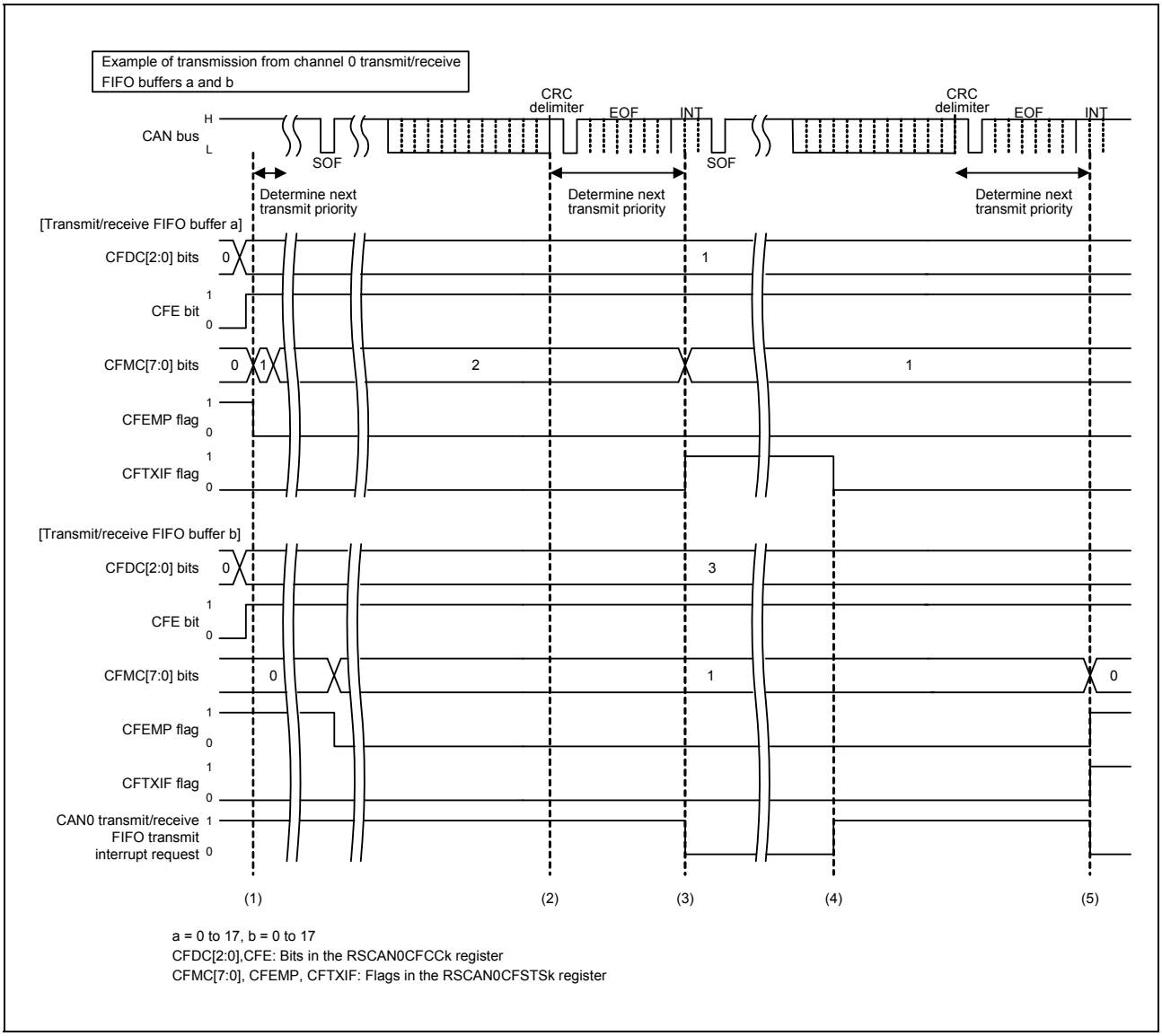


Figure 20.30 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)

(1) While the CAN bus is idle, when the CFE bit in the RSCA0CFCCa register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] bits in the RSCAN0CFCCa register are set to 001_B (4 messages) or more and the CFMC[7:0] bit value in the RSCAN0CFSTSa register is 01_H or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer a of channel 0.

(2) When a transmit request from a buffer is present, the priority determination starts at the first bit of CRC field for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(3) When transmission completes successfully, the CFMC[7:0] bits in the RSCAN0CFSTSa register are decremented. Setting the CFIM bit in the RSCAN0CFCCa register to 1 (a FIFO transmit

interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RSCAN0CFSTSk register to 1 (a transmit/receive FIFO transmit interrupt request is present).

(4) The program can clear the CFTXIF flag.

(5) Message transmission from transmit/receive FIFO buffer b of channel 0 has been completed and the CFMC[7:0] bits in the RSCAN0CFSTSb register are decremented. The CFMC[7:0] bits are cleared to 00_H and therefore the CFEMP flag in the RSCAN0CFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

Transmission is continued until the CFEMP flag is set to 1. It is possible to continuously store transmit messages in FIFO buffers until the CFFLL flag in the RSCAN0CFSTSa and RSCAN0CFSTSb register is set to 1 (the transmit/receive FIFO buffer is full).

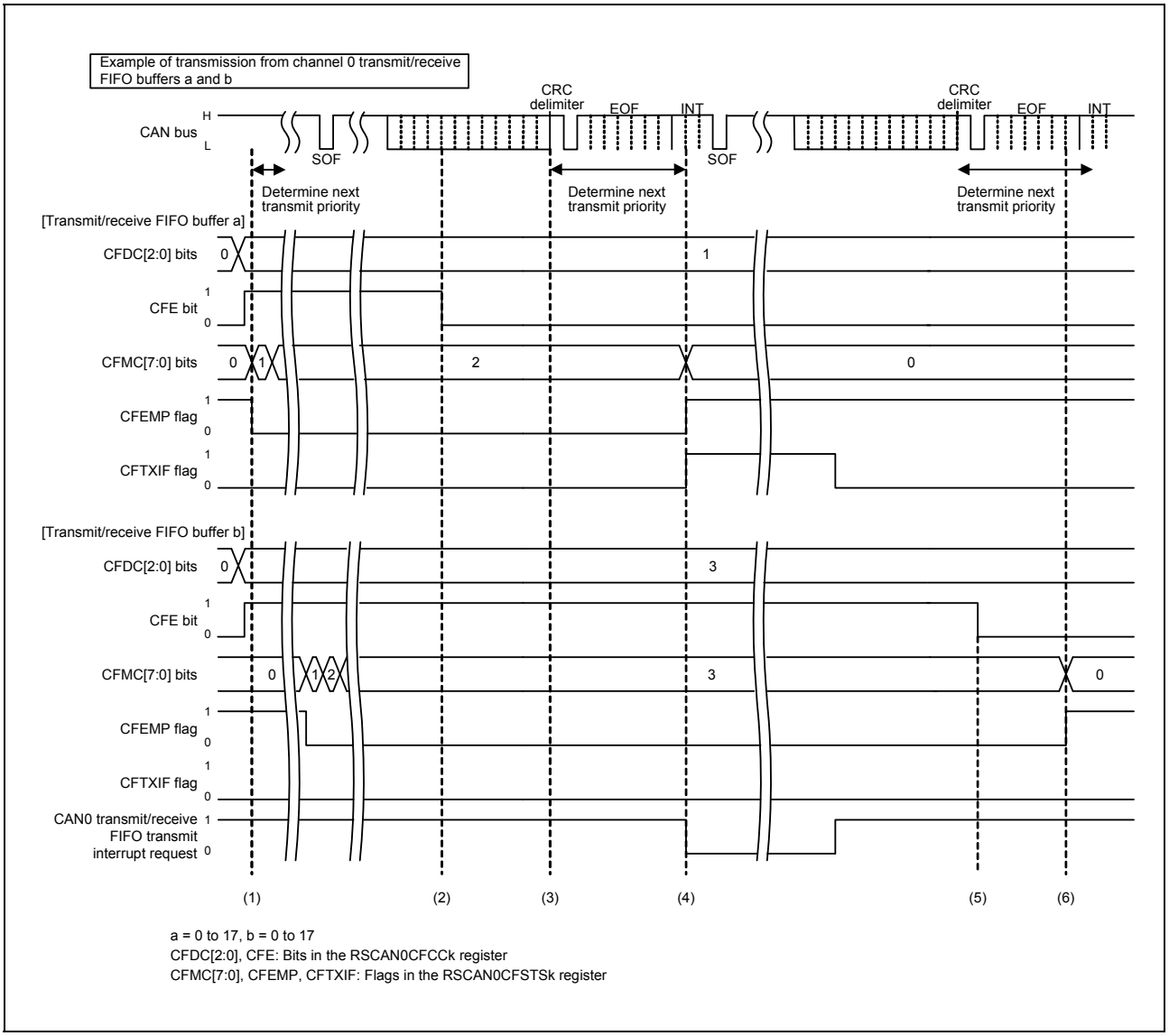


Figure 20.31 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) While the CAN bus is idle, when the CFE bit in the RSCAN0CFCCa register (a = 0 to 17) is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] bits in the RSCAN0CFCCa register are set to 001_B (4 messages) or more and the CFMC[7:0] bit value in the RSCAN0CFSTSa register is 01_H or more, the priority determination processing starts to determine the highest-priority

transmit message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer a of channel 0.

(2) When transmission is in progress or it is determined that the transmit/receive FIFO buffer is used for the next transmission, message transmission is not aborted unless an error or arbitration loss occurs even if the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

(3) When a transmit request from a buffer is present, the priority determination starts at the first bit of CRC field for the next transmission. In this figure, transmit/receive FIFO buffer b is not selected as a buffer for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(4) When transmit completes successfully, the CFMC[7:0] bit value is cleared to 00_H. Setting the CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RSCAN0CFSTSa register to 1 (a transmit/receive FIFO transmit interrupt request is present). The program can clear the CFTXIF flag.

(5) If another CAN node on the CAN bus is transmitting data (not from transmit/receive FIFO buffer b), transmit/receive FIFO buffers cannot be disabled immediately even if the CFE bit in the RSCAN0CFCCb register is cleared to 0 (no transmit/receive FIFO buffer is used) during transmit priority determination. (The CFEMP flag in the RSCAN0CFSTSb register is not set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) immediately.)

(6) After the internal processing time has passed, transmit/receive FIFO buffers are disabled and the CFMC[7:0] bits in the RSCAN0CFSTSb register are cleared to 00_H and the CFEMP flag is set to 1. When the transmit/receive FIFO buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not in progress, the transmit/receive FIFO buffer is immediately disabled. (The CFMC[7:0] bits are cleared to 00_H and the CFEMP flag is set to 1.)

20.10.3.3 Procedure for Transmission from the Transmit Queue

Figure 20.32 shows the procedure for transmission from the transmit queue.

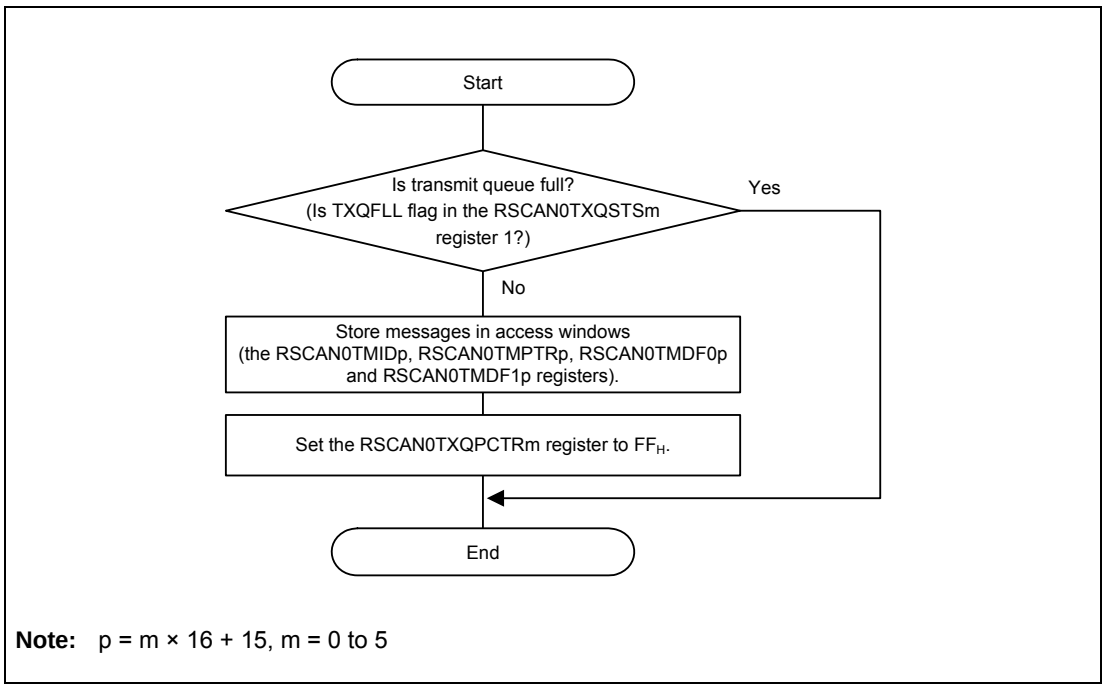


Figure 20.32 Procedure for Transmission from the Transmit Queue

20.10.3.4 Transmit History Buffer Reading Procedure

Transmit history data can be read from the RSCAN0THLACCm register. The next data can be accessed by writing FF_H to the corresponding RSCAN0THLPCTRm register (m = 0 to 5) after reading a set of data. **Figure 20.33** shows the transmit history buffer reading procedure.

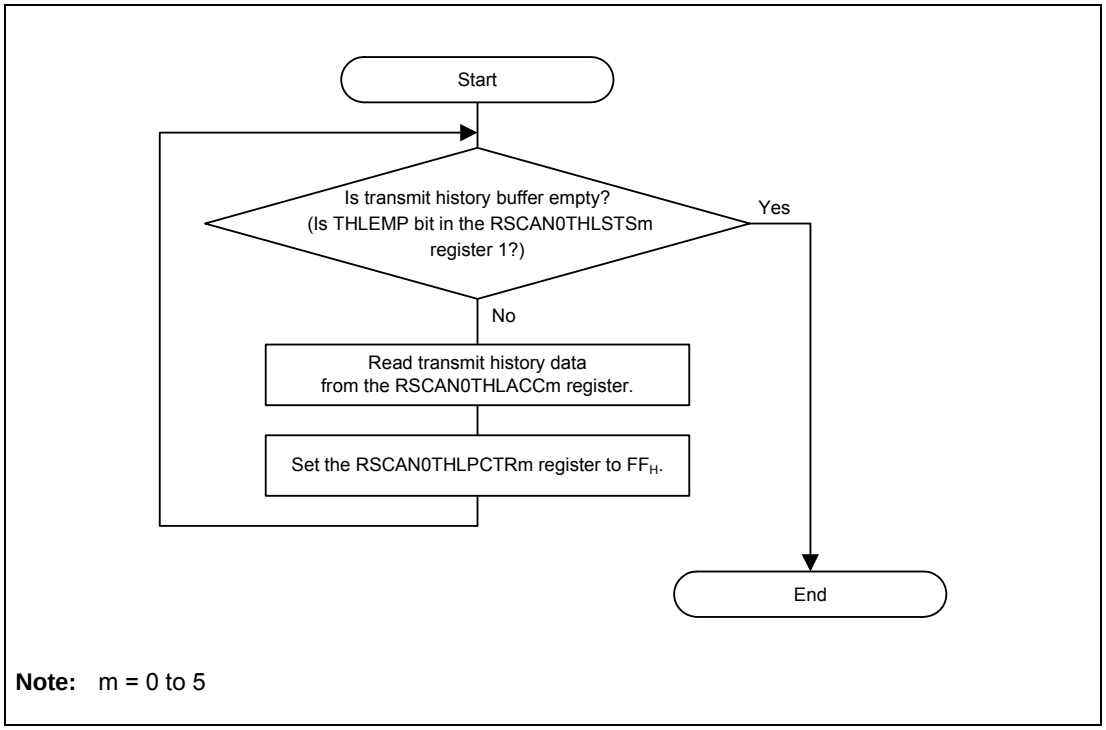


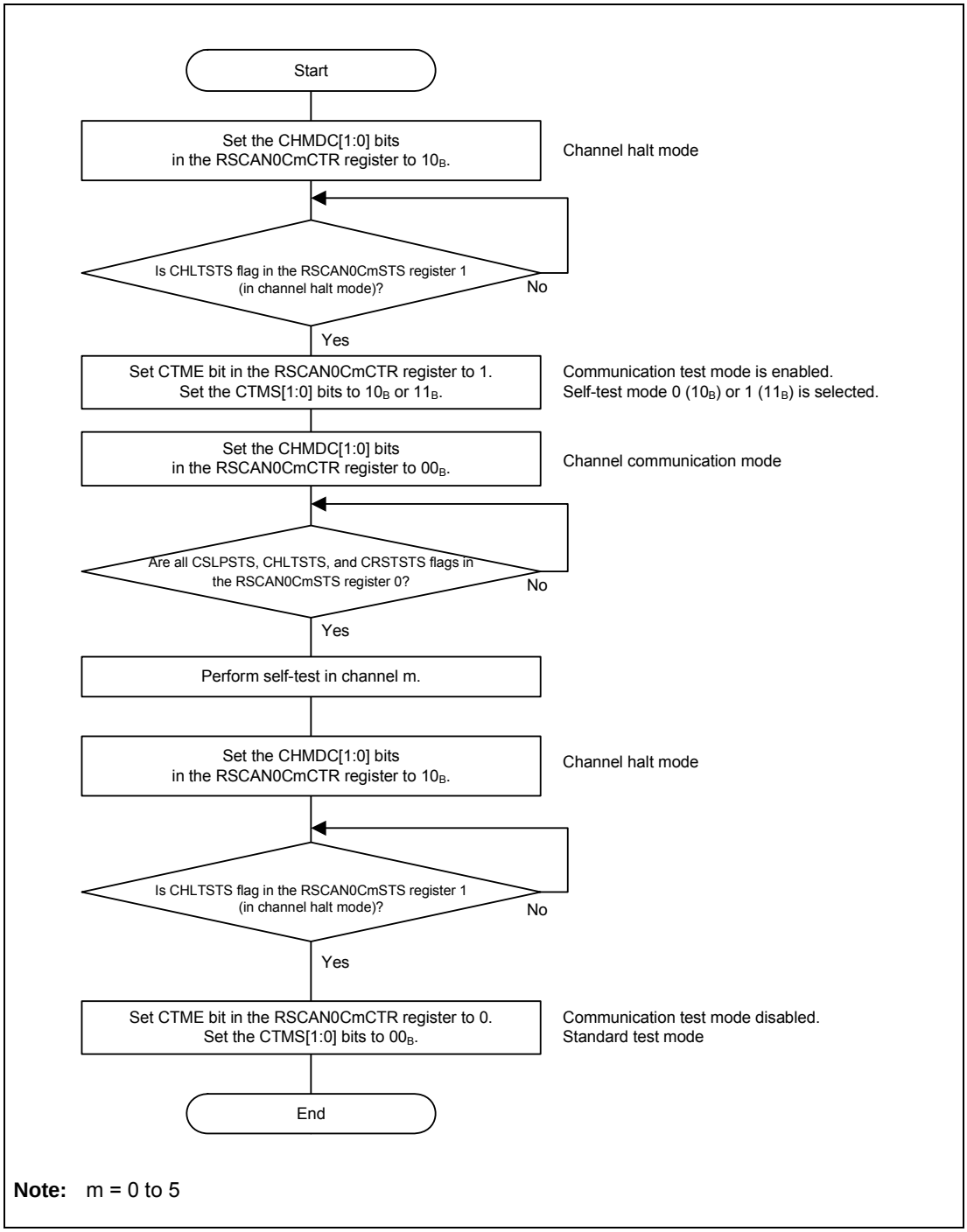
Figure 20.33 Transmit History Buffer Reading Procedure

20.10.4 Test Settings

20.10.4.1 Self-Test Mode Setting Procedure

Self-test mode allows communication test on a channel basis by enabling a CAN node to receive its own transmitted messages.

Figure 20.34 shows the self-test mode setting procedure.



Note: m = 0 to 5

Figure 20.34 Self-Test Mode Setting Procedure

20.10.4.2 Procedure for Releasing the Protection

Since the global test function in **Table 20.98** is protected, write the protection release data 1 and release data 2 in succession to the LOCK[15:0] bits in the RSCAN0GLOCKK register, then set the target test bit to 1.

Table 20.98 Protection Release Data for Test Function

Test Function	Protection Release Data 1	Protection Release Data 2	Target Bit
RAM test	7575_H	$8A8A_H$	RTME bit in the RSCAN0GTSTCTR register

If an incorrect value is written to the LOCK[15:0] bits, restart from writing the protection release data 1. **Figure 20.35** shows the procedure for releasing the protection.

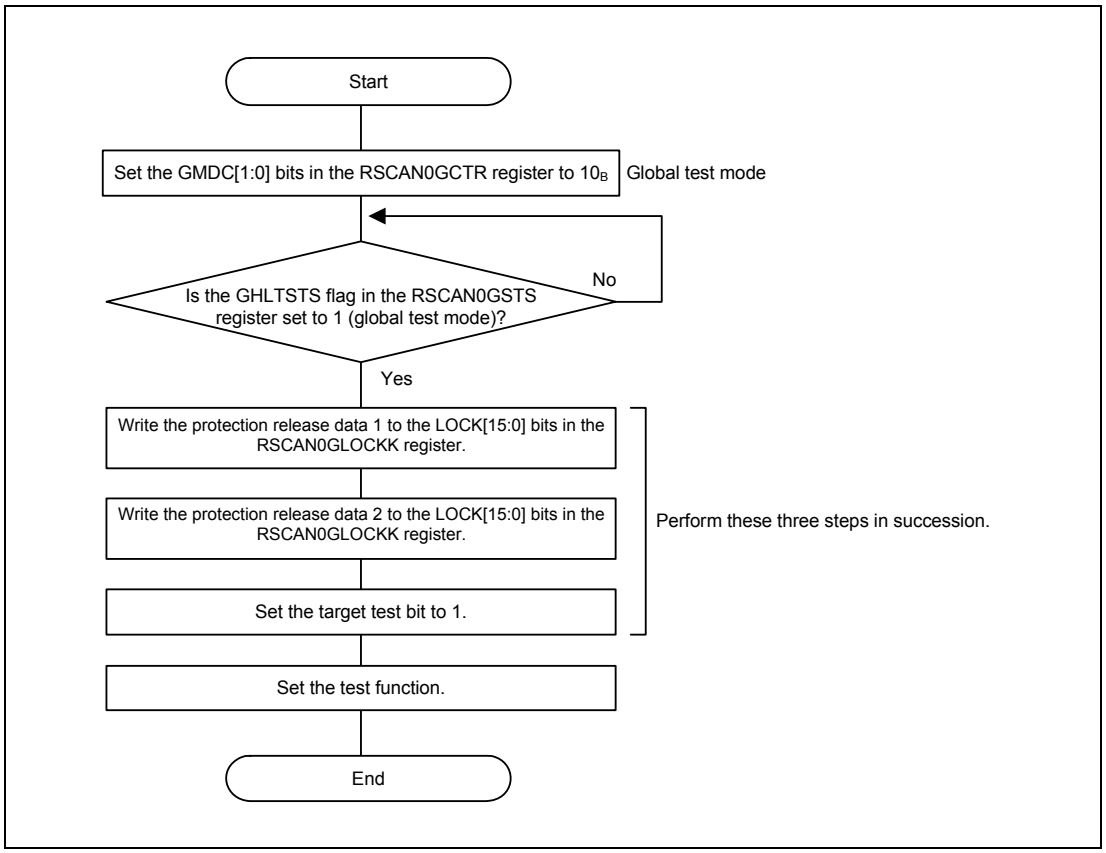


Figure 20.35 Protection Release Procedure

20.10.4.3 RAM Test Setting Procedure

RAM tests include CAN RAM read/write test. The read/write test verifies that data written to the RAM is read correctly. Before closing the RAM test, write 0000 0000_H to all pages of the CAN RAM.

Figure 20.36 shows the RAM test setting procedure.

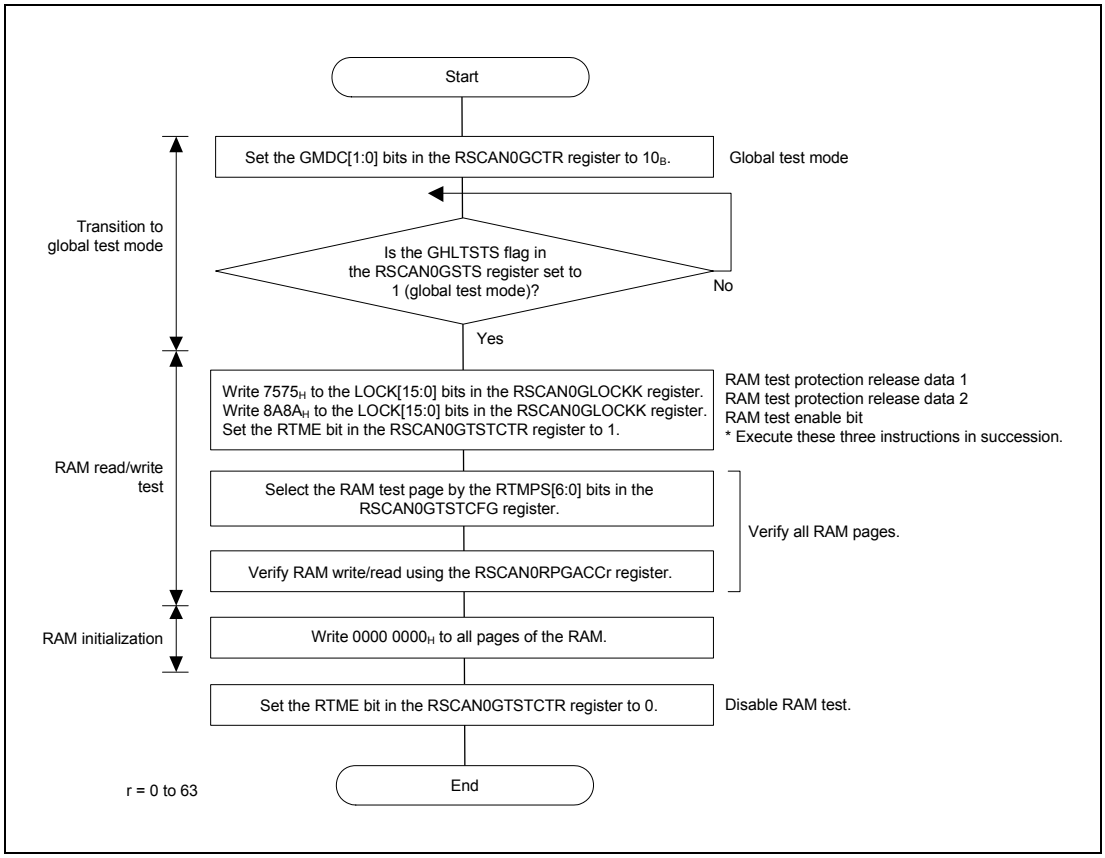
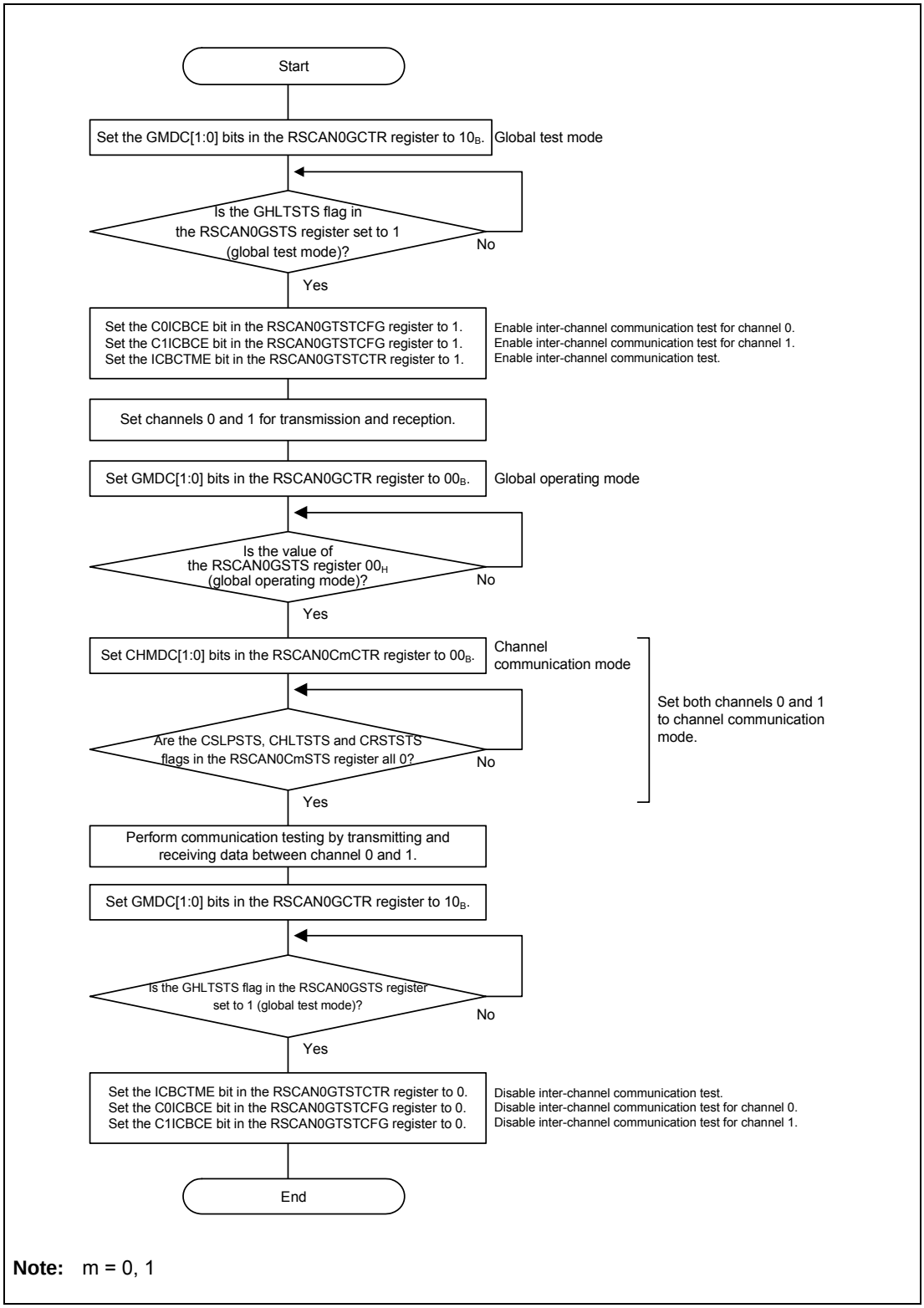


Figure 20.36 RAM Test Setting Procedure

20.10.4.4 Inter-Channel Communication Test Setting Procedure

Communication testing can be performed by transmitting and receiving data between different channels.

Figure 20.37 shows the inter-channel communication test setting procedure.



Note: m = 0, 1

Figure 20.37 Inter-Channel Communication Test Setting Procedure (Example of Communication Test between Channel 0 and Channel 1)

20.11 Detection and Correction of Errors in RS-CAN RAM

20.11.1 ECC for the RSCAN0 RAM

Table 20.99 gives an outline of the ECC functions for the RSCAN0 RAM.

Table 20.99 List of the ECC Functions for the RSCAN0 RAM

Item	Outline of Functions
ECC error detection/ correction	The RAM is checked for ECC errors. The following options are selectable. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection The ECC error detection/correction can be disabled by using through mode. With the initial settings, error detection/correction is enabled.
Error notification	When an ECC 2-bit error is generated, the error is notified. • Error notification can be enabled or disabled when an ECC 2-bit error is detected. In the initial setting, 2-bit error notification is enabled. However, when the interrupt is masked by the FEINTFMSK register, interrupt processing is not performed.
Error status	Monitoring for the detection of two-bit ECC errors and for the detection of one-bit ECC errors is available. A register for clearing the error status is provided.

CAUTION

When ECC error detection/correction is performed, confirm initialization of the RSCANn RAM (RSCANnGSTS.GRAMINIT = 0) before it is used.

20.11.2 Interrupt Request

Table 20.100 lists the ECC interrupt request of RSCAN0 RAM.

Table 20.100 RSCANn ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal Name	Description	Name	DMA Trigger Number
—	RSCANn ECC 2 bit error interrupt	INTECCDCNRAM0	—

20.11.3 List of Registers

RSCAN0 registers are listed in the table below.

ECCCAN00 is an ECC register for the message buffer RAM and ECCCAN01 is an ECC register for the reception rule table RAM.

ECCCAN0 includes the registers of both ECCCAN00 and ECCCAN01.

Table 20.101 Registers

Module	Register	Symbol	Address
ECCCAN0	RSCAN0 ECC control register 0	ECCRCAN0CTL	FFC7 1000$_H$
ECCCAN0	RSCAN0 ECC test mode control register 0	ECCRCAN0TMC	FFC7 1004$_H$
ECCCAN0	RSCAN0 ECC encode/decode input/output replacement test register 0	ECCRCAN0TED	FFC7 100C$_H$
ECCCAN0	RSCAN0 ECC redundant bit data control test register 0	ECCRCAN0TRC	FFC7 1008$_H$
ECCCAN0	RSCAN0 ECC redundant bit input/output replacement Buffer register 0	ECCRCAN0ERDB	FFC7 1008$_H$
ECCCAN0	RSCAN0 ECC encode test register 0	ECCRCAN0ECRD	FFC7 1009$_H$
ECCCAN0	RSCAN0 ECC 7-bit redundant bit data hold test register 0	ECCRCAN0HORD	FFC7 100A$_H$
ECCCAN0	RSCAN0 ECC decode syndrome data register 0	ECCRCAN0SYND	FFC7 100B$_H$
ECCCAN00	RSCAN0 ECC control register 00	ECCRCAN00CTL	FFC7 1040$_H$
ECCCAN00	RSCAN0 ECC test mode control register 00	ECCRCAN00TMC	FFC7 1044$_H$
ECCCAN00	RSCAN0 ECC encode/decode input/output replacement test register 00	ECCRCAN00TED	FFC7 104C$_H$
ECCCAN00	RSCAN0 ECC redundant bit data control test register 00	ECCRCAN00TRC	FFC7 1048$_H$
ECCCAN00	RSCAN0 ECC redundant bit input/output replacement Buffer register 00	ECCRCAN00ERDB	FFC7 1048$_H$
ECCCAN00	RSCAN0 ECC encode test register 00	ECCRCAN00ECRD	FFC7 1049$_H$
ECCCAN00	RSCAN0 ECC 7-bit redundant bit data hold test register 00	ECCRCAN00HORD	FFC7 104A$_H$
ECCCAN00	RSCAN0 ECC decode syndrome data register 00	ECCRCAN00SYND	FFC7 104B$_H$
ECCCAN01	RSCAN0 ECC control register 01	ECCRCAN01CTL	FFC7 1050$_H$
ECCCAN01	RSCAN0 ECC test mode control register 01	ECCRCAN01TMC	FFC7 1054$_H$
ECCCAN01	RSCAN0 ECC encode/decode input/output replacement test register 01	ECCRCAN01TED	FFC7 105C$_H$
ECCCAN01	RSCAN0 ECC redundant bit data control test register 01	ECCRCAN01TRC	FFC7 1058$_H$
ECCCAN01	RSCAN0 ECC redundant bit input/output replacement Buffer register 01	ECCRCAN01ERDB	FFC7 1058$_H$
ECCCAN01	RSCAN0 ECC encode test register 01	ECCRCAN01ECRD	FFC7 1059$_H$
ECCCAN01	RSCAN0 ECC 7-bit redundant bit data hold test register 01	ECCRCAN01HORD	FFC7 105A$_H$
ECCCAN01	RSCAN0 ECC decode syndrome data register 01	ECCRCAN01SYND	FFC7 105B$_H$

- When ECCCAN0 is written, the same value can be written in both ECCCAN00 and ECCCAN01 registers.
- When ECC error is detected, error status can be read by ECCCAN0 register.
- When ECC error is not detected, ECCCAN00 registers can be read by ECCCAN0 registers.
- When more than twice of error has occurred to one of ECCCAN00 or ECCCAN01, latest error status can be read by ECCCAN0 register.
- When ECC errors are detected by both ECCCAN00 and ECCCAN01 registers, ECCCAN00 error status can be read by ECCCAN0 register.
- Each ECCCAN00 and ECCCAN01 can be read or written directly.

20.11.4 ECCRCANzCTL — RSCAN0 ECC Control Register (z = 0/00/01)

The ECCRCANzCTL register controls the mode of the ECC and the status for RSCAN0.

Bits 7, 5 and 4 should be set (written) while the RSCAN0 operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be 01_B.

Access: This register can be read or written in 16-bit units.

Address: See **Table 20.101, Registers**.

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	EMCA0	—	—	—	ECER2C	ECER1C	—	ECTHM	—	EC1ECP	EC2EDIC	—	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	—
R/W	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R	R/W	R	R/W	R/W	R	R	R	R

Note 1. These bits are always read as 0.

Table 20.102 ECCRCANzCTL Register Contents (1/2)

Bit position	Bit Name	Function
15	EMCA1	Access control bits 1 and 0 to ECC mode selection bit These bits specify whether modifying the ECTHM bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 01_B, writing to bit 7 is enabled.
14	EMCA0	
13 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	ECER2C	2-bit ECC error detection flag clear bit This bit is used to clear the 2 bit error detection flag of ECER2F (bit 2). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between this bit writing and ECER2F bit setting occurs, writing to this bit has a priority.
9	ECER1C	1-bit ECC error correction accumulation flag clear bit This bit is used to clear the 1 bit error detection/correction flag of ECER1F (bit 1). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between this bit writing and ECER1F bit setting occurs, writing to this bit has a priority.
8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTHM	ECC function through mode selection bit This bit is used to set enabling and disabling of ECC. Setting this bit to 1 disables ECC function. When writing to this bit, (0, 1) must be written to (EMCA1, EMCA0) at the same time. 0: Through mode is disabled (normal operation mode). 1: Through mode is enabled. (ECC function disable)
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-bit error correction enable bit This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.

Table 20.102 ECCRCANzCTL Register Contents (2/2)

Bit position	Bit Name	Function
4	EC2EDIC	2-bit error detection interrupt control bit This bit controls whether to generate an interrupt when 2-bit error is detected. 0: When 2-bit error is detected, a INTECCDCNRAM0 interrupt will not be generated. 1: When 2-bit error is detected, a INTECCDCNRAM0 interrupt will be generated. (initial value)
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ECER2F	2-bit error detection flag bit This flag indicates whether 2-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). When 2-bit error interrupt is enabled (EC2EDIC = 1) and this flag is set, an ECC 2-bit error interrupt (INTECCDCNRAM0) is output. Write 1 to the ECER2C bit (bit 10) to clear the flag. This bit is cleared at the time of through mode enable selection (ECTHM = 1). If 2-bit error is detected again while this bit is set, an interrupt signal will not be generated. 0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred.
1	ECER1F	1-bit error detection/correction flag bit This flag indicates whether 1-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). Write 1 to the ECER1C bit (bit 9) to clear the flag. This bit is cleared at the time of through mode enable selection (ECTHM = 1). 0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred.
0	ECEMF	ECC error message flag This flag indicates whether an error exists in the current read data. This bit is updated whenever the RAM reads data. This bit might be set if it is read before initialization of the RAM. This bit is cleared at the time of through mode enable selection (ECTHM = 1) and when there is no 1 bit error in the decode circuit input data. 0: The currently-read RAM data does not have bit errors. 1: The currently-read RAM data have bit errors.

CAUTION

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.

We recommend initializing the RAM before clearing bits 2 and 1.

20.11.5 ECCRCANzTMC — RSCAN0 ECC Test Mode Control Register (z = 0/00/01)

The ECCRCANzTMC register switches to and controls the test mode.

This register can be used when RS-CAN is not accessed to RAM.

Access: This register can be read or written in 16-bit units.

Address: See **Table 20.101, Registers**.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA1	ETMA0	—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are always read as 0.

Table 20.103 ECCRCANzTMC Register Contents (1/2)

Bit position	Bit Name	Function
15	ETMA1	Access control bits 1 and 0 to ECC test mode bit These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 10_B, writing to bit 7 is enabled.
14	ETMA0	
13 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTMCE	ECC test mode enable bit This bit specifies whether to enable access to the test registers and test control bits. When writing to this bit, (1, 0) should be written to (ETMA1, ETMA0) at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. Test registers: ECCRCANzTED, ECCRCANzTRC, ECCRCANzSYND, ECCRCANzHORD, ECCRCANzECRD, ECCRCANzERDB Register test control bit: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	ECTRRS	ECC RAM read test mode selection bit Select reading destination when reading ECCRCANzTED register and reading destination when reading ECCRCANzERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: The read value of the ECCRCANzTED register is the write value of the ECCRCANzTED register. The read value of the ECCRCANzERDB register is the write value of the ECCRCANzERDB register. 1: The read value of the ECCRCANzTED register can read RAM data. The read value of the ECCRCANzERDB register is the ECC data to be written to RAM.
3	ECREOS	ECC redundant bit output data selection bit This bit is used to identify and select the ECC data to be stored in RAM as ECC data generated for write data or the value of the ECCRCANzERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Store ECC data generated for write data to RAM. 1: Store the value of ECCRCANzERDB register to RAM.

Table 20.103 ECCRCANzTMC Register Contents (2/2)

Bit position	Bit Name	Function
2	ECENS	ECC encoder input selection bit This bit is used to identify the targeted data for ECC data generation as write data to RAM or the value of the ECCRCANzTED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate ECC data from the write data to RAM. 1: Generate ECC data from the value of the ECCRCANzTED register.
1	ECDCS	ECC decoder input selection bit This bit is used to identify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the ECCRCANzTED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from RAM data and detect errors. 1: Generate syndrome code from the value of the ECCRCANzTED register and detect errors.
0	ECREIS	ECC redundant bit input data selection bit This bit is used to identify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the ECCRCANzERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from ECC data stored in RAM and detect errors. 1: Generate syndrome code from the value of the ECCRCANzERDB register and detect errors.

20.11.6 ECCRCANzTED — RSCAN0 ECC Encode/Decode Input/Output Replacement Test Register (z = 0/00/01)

In ECC test mode, this register handles test data.

The value of the register can be used to generate ECC data or syndrome code.

When ECC test mode is enabled (ECCRCANzTMC.ECTMCE = 1), it is accessible. When ECCRCANzTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when RS-CAN is not accessed to RAM.

Access: This register can be read or written in 32-bit units.

Address: See **Table 20.101, Registers**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.104 ECCRCANzTED Register Contents

Bit position	Bit Name	Function
31 to 0	ECEDB[31:0]	When ECCRCANzTMC.ECENS = 1, the value of this register is used to generate ECC data and the value this register to RAM is stored to RAM. When ECCRCANzTMC.ECDCS = 1, the value of this register is used to generate syndrome code and the value of this register is stored in ECC decode syndrome data register (ECCRCANzSYND). In addition, when ECCRCANzTMC.ECTRRS = 1, RAM data [31:0], instead of written data, is read for the value of this register.

20.11.7 ECCRCANzTRC — RSCAN0 ECC Redundant Bit Data Control Test Register (z = 0/00/01)

In ECC test mode, this test register, for ECC data, consists of four 8-bit registers, ECCRCANzSYND, ECCRCANzHORD, ECCRCANzECRD, and ECCRCANzERDB.

When ECC test mode is enabled (ECCRCANzTMC.ECTMCE = 1), this register can be accessed. When ECCRCANzTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when RS-CAN is not accessed to RAM.

Access: This register can be read or written in 32-bit units.

Address: See **Table 20.101, Registers**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECCRCANzSYND (See **Section 20.11.8**)								ECCRCANzHORD (See **Section 20.11.9**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECCRCANzECRD (See **Section 20.11.10**)								ECCRCANzERDB (See **Section 20.11.11**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

20.11.8 ECCRCANzSYND — RSCAN0 ECC Decode Syndrome Data Register (z = 0/00/01)

In ECC test mode, this is a read-only register for storing generated syndrome code.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCANzTMC.ECTMCE = 1), this register can be accessed. When ECC test mode is disabled (ECCRCANzTMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 20.101, Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SYND6	SYND5	SYND4	SYND3	SYND2	SYND1	SYND0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 20.105 ECCRCANzSYND Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	SYND[6:0]	The generated syndrome code is stored as needed.

20.11.9 ECCRCANzHORD — RSCAN0 ECC 7-Bit Redundant Bit Data Hold Test Register (z = 0/00/01)

In ECC test mode, this register is used to store ECC data for read RAM data.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCANzTMC.ECTMCE = 1), this register can be accessed. When ECC test mode is disabled (ECCRCANzTMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 20.101, Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	HORD6	HORD5	HORD4	HORD3	HORD2	HORD1	HORD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 20.106 ECCRCANzHORD Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	HORD[6:0]	ECC code for read RAM data is stored as needed. When ECCRCANzTMC.ECTRRS = 1 and ECCRCANzTED register is read, ECC code is stored.

20.11.10 ECCRCANzECRD — RSCAN0 ECC Encode Test Register (z = 0/00/01)

In ECC test mode, this is a read-only register for storing ECC data generated for write RAM data.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCANzTMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCRCANzTMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 20.101, Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ECRD6	ECRD5	ECRD4	ECRD3	ECRD2	ECRD1	ECRD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 20.107 ECCRCANzECRD Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	ECRD[6:0]	These bits can read ECC data generated at the time of RAM data writing, and can read ECC data for the data written to the ECCRCANzTED register when ECCRCANzTMC.ECENS = 1.

20.11.11 ECCRCANzERDB — RSCAN0 ECC Redundant Bit Input/Output Replacement Buffer Register (z = 0/00/01)

In ECC test mode, this register handles ECC data.

The value of this register can be used as ECC data generated at the time of RAM writing or ECC data read at the time of RAM data reading.

When ECC test mode is enabled (ECCRCANzTMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCRCANzTMC.ECTMCE = 0), 00_H is read.

Access: This register can be read or written in 8-bit units.

Address: See **Table 20.101, Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ERDB6	ERDB5	ERDB4	ERDB3	ERDB2	ERDB1	ERDB0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.108 ECCRCANzERDB Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ERDB[6:0]	When ECCRCANzTMC.ECREOS = 1, these bits store the value of this register as ECC data to RAM. When ECCRCANzTMC.ECREIS = 1, the value of this register is read as ECC data read from RAM. When ECCRCANzTMC.ECTRRS = 1, ECC data to be stored in RAM is read instead of the write data for the read value of this register.

20.11.12 SELB_READTEST — ECCREAD Test Select Register

SELB_READTEST is used to check read/write access to the RSCAN0 ECC registers. For detail, see **Section 16.7.11, SELB_READTEST — ECCREAD Test Select Register**.

20.12 Notes on the RS-CAN Module

- When changing a global mode, check the GSLPSTS, GHLTSTS, and GRSTSTS flags in the RSCAN0GSTS register for transitions. When changing a channel mode, check the CSLPSTS, CHLTSTS, and CRSTSTS flags in the RSCAN0CmSTS register (m = 0 to 5) for transitions.
- The acceptance filter processing checks receive rules sequentially in ascending order from the minimum rule number. If the same ID, IDE bit, or RTR bit value is set for multiple receive rules, the minimum number of receive rule is used for the acceptance filter processing. If the message does not pass through the subsequent DLC filter processing, the data processing is terminated without returning to the acceptance filter processing and the message is not stored in the buffer.
- When linking transmit buffers to transmit/receive FIFO buffers or allocating transmit buffers to transmit queues, set the control register (RSCAN0TMCp) of the corresponding transmit buffer to 00_H. The status register (RSCAN0TMSTSp) of the corresponding transmit buffer should not be used. Flags in other status registers (registers RSCAN0TMTRSTS0 to RSCAN0TMTRSTS2, RSCAN0TMTARSTS0 to RSCAN0TMTARSTS2, RSCAN0TMTCSTS0 to RSCAN0TMTCSTS2, and RSCAN0TMTASTS0 to RSCAN0TMTASTS2), which correspond to transmit buffers linked to transmit/receive FIFO buffers or allocated to transmit queues remain unchanged. Set the enable bit in the corresponding interrupt enable register (registers RSCAN0TMIEC0 to RSCAN0TMIEC2) to 0 (transmit buffer interrupt is disabled).
- Transmit buffers that are linked to transmit/receive FIFO buffers must not bet allocated to transmit queues.
- Only a single transmit/receive FIFO buffer can be linked to a transmit buffer. Do not link two or more transmit/receive FIFO buffers to transmit buffers of the same number.
- When the CANm bit time clock is selected as a timestamp counter clock source, the timestamp counter stops when the corresponding channel has transitioned to channel reset mode or channel halt mode.
- In case of an attempt to store a new received message when the receive FIFO buffer and the transmit/receive FIFO buffer are full, the new message is discarded. If you wish to store a new transmit message in the transmit/receive FIFO buffer or the transmit queue, check that the transmit/receive FIFO buffer or the transmit queue is not full.
- The values of unused receive buffers (RSCAN0RMIDq, RSCAN0RMPTRq, RSCAN0RMDF0q, and RSCAN0RMDF1q registers), receive FIFO buffer access registers (RSCAN0RFIDx, RSCAN0RFPTRx, RSCAN0RFDF0x, and RSCAN0RFDF1x registers), and transmit/receive FIFO buffer access registers (RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers) are undefined when the RS-CAN module transitions to global operation mode or global test mode after exiting from global reset mode.

Section 21 FlexRay (FLXA)

This section contains a generic description of the FlexRay (FLXA).

The first part of this section describes RH850/F1M specific properties, such as the number of units, register base addresses, etc.

The remainder of this section describes the FLXA functions and registers.

CAUTION

Please read “global RAM” described in this section as “retention RAM”.

NOTE

The POC described in this section is an abbreviated form of Protocol Operation Control and differs from the *POC (power-on-clear)* described in *Section 10*.

21.1 Features of RH850/F1M FLXA

21.1.1 Number of Units and Channels

This microcontroller has the following number of units of the FLXA.

Table 21.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	1		
Name	FLXAn (n = 0)		

Table 21.2 FLXA Unit Configurations and Channels

Unit Name	No. of Channels	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
FLXA0	2 (A ch, B ch)	√	√	√

Table 21.3 Index

Index	Description
n	Throughout this section, the individual FLXA units are identified by the index “n”; for example, FLXAnFROC is the FlexRay control register.

21.1.2 Register Base Address

FLXAn base address is listed in the following table.

FLXAn register addresses are given as offsets from the base address in general.

Table 21.4 Register Base Address

Base Address Name	Base Address
<FLXA0_base>	1002 0000$_H$

21.1.3 Clock Supply

The FLXAn clock supply is shown in the following table.

Table 21.5 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
FLXA0	hclk	CPUCLK2 *1
	clkc	PPLLCLK *1
	Register access clock	CPUCLK2

Note 1. The frequency should be within the following range.
hclk = 45 - 80 MHz
clkc = 80 MHz

21.1.4 Interrupt Requests

The FLXAn interrupt requests are listed in the following table.

Table 21.6 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
FLXA0			
INTFLXA0LINE0	FlexRay0 interrupt	179	—
INTFLXA0LINE1	FlexRay1 interrupt	180	—
INTFLXA0TIM0	Timer 0 interrupt	181	—
INTFLXA0TIM1	Timer 1 interrupt	182	—
INTFLXA0TIM2	Timer 2 interrupt	183	—
INTFLXA0FDA	FIFO transfer interrupt	173	—
INTFLXA0FW	FIFO transfer warning interrupt	174	—
INTFLXA0OW	Output transfer warning interrupt	178	—
INTFLXA0OT	Output transfer end interrupt	177	—
INTFLXA0IQF	Input queue full interrupt	176	—
INTFLXA0IQE	Input queue empty interrupt	175	—

21.1.5 Reset Sources

FLXAn reset sources are listed in the following table. FLXAn is initialized by the following reset signal:

Table 21.7 Reset Sources

Unit Name	Reset Source
FLXA0	All reset sources (ISORES)

21.1.6 External Input/Output Signals

External input/output signals of FLXAn are listed below.

Table 21.8 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal Name
FLXA0		
rxda_extfxr	Channel A receive data input	FLXA0RXDA
fxr_txda	Channel A transmit data output	FLXA0TXDA
fxr_txena_n	Channel A transmit data enable	FLXA0TXENA
rxdb_extfxr	Channel B receive data input	FLXA0RXDB
fxr_txdb	Channel B transmit data output	FLXA0TXDB
fxr_txenb_n	Channel B transmit data enable	FLXA0TXENB
stpwt_extfxr	Stop watch trigger input	FLXA0STPWT

21.1.7 Functions

For communication on a FlexRay network, individual message buffers that can hold up to 254 bytes of data can be configured. Up to 128 message buffers can be configured in the message RAM. All functions concerning the handling of messages are implemented in the Message Handler. These functions include acceptance filtering, the transfer of messages between the two FlexRay Channel Protocol Controllers and the Message RAM, maintaining the transmission schedule, and providing message status information.

The register set of the FlexRay IP-module can be accessed directly by an external Host via the module's Host interface. These registers are used to control/configure/monitor the FlexRay Channel Protocol Controllers, Message Handler, Global Time Unit, System Universal Control, Frame and Symbol Processing, Network Management, Interrupt Control, to access the Message RAM via Input / Output Buffer; and to control the data transfer between the Message RAM and the Local RAM/Global RAM.

The FlexRay IP-module supports the following features:

Item	Specification
Communication	Conformance with FlexRay protocol specification v2.1 revision A
Data transfer rate	Up to 10 Mbit/s on each channel
Input/Output pins per channel	TxD, RxD, TxEN
FlexRay channels	2 (channels A and B)
Message buffers	Up to 128 message buffers configurable Configuration of message buffers with different payload lengths possible Each message buffer can be configured as a receive buffer, transmit buffer, part of a receive FIFO buffer. Filtering for slot counter, cycle counter, and channel
Message RAM	8 Kbyte of Message RAM for storage can be configured 128 message buffers with a data section of up to 48 bytes or up to 30 message buffers with a data section of 254 bytes
FIFO	One configurable receive FIFO
Message buffer access	By host CPU via input and output buffers Input Buffer: Holds message to be transferred to the Message RAM Output Buffer: Holds message read from the Message RAM
	By data transfer function Input transfer: Message buffer content is transferred from Local RAM/Global RAM to Message RAM on CPU request Output transfer: Message buffer content is transferred from Message RAM to Local RAM/Global RAM automatically
Network management	Supported
Interrupts	Maskable module interrupts
Timer	Two absolute timers One relative timer One stop watch timer

21.1.8 Block Diagram

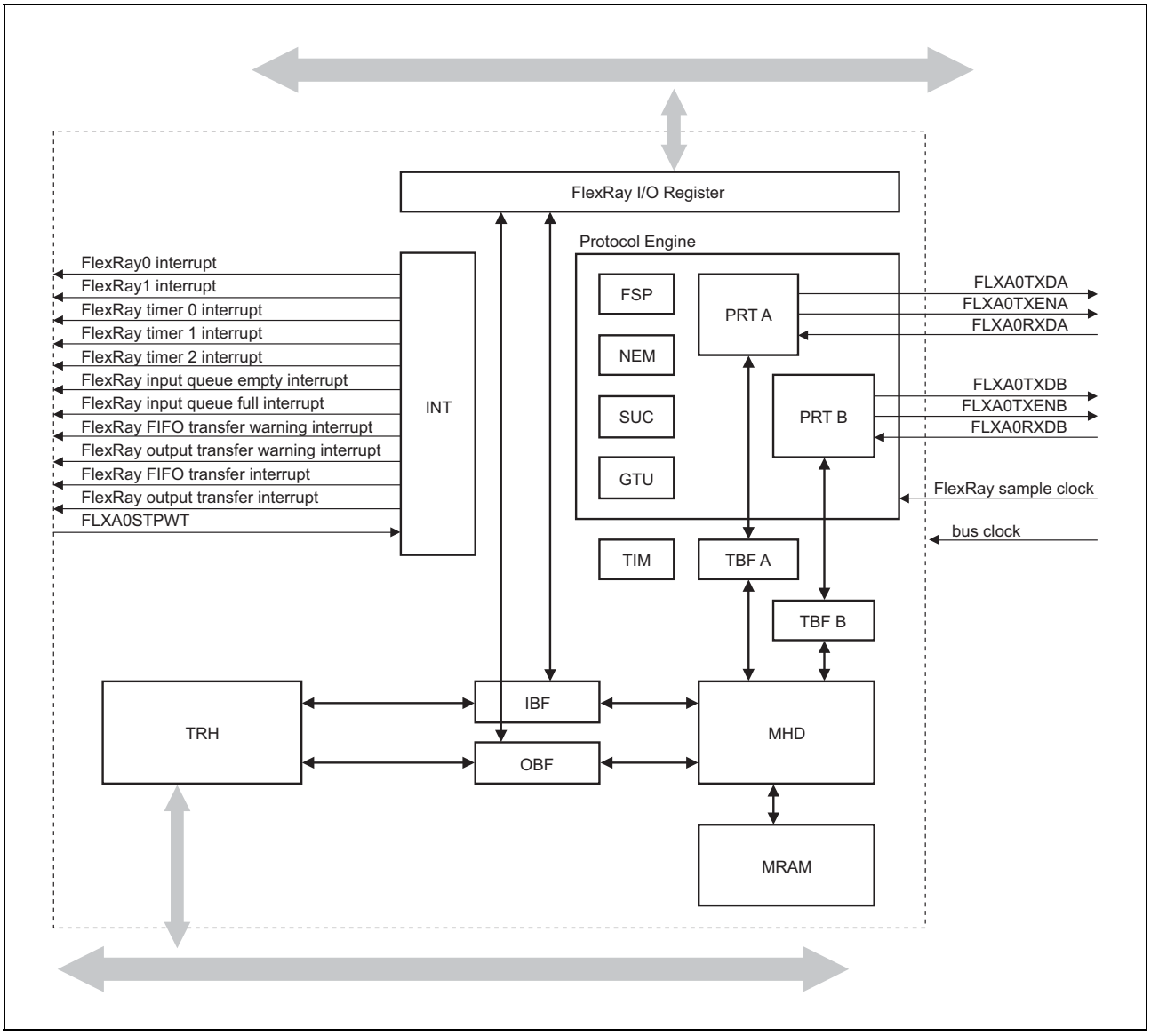


Figure 21.1 FlexRay IP Block Diagram

Input Buffer (IBF)

For write access to the message buffers configured in the Message RAM, the Host can write the header and data section for a specific message buffer to the Input Buffer. The Message Handler then transfers the data from the Input Buffer to the selected message buffer in the Message RAM.

Output Buffer (OBF)

For read access to a message buffer configured in the Message RAM the Message Handler transfers the selected message buffer to the Output Buffer. After the transfer has completed, the Host can read the header and data section of the transferred message buffer from the Output Buffer.

Message Handler (MHD)

The FlexRay Message Handler controls data transfers between the following components:

- Input / Output Buffer and Message RAM
- Temporary buffer RAMs of the two FlexRay Protocol Controllers and Message RAM

Message RAM (MRAM)

The Message RAM consists of a single-ported RAM that stores up to 128 FlexRay message buffers together with the related configuration data (header and data partition).

Temporary buffer RAM (TBF A/B)

Stores the data section of two complete messages.

FlexRay Channel Protocol Controller (PRT A/B)

The FlexRay Channel Protocol Controllers consist of shift register and FlexRay protocol FSM. They are connected to the Temporary buffer RAMs for intermediate message storage and to the physical layer via bus driver (BD).

This controller has the following functionality:

- Control and check of bit timing
- Reception / transmission of FlexRay frames and symbols
- Check of header CRC
- Generation / check of frame CRC
- Interfacing with bus driver

Global Time Unit (GTU)

The Global Time Unit performs the following functions:

- Generation of microtick (µT)
- Generation of macrotick (MT)
- Fault tolerant clock synchronization, by FTM algorithm, for:
  - rate correction
  - offset correction
- Cycle counter
- Timing control of static segment
- Timing control of dynamic segment (minislotting)
- Support of external clock correction

System Universal Control (SUC)

The System Universal Controller has the following functions:

- Configuration
- Wakeup
- Startup
- Normal Operation
- Passive Operation

Frame and Symbol Processing (FSP)

The Frame and Symbol Processing has the following functions:

- Checks whether the timing of frames and symbols is correct or not
- Checks whether the syntax and semantics of the received frame are correct or not
- Sets the slot status flags

Network Management (NEM)

Handles the network management vector.

Interrupt Control (INT)

The Interrupt Controller has the following functions:

- Provides error and status interrupt flags
- Enables and disables interrupt sources
- Assignment of interrupt sources to one of the two general module interrupt lines
- Enables and disables module interrupt lines

Timer (TIM)

The Timer module includes the following macrotick timer:

- one absolute timer
- one relative timer
- one stop watch timer

Transfer Handler (TRH)

Handles the data transfer between Local RAM/Global RAM and FlexRay module.

The Transfer Handler supports the following transfer types:

- Transfer of buffer configuration data from the Local RAM/Global RAM to the Message RAM
- Transfer of payload data for temporary buffers from the Local RAM/Global RAM to the Message RAM
- Transfer of buffer configuration data and payload data for temporary buffer from the Local RAM/Global RAM to the Message RAM
- Automatic transfer of payload data from receive buffer to the Local RAM/Global RAM upon frame reception
- Automatic transfer of payload data, buffer configuration data and message buffer status data from receive buffer to the Local RAM/Global RAM upon frame reception
- Automatic transfer of buffer configuration data and message buffer status data from the dedicated transmit/receive buffer to the Local RAM/Global RAM in response to slot status update
- Manual transfer of payload data, buffer configuration data and message buffer status data from the dedicated transmit/receive buffer to the Local RAM/Global RAM

21.2 Register

21.2.1 List of Registers

The FLXAn registers are listed in the following table.

For details on <FLXAn_base>, see **Section 21.1.2, Register Base Address**.

Table 21.9 List of Registers (1/3)

Module Name	Register Name	Symbol	Address
FLXAn	FlexRay Operation Control Register	FLXAnFROC	<FLXAn_base> + 0004_H
FLXAn	FlexRay Operation Status Register	FLXAnFROS	<FLXAn_base> + $000C_H$
FLXAn	FlexRay Test Register 1	FLXAnFRTEST1	<FLXAn_base> + 0010_H
FLXAn	FlexRay Test Register 2	FLXAnFRTEST2	<FLXAn_base> + 0014_H
FLXAn	FlexRay Lock Register	FLXAnFRLCK	<FLXAn_base> + $001C_H$
FLXAn	FlexRay Error Interrupt Register	FLXAnFREIR	<FLXAn_base> + 0020_H
FLXAn	FlexRay Status Interrupt Register	FLXAnFRSIR	<FLXAn_base> + 0024_H
FLXAn	FlexRay Error Interrupt Line Select	FLXAnFREILS	<FLXAn_base> + 0028_H
FLXAn	FlexRay Status Interrupt Line Select	FLXAnFRSILS	<FLXAn_base> + $002C_H$
FLXAn	FlexRay Error Interrupt Enable Set Register	FLXAnFREIES	<FLXAn_base> + 0030_H
FLXAn	FlexRay Error Interrupt Enable Reset Register	FLXAnFREIER	<FLXAn_base> + 0034_H
FLXAn	FlexRay Status Interrupt Enable Set Register	FLXAnFRSIES	<FLXAn_base> + 0038_H
FLXAn	FlexRay Status Interrupt Disable Register	FLXAnFRSIER	<FLXAn_base> + $003C_H$
FLXAn	FlexRay Interrupt Line Enable Register	FLXAnFRILE	<FLXAn_base> + 0040_H
FLXAn	FlexRay Timer 0 Configuration Register	FLXAnFRT0C	<FLXAn_base> + 0044_H
FLXAn	FlexRay Timer 1 Configuration Register	FLXAnFRT1C	<FLXAn_base> + 0048_H
FLXAn	FlexRay Stop Watch Register 1	FLXAnFRSTPW1	<FLXAn_base> + $004C_H$
FLXAn	FlexRay Stop Watch Register 2	FLXAnFRSTPW2	<FLXAn_base> + 0050_H
FLXAn	FlexRay SUC Configuration Register 1	FLXAnFRSUCC1	<FLXAn_base> + 0080_H
FLXAn	FlexRay SUC Configuration Register 2	FLXAnFRSUCC2	<FLXAn_base> + 0084_H
FLXAn	FlexRay SUC Configuration Register 3	FLXAnFRSUCC3	<FLXAn_base> + 0088_H
FLXAn	FlexRay NEM Configuration Register	FLXAnFRNEMC	<FLXAn_base> + $008C_H$
FLXAn	FlexRay PRT Configuration Register 1	FLXAnFRPRTC1	<FLXAn_base> + 0090_H
FLXAn	FlexRay PRT Configuration Register 2	FLXAnFRPRTC2	<FLXAn_base> + 0094_H
FLXAn	FlexRay MHD Configuration Register	FLXAnFRMHDC	<FLXAn_base> + 0098_H
FLXAn	FlexRay GTU Configuration Register 1	FLXAnFRGTUC1	<FLXAn_base> + $00A0_H$
FLXAn	FlexRay GTU Configuration Register 2	FLXAnFRGTUC2	<FLXAn_base> + $00A4_H$
FLXAn	FlexRay GTU Configuration Register 3	FLXAnFRGTUC3	<FLXAn_base> + $00A8_H$
FLXAn	FlexRay GTU Configuration Register 4	FLXAnFRGTUC4	<FLXAn_base> + $00AC_H$
FLXAn	FlexRay GTU Configuration Register 5	FLXAnFRGTUC5	<FLXAn_base> + $00B0_H$
FLXAn	FlexRay GTU Configuration Register 6	FLXAnFRGTUC6	<FLXAn_base> + $00B4_H$
FLXAn	FlexRay GTU Configuration Register 7	FLXAnFRGTUC7	<FLXAn_base> + $00B8_H$
FLXAn	FlexRay GTU Configuration Register 8	FLXAnFRGTUC8	<FLXAn_base> + $00BC_H$
FLXAn	FlexRay GTU Configuration Register 9	FLXAnFRGTUC9	<FLXAn_base> + $00C0_H$
FLXAn	FlexRay GTU Configuration Register 10	FLXAnFRGTUC10	<FLXAn_base> + $00C4_H$
FLXAn	FlexRay GTU Configuration Register 11	FLXAnFRGTUC11	<FLXAn_base> + $00C8_H$
FLXAn	FlexRay CC Status Vector Register	FLXAnFRCCSV	<FLXAn_base> + 0100_H
FLXAn	FlexRay CC Error Vector Register	FLXAnFRCCEV	<FLXAn_base> + 0104_H

Table 21.9 List of Registers (2/3)

Module Name	Register Name	Symbol	Address
FLXAn	FlexRay Slot Counter Value Register	FLXAnFRSCV	<FLXAn_base> + 0110_H
FLXAn	FlexRay MT Value/Cycle Counter Value Register	FLXAnFRMTCCV	<FLXAn_base> + 0114_H
FLXAn	FlexRay Rate Correction Value Register	FLXAnFRRCV	<FLXAn_base> + 0118_H
FLXAn	FlexRay Offset Correction Value Register	FLXAnFROCV	<FLXAn_base> + $011C_H$
FLXAn	FlexRay Sync Frame Status Register	FLXAnFRSFS	<FLXAn_base> + 0120_H
FLXAn	FlexRay Symbol Window and NIT Status Register	FLXAnFRSWNIT	<FLXAn_base> + 0124_H
FLXAn	FlexRay Aggregated Channel Status Register	FLXAnFRACS	<FLXAn_base> + 0128_H
FLXAn	FlexRay Even Sync ID Register m (m = 1 to 15)	FLXAnFRESIDm (m = 1 to 15)	<FLXAn_base> + 0130_H to <FLXAn_base> + 0168_H (<FLXAn_base> + 0130_H + (n-1) × 4_H)
FLXAn	FlexRay Odd Sync ID Register m (m = 1 to 15)	FLXAnFROSIDm (m = 1 to 15)	<FLXAn_base> + 0170_H to <FLXAn_base> + $01A8_H$ (<FLXAn_base> + 0170_H + (m-1) × 4_H)
FLXAn	FlexRay Network Management Vector Register m (m = 1 to 3)	FLXAnFRNMVm (m = 1 to 3)	<FLXAn_base> + $01B0_H$ to <FLXAn_base> + $01B8_H$ (<FLXAn_base> + $01B0_H$ + (m-1) × 4_H)
FLXAn	FlexRay Message RAM Configuration Register	FLXAnFRMRC	<FLXAn_base> + 0300_H
FLXAn	FlexRay FIFO Rejection Filter Register	FLXAnFRFRF	<FLXAn_base> + 0304_H
FLXAn	FlexRay FIFO Rejection Filter Mask Register	FLXAnFRFRFM	<FLXAn_base> + 0308_H
FLXAn	FlexRay FIFO Critical Level Register	FLXAnFRFCL	<FLXAn_base> + $030C_H$
FLXAn	FlexRay Message Handler Status Register	FLXAnFRMHDS	<FLXAn_base> + 0310_H
FLXAn	FlexRay Last Dynamic Transmit Slot Register	FLXAnFRLDTS	<FLXAn_base> + 0314_H
FLXAn	FlexRay FIFO Status Register	FLXAnFRFSR	<FLXAn_base> + 0318_H
FLXAn	FlexRay Message Handler Constraints Flags Register	FLXAnFRMHDF	<FLXAn_base> + $031C_H$
FLXAn	FlexRay Transmission Request Register i (i = 1 to 4)	FLXAnFRTXRQi (i = 1 to 4)	<FLXAn_base> + 0320_H to <FLXAn_base> + $032C_H$ (<FLXAn_base> + 0320_H + (i-1) × 4_H)
FLXAn	FlexRay New Data Register i (i = 1 to 4)	FLXAnFRNDATi (i = 1 to 4)	<FLXAn_base> + 0330_H to <FLXAn_base> + $033C_H$ (<FLXAn_base> + 0330_H + (i-1) × 4_H)
FLXAn	FlexRay Message Buffer Status Changed Register i (i = 1 to 4)	FLXAnFRMBSCi (i = 1 to 4)	<FLXAn_base> + 0340_H to <FLXAn_base> + $034C_H$ (<FLXAn_base> + 0340_H + (i-1) × 4_H)
FLXAn	FlexRay Write Data Section Register x (x = 1 to 64)	FLXAnFRWRDSx (x = 1 to 64)	<FLXAn_base> + 0400_H to <FLXAn_base> + $04FC_H$ (<FLXAn_base> + 0400_H + (x-1) × 4_H)
FLXAn	FlexRay Write Header Section Register 1	FLXAnFRWRHS1	<FLXAn_base> + 0500_H
FLXAn	FlexRay Write Header Section Register 2	FLXAnFRWRHS2	<FLXAn_base> + 0504_H
FLXAn	FlexRay Write Header Section Register 3	FLXAnFRWRHS3	<FLXAn_base> + 0508_H
FLXAn	FlexRay Input Buffer Command Mask Register	FLXAnFRIBCM	<FLXAn_base> + 0510_H
FLXAn	FlexRay Input Buffer Command Request Register	FLXAnFRIBCR	<FLXAn_base> + 0514_H

Table 21.9 List of Registers (3/3)

Module Name	Register Name	Symbol	Address
FLXAn	FlexRay Read Data Section Register x (x = 1 to 64)	FLXAnFRRDDSx (x = 1 to 64)	<FLXAn_base> + 0600_H to <FLXAn_base> + $06FC_H$ (<FLXAn_base> + 0600_H + (x-1) × 4_H)
FLXAn	FlexRay Read Header Section Register 1	FLXAnFRRDHS1	<FLXAn_base> + 0700_H
FLXAn	FlexRay Read Header Section Register 2	FLXAnFRRDHS2	<FLXAn_base> + 0704_H
FLXAn	FlexRay Read Header Section Register 3	FLXAnFRRDHS3	<FLXAn_base> + 0708_H
FLXAn	FlexRay Message Buffer Status Register	FLXAnFRMBS	<FLXAn_base> + $070C_H$
FLXAn	FlexRay Output Buffer Command Mask Register	FLXAnFROBCM	<FLXAn_base> + 0710_H
FLXAn	FlexRay Output Buffer Command Request Register	FLXAnFROBCR	<FLXAn_base> + 0714_H
FLXAn	FlexRay Input Transfer Configuration Register	FLXAnFRITC	<FLXAn_base> + 0800_H
FLXAn	FlexRay Output Transfer Configuration Register	FLXAnFROTC	<FLXAn_base> + 0804_H
FLXAn	FlexRay Input pointer table Base Address Register	FLXAnFRIBA	<FLXAn_base> + 0808_H
FLXAn	FlexRay FIFO pointer table Base Address Register	FLXAnFRFBA	<FLXAn_base> + $080C_H$
FLXAn	FlexRay Output pointer table Base Address Register	FLXAnFROBA	<FLXAn_base> + 0810_H
FLXAn	FlexRay Input Queue Control Register	FLXAnFRIQC	<FLXAn_base> + 0814_H
FLXAn	FlexRay User Input transfer Request Register	FLXAnFRUIR	<FLXAn_base> + 0818_H
FLXAn	FlexRay User Output transfer Request Register	FLXAnFRUOR	<FLXAn_base> + $081C_H$
FLXAn	FlexRay Input Transfer Status Register	FLXAnFRITS	<FLXAn_base> + 0820_H
FLXAn	FlexRay Output Transfer Status Register	FLXAnFROTS	<FLXAn_base> + 0824_H
FLXAn	FlexRay Access Error Status Register	FLXAnFRAES	<FLXAn_base> + 0828_H
FLXAn	FlexRay Access Error Address Register	FLXAnFRAEA	<FLXAn_base> + $082C_H$
FLXAn	FlexRay message Data Available Register i (i = 0 to 3)	FLXAnFRDAi (i = 0 to 3)	<FLXAn_base> + 0830_H to <FLXAn_base> + $083C_H$ (<FLXAn_base> + 0830_H + i × 4_H)
FLXAn	FlexRay H-Bus Configuration Register	FLXAnFRAHBC	<FLXAn_base> + 0840_H
FLXAn	FlexRay Timer 2 Configuration Register	FLXAnFRT2C	<FLXAn_base> + 0844_H

21.2.2 FlexRay Operation register

21.2.2.1 FLXAnFROC — FlexRay Operation Control Register

Access: FLXAnFROC can be read or written in 32-bit units.
FLXAnFROCL and FLXAnFROCH can be read or written in 16-bit units.
FLXAnFROCLL and FLXAnFROCHL can be read or written in 8-bit units.

Address: FLXAnFROC: <FLXAn_base> + 0004_H
FLXAnFROCL: <FLXAn_base> + 0004_H, FLXAnFROCH: <FLXAn_base> + 0006_H
FLXAnFROCLL: <FLXAn_base> + 0004_H, FLXAnFROCHL: <FLXAn_base> + 0006_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	T2IE	T1IE	T0IE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OEP	—	—	—	—	—	BEC	OE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W

Table 21.10 FLXAnFROC Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	T2IE	Timer 2 interrupt enable Bit 0: Disabled 1: Enabled
17	T1IE	Timer 1 interrupt enable Bit 0: Disabled 1: Enabled
16	T0IE	Timer 0 interrupt enable Bit 0: Disabled 1: Enabled
15 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	OEP	Operation Enable bit Protection Bit 0: OE is unprotected 1: OE is protected
6 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	BEC	Byte Endian Control Bit 0: Little endian 1: Big endian
0	OE	Operation Enable Bit 0: Operation disabled, SW reset 1: Operation Enabled

(1) FLXAnFROC.T2IE

Timer 2 interrupt enable bit

This bit controls the timer 2 interrupt.

0: Disabled

No interrupt will be requested and the timer 2 interrupt line will be released if pending.

1: Enabled

Timer 2 interrupt will be asserted when FLXAnFROTS.T2IS is 1.

(2) FLXAnFROC.T1IE

Timer 1 interrupt enable bit

The user can only set this bit to 1 when timer 1 interrupt is not enabled in the FlexRay Status interrupt enable register (FLXAnFRSIES.TI1E is 0).

This bit controls the timer 1 interrupt.

0: Disabled

No interrupt will be requested and the timer 1 interrupt line will be released if pending.

1: Enabled

Timer 1 interrupt will be asserted when FLXAnFROTS.T1IS is 1.

(3) FLXAnFROC.T0IE

Timer 0 interrupt enable bit

The user can only set this bit to 1 when timer 0 interrupt is not enabled in the FlexRay Status interrupt enable register (FLXAnFRSIES.TI0E is 0).

This bit controls the timer 0 interrupt.

0: Disabled

No interrupt will be requested and the timer 0 interrupt line will be released if pending.

1: Enabled

Timer 0 interrupt will be asserted when FLXAnFROTS.T0IS is 1.

(4) FLXAnFROC.OEP

Operation Enable bit Protection bit

This bit protects against unintended write access to the OE bit.

0: OE is unprotected

Write access to the OE bit is enabled

1: OE is protected

Write access to the OE bit is disabled

(5) FLXAnFROC.BEC

Byte Endian Control bit

The user can only change this bit when FLXAnFROS.OS is 1.

This bit controls the byte order on reading and writing the FlexRay Network Management Vector register (FLXAnFRNMVm), FlexRay Write Data Section (FLXAnFRWRDSx) and FlexRay Read Data Section (FLXAnFRRDDSx). This bit also controls the byte order when reading or writing FlexRay payload data using the data transfer function.

For details about the byte alignment please refer to **Section 21.3.17, Byte Alignment**.

0: Little endian

Byte alignment in FLXAnFRNMVm, FLXAnFRWRDSx and FLXAnFRRDDSx is in little endian style.

1: Big endian

Byte alignment in FLXAnFRNMVm, FLXAnFRWRDSx and FLXAnFRRDDSx is in big endian style.

(6) FLXAnFROC.OE

Operation Enable bit

The user can only write to this bit when FLXAnFROC.OEP is 0.

The user should only write this bit with 0 when FLXAnFROS.OS is 1.

The user should only write this bit with 1 when FLXAnFROS.OS is 0 and the FlexRay sample clock is enabled.

This bit controls the operation state and executes the software reset of the FlexRay module. The operation status bit (FLXAnFROS.OS) indicates whether the FlexRay module is in reset state or not.

0: Operation disabled, SW reset

Forcibly moves the FlexRay module to its reset state, whatever the state of the FlexRay module is.

1: Operation Enabled

Reset state of the FlexRay module is released.

21.2.2.2 FLXAnFROS — FlexRay Operation Status Register

Access: FLXAnFROS can be read or written in 32-bit units.
FLXAnFROSL is a read-only register that can be read in 16-bit units.
FLXAnFROSH can be read or written in 16-bit units.
FLXAnFROSLL is a read-only register that can be read in 8-bit units.
FLXAnFROSHL can be read or written in 8-bit units.

Address: FLXAnFROS: <FLXAn_base> + $000C_H$
FLXAnFROSL: <FLXAn_base> + $000C_H$, FLXAnFROSH: <FLXAn_base> + $000E_H$,
FLXAnFROSLL: <FLXAn_base> + $000C_H$, FLXAnFROSHL: <FLXAn_base> + $000E_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	T2IS	T1IS	T0IS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.11 FLXAnFROS Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	T2IS	Timer 2 Interrupt Status Bit 0: Timer 2 has not matched the conditions configured in the FLXAnFRT2C register 1: Timer 2 matched the conditions configured in the FLXAnFRT2C register
17	T1IS	Timer 1 Interrupt Status Bit 0: Timer 1 has not matched the conditions configured in the FLXAnFRT1C register 1: Timer 1 matched the conditions configured in the FLXAnFRT1C register
16	T0IS	Timer 0 Interrupt Status Bit 0: Timer 0 has not matched the conditions configured in the FLXAnFRT0C register 1: Timer 0 matched the conditions configured in the FLXAnFRT0C register
15 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OS	Operation Status Bit 0: Operation disabled, reset state 1: Operation enabled

(1) FLXAnFROS.T2IS

Timer 2 Interrupt Status Bit

Writing 0 has no effect on the bit value.

This bit indicates that the expiration criteria configured in FLXAnFRT2C register has matched the FlexRay local time.

If enabled in FLXAnFROC.T2IE the timer 2 interrupt is generated when FLXAnFROS.T2IS is 1.

[Clearing condition]

This bit is cleared by writing 1 to FLXAnFROS.T2IS.
This bit is cleared when FLXAnFROS.OS changes from 1 to 0.

[Setting condition]

This bit is set when the expiration criteria configured in FLXAnFRT2C matches the FlexRay local time.

(2) FLXAnFROS.T1IS

Timer 1 Interrupt Status Bit

Writing 0 has no effect on the bit value.

This bit indicates that the expiration criteria configured in FLXAnFRT1C register has matched the FlexRay local time.

If enabled in FLXAnFROC.T1IE the timer 1 interrupt is generated when FLXAnFROS.T1IS is 1.

[Clearing condition]

This bit is cleared by writing 1 to FLXAnFROS.T1IS.
This bit is cleared when FLXAnFROS.OS changes from 1 to 0.

[Setting condition]

This bit is set when the expiration criteria configured in FLXAnFRT1C matches the FlexRay local time.

(3) FLXAnFROS.T0IS

Timer 0 Interrupt Status Bit

Writing 0 has no effect on the bit value.

This bit indicates that the expiration criteria configured in FLXAnFRT0C register has matched the FlexRay local time.

If enabled in FLXAnFROC.T0IE the timer 0 interrupt is generated when FLXAnFROS.T0IS is 1.

[Clearing condition]

This bit is cleared by writing 1 to FLXAnFROS.T0IS.
This bit is cleared when FLXAnFROS.OS changes from 1 to 0.

[Setting condition]

This bit is set when the expiration criteria configured in FLXAnFRT0C matches the FlexRay local time.

(4) FLXAnFROS.OS

Operation Status Bit

This bit indicates if the FlexRay module is in the reset or the operation state.

When FLXAnFROS.OS is 0 the FlexRay module gets initialized and registers mapped to the address area <FLXAn_base>+ 0010_H to <FLXAn_base> + $0FFF_H$ cannot be accessed; read access from these registers will return undefined data.

When FLXAnFROS.OS is 1 it is possible to access to the address area <FLXAn_base> + 0010_H to <FLXAn_base> + $0FFF_H$ and to perform FlexRay communication.

When FLXAnFROS.OS changes from 0 to 1 all registers in the address range <FLXAn_base> + 0010_H to <FLXAn_base> + $0FFF_H$ are set to the “Values after reset”.

[Clearing condition]

When FLXAnFROC.OE is set to 0. It takes up to two peripheral bus clock cycles until FLXAnFROS.OS is set to 0.

[Setting condition]

When FLXAnFROC.OE is set to 1 it takes up to four peripheral clock cycles of the clock with the lower frequency out of the FlexRay sample clock and peripheral bus clock until FLXAnFROS.OS is set to 1.

21.2.3 Special Registers

21.2.3.1 FLXAnFRTEST1 — FlexRay Test Register 1

The FlexRay Test Register 1 holds the control bits to configure the test modes of the FlexRay module.

When the FlexRay is operated in one of its test modes that requires FLXAnFRTEST1.WRTEN to be set (RAM Test Mode, I/O Test Mode, Asynchronous Transmit Mode, and Loop Back Mode) only the selected test mode functionality is available.

To return from test mode operation to regular FlexRay operation we strongly recommend to apply a reset using FLXAnFROC.OE to reset all FlexRay internal state machines to their initial state.

The test functions are not available in addition to the normal operational mode functions, they change the functions of parts of the FlexRay module. Therefore normal operation as specified outside this chapter and as required by the FlexRay protocol specification and the FlexRay conformance test is not possible. Test mode functions may not be combined with each other or with FlexRay protocol functions.

The test mode features are intended for hardware testing or for FlexRay bus analyzer tools. They are not intended to be used in FlexRay applications.

Access: FLXAnFRTEST1 can be read or written in 32-bit units.

Address: FLXAnFRTEST1: <FLXAn_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CERB[3:0]				CERA[3:0]				—	—	TXENB	TXENA	TXB	TXA	RXB	RXA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	AOB	AOA	—	—	TMC[1:0]		—	—	ELBE	WRTEN
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 21.12 FLXAnFRTEST1 Register Contents (1/2)

Bit	Bit Name	Function
31 to 28	CERB[3:0]	Coding Error Report Channel B 0000_B: No coding error detected 0001_B: Header CRC error detected 0010_B: Frame CRC error detected 0011_B: Frame Start Sequence FSS too long 0100_B: First bit of Byte Start Sequence BSS seen low-level 0101_B: Second bit of Byte Start Sequence BSS seen high-level 0110_B: First bit of Frame End Sequence FES seen high-level 0111_B: Second bit of Frame End Sequence FES seen low-level 1000_B: CAS / MTS symbol seen too short 1001_B: CAS / MTS symbol seen too long 1010_B to 1111_B: reserved

Table 21.12 FLXAnFRTEST1 Register Contents (2/2)

Bit	Bit Name	Function
27 to 24	CERA[3:0]	Coding Error Report Channel A 0000_B: No coding error detected 0001_B: Header CRC error detected 0010_B: Frame CRC error detected 0011_B: Frame Start Sequence FSS too long 0100_B: First bit of Byte Start Sequence BSS seen low-level 0101_B: Second bit of Byte Start Sequence BSS seen high-level 0110_B: First bit of Frame End Sequence FES seen high-level 0111_B: Second bit of Frame End Sequence FES seen low-level 1000_B: CAS / MTS symbol seen too short 1001_B: CAS / MTS symbol seen too long 1010_B to 1111_B: reserved
23, 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	TXENB	Control of Channel B Transmit Enable Pin 0: FLXAnTENB pin drives a 0 1: FLXAnTENB pin drives a 1
20	TXENA	Control of Channel A Transmit Enable Pin 0: FLXAnTENA pin drives a 0 1: FLXAnTENA pin drives a 1
19	TXB	Control of Channel B Transmit Pin 0: FLXAnTXDB pin drives a 0 1: FLXAnTXDB pin drives a 1
18	TXA	Control of Channel A Transmit Pin 0: FLXAnTXDA pin drives a 0 1: FLXAnTXDA pin drives a 1
17	RXB	Monitor Channel B Receive Pin 0: FLXAnRXDB is 0 1: FLXAnRXDB is 1
16	RXA	Monitor Channel A Receive Pin 0: FLXAnRXDA is 0 1: FLXAnRXDA is 1
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	AOB	Activity on B 0: No activity detected, channel B idle 1: Activity detected, channel B not idle
8	AOA	Activity on A 0: No activity detected, channel A idle 1: Activity detected, channel A not idle
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5, 4	TMC[1:0]	Test Multiplexer Control 00_B: Normal signal path 01_B: RAM Test Mode 10_B: I/O Test Mode 11_B: Normal signal path
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ELBE	External Loop Back Enable 0: Internal Loop Back 1: External Loop Back
0	WRTEN	Write Test Register Enable 0: Write access to test registers disabled 1: Write access to test registers enabled

(1) FLXAnFRTEST1.CERB

Coding Error Report Channel B

Set when a coding error is detected on channel B.

Reset to zero when register FLXAnFRTEST1 is read or written. Once the FLXAnFRTEST1.CERB is set it will remain unchanged until FLXAnFRTEST1 register is accessed.

NOTES

1. Coding errors are signalled in all states where frame decoding is possible. FLXAnFRTEST1.CERB should be ignored in all other states.
2. The error codes regarding CAS / MTS symbols concern only the monitored bit pattern, irrelevant whether those bit patterns are seen in the symbol window or elsewhere.

(2) FLXAnFRTEST1.CERA

Coding Error Report Channel A

Set when a coding error is detected on channel A.

Reset to zero when register FLXAnFRTEST1 is read or written. Once the FLXAnFRTEST1.CERA is set it will remain unchanged until FLXAnFRTEST1 register is accessed.

NOTES

1. Coding errors are signalled in all states where frame decoding is possible. FLXAnFRTEST1.CERA should be ignored in all other states.
2. The error codes regarding CAS / MTS symbols concern only the monitored bit pattern, irrelevant whether those bit patterns are seen in the symbol window or elsewhere.

(3) FLXAnFRTEST1.TXENB

Control of Channel B Transmit Enable Pin

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.

This bit is used to test the interface to the physical layer (connectivity test) by driving the FLXAnTENB pin.

(4) FLXAnFRTEST1.TXENA

Control of Channel A Transmit Enable Pin

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.

This bit is used to test the interface to the physical layer (connectivity test) by driving the FLXAnTENA pin.

(5) FLXAnFRTEST1.TXB

Control of Channel B Transmit Pin

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.

This bit is used to test the interface to the physical layer (connectivity test) by driving the FLXAnTXDB pin.

(6) FLXAnFRTEST1.TXA

Control of Channel A Transmit Pin

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.

This bit is used to test the interface to the physical layer (connectivity test) by driving the FLXAnTXDA pin.

(7) FLXAnFRTEST1.RXB

Monitor Channel B Receive Pin

This bit is used to test the interface to the physical layer (connectivity test) by reading the FLXAnRXDB pin.

(8) FLXAnFRTEST1.RXA

Monitor Channel A Receive Pin

This bit is used to test the interface to the physical layer (connectivity test) by reading the FLXAnRXDA pin.

(9) FLXAnFRTEST1.AOB

Activity on B

FLXAnFRTEST1.AOB is set when there is activity on channel B or if the POC state is in DEFAULT_CONFIG or CONFIG state.

During STARTUP, NORMAL_ACTIVE or NORMAL_PASSIVE the function of FLXAnFRTEST1.AOB is inverse of zChannelIdle as specified in the FlexRay protocol specification V2.1, section 3, BITSTRB process. FLXAnFRTEST1.AOB should be ignored in all other POC states.

(10) FLXAnFRTEST1.AOA

Activity on A

FLXAnFRTEST1.AOA is set when there is activity on channel A or if the POC state is in DEFAULT_CONFIG or CONFIG state.

During STARTUP, NORMAL_ACTIVE or NORMAL_PASSIVE the function of FLXAnFRTEST1.AOA is inverse of zChannelIdle as specified in the FlexRay protocol specification V2.1, section 3, BITSTRB process. FLXAnFRTEST1.AOA should be ignored in all other POC states.

(11) FLXAnFRTEST1.TMC

Test Multiplexer Control

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.

00_B = Normal signal path

01_B = RAM Test Mode

Internal busses are multiplexed to make the Message RAM, Transient buffer RAM A and Transient buffer RAM B of the FlexRay module directly accessible by the Host. This mode is intended to enable testing of the FlexRay RAM during production testing.

10_B = I/O Test Mode

Output pins FLXAnTXDA, FLXAnTXDB, FLXAnTENA, FLXAnTENB, are driven to the values defined by bits FLXAnFRTEST1.TXA, FLXAnFRTEST1.TXB, FLXAnFRTEST1.TXENA,

FLXAnFRTEST1.TXENB. The values applied to the input pins FLXAnRXDA, FLXAnRXDB can be read from register bits FLXAnFRTEST1.RXA, FLXAnFRTEST1.RXB.

11_B = Normal signal path

(12) FLXAnFRTEST1.ELBE

External Loop Back Enable

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to 1.

There are two possibilities to perform a Loop Back test. External Loop Back via physical layer or internal Loop Back for in-system self-test.

In case of an internal Loop Back the FlexRay IP module pins FLXAnTENA, FLXAnTENB are in their inactive state, pins FLXAnTXDA, FLXAnTXDB are set to HIGH, pins FLXAnRXDA, FLXAnRXDB are not evaluated. Bit FLXAnFRTEST1.ELBE is evaluated only when POC is in Loop Back Mode and test multiplexer control is in non-multiplexing mode FLXAnFRTEST1.TMC = 00_B.

(13) FLXAnFRTEST1.WRTEN

Write Test Register Enable

To set the bit from 0 to 1 the test mode key has to be written as defined in **Section 21.2.3.3, FLXAnFRLCK — FlexRay Lock Register**. Enables write access to the test registers. The unlock sequence is not required when FLXAnFRTEST1.WRTEN is kept at 1 while other bits of the register are changed.

The bit can be reset to 0 at any time.

21.2.3.2 FLXAnFRTEST2 — FlexRay Test Register 2

The FlexRay Test Register 2 holds all bits required for the RAM test of the Message RAM, Transient buffer RAM A and Transient buffer RAM B of the FlexRay module.

Access: FLXAnFRTEST2 can be read or written in 32-bit units.
Address: FLXAnFRTEST2: <FLXAn_base> + 0014_H
Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SSEL[2:0]			—	RS[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 21.13 FLXAnFRTEST2 Register Contents

Bit	Bit Name	Function
31 to 7	Reserved	These bits are always read as 0. The write value should be always 0.
6 to 4	SSEL[2:0]	Segment Select 000_B: Access to RAM bytes 0000_H to $03FF_H$ enabled 001_B: Access to RAM bytes 0400_H to $07FF_H$ enabled 010_B: Access to RAM bytes 0800_H to $0BFF_H$ enabled 011_B: Access to RAM bytes $0C00_H$ to $0FFF_H$ enabled 100_B: Access to RAM bytes 1000_H to $13FF_H$ enabled 101_B: Access to RAM bytes 1400_H to $17FF_H$ enabled 110_B: Access to RAM bytes 1800_H to $1BFF_H$ enabled 111_B: Access to RAM bytes $1C00_H$ to $1FFF_H$ enabled
3	Reserved	This bit is always read as 0. The write value should be always 0.
2 to 0	RS[2:0]	RAM Select 000_B = Unused 001_B = Unused 010_B = Unused 011_B = Unused 100_B = Transient buffer RAM A (TBF1) 101_B = Transient buffer RAM B (TBF2) 110_B = Message RAM (MBF) 111_B = Unused

(1) FLXAnFRTEST2.SSEL

Segment Select

Write access to these bits is only possible if bit FLXAnFRTEST1.WRTEN is set to ’1’.

To enable access to the complete Message RAM (8192 byte addresses) the Message RAM is segmented into portions of 1024 bytes.

(2) FLXAnFRTEST2.RS

RAM Select

Write access to this register is only possible when FLXAnFRTEST1.WRTEN is set to ’1’.

In RAM Test mode the RAM blocks selected by FLXAnFRTEST2.RS are mapped to module address <FLXAn_base> + 0400_H to ERAY+$07FF_H$ (1024 byte addresses).

21.2.3.3 FLXAnFRLCK — FlexRay Lock Register

Access: FLXAnFRLCK can be read or written in 32-bit units.
FLXAnFRLCKL can be read or written in 16-bit units.
FLXAnFRLCKLL can be read or written in 8-bit units.

Address: FLXAnFRLCK: <FLXAn_base> + $001C_H$,
FLXAnFRLCKL: <FLXAn_base> + $001C_H$,
FLXAnFRLCKLL: <FLXAn_base> + $001C_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLK[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.14 FLXAnFRLCK Register Contents

Bit	Symbol	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	CLK[7:0]	Configuration Lock Key Bit

(1) FLXAnFRLCK.CLK

Configuration Lock Key Bit

The Lock Register is write-only. Reading the register will return $0000\ 0000_H$.

To leave CONFIG state by writing FLXAnFRSUCC1.CMD (command READY), the write operation has to be directly preceded by two write accesses to the Configuration Lock Key (unlock sequence). If the write sequence below is interrupted by other write accesses between the second write to the Configuration Lock Key and the write access to the FLXAnFRSUCC1 register, the CC remains in CONFIG state and the sequence has to be repeated.

First write: FLXAnFRLCK.CLK = "$1100\ 1110_B$" (CE_H)

Second write: FLXAnFRLCK.CLK = "$0011\ 0001_B$" (31_H)

Third write: FLXAnFRSUCC1.CMD

CAUTION

If the Host uses 8/16-bit accesses to write the listed bit fields, the user has to ensure that no dummy accesses to the remaining register bytes / words are inserted by the compiler.

21.2.4 Interrupt Registers

21.2.4.1 FLXAnFREIR — FlexRay Error Interrupt Register

The flags are set when the CC detects one of the listed error conditions. The flags remain set until cleared.

Access: FLXAnFREIR can be read or written in 32-bit units.
FLXAnFREIRL and FLXAnFREIRH can be read or written in 16-bit units.
FLXAnFREIRLL, FLXAnFREIRLH, FLXAnFREIRHL, and FLXAnFREIRHH can be read or written in 8-bit units.

Address: FLXAnFREIR: <FLXAn_base> + 0020_H,
FLXAnFREIRL: <FLXAn_base> + 0020_H, FLXAnFREIRH: <FLXAn_base> + 0022_H,
FLXAnFREIRLL: <FLXAn_base> + 0020_H, FLXAnFREIRLH: <FLXAn_base> + 0021_H,
FLXAnFREIRHL: <FLXAn_base> + 0022_H, FLXAnFREIRHH: <FLXAn_base> + 0023_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABB	LTVB	EDB	—	—	—	—	—	TABA	LTVA	EDA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHF	IOBA	IIBA	EFA	RFO	AERR	CCL	CCF	SFO	SFBM	CNA	PEMC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.15 FLXAnFREIR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABB	Transmission Across Boundary Channel B Flag 0: No transmission across slot boundary detected on channel B 1: Transmission across slot boundary detected on channel B
25	LTVB	Latest Transmit Violation Channel B Flag 0: No latest transmit violation detected on channel B 1: Latest transmit violation detected on channel B
24	EDB	Error Detected on Channel B Flag 0: No error detected on channel B 1: Error detected on channel B
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABA	Transmission Across Boundary Channel A Flag 0: No transmission across slot boundary detected on channel A 1: Transmission across slot boundary detected on channel A
17	LTVA	Latest Transmit Violation Channel A Flag 0: No latest transmit violation detected on channel A 1: Latest transmit violation detected on channel A
16	EDA	Error Detected on Channel A Flag 0: No error detected on channel A 1: Error detected on channel A
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 21.15 FLXAnFREIR Register Contents (2/2)

Bit Position	Bit Name	Function
11	MHF	Message Handler Constraints Flag 0: No message handler constraint violation detected 1: Message handler constraint violation detected
10	IOBA	Illegal Output buffer Access Flag 0: No illegal Host access to Output Buffer occurred 1: Illegal Host access to Output Buffer occurred
9	IIBA	Illegal Input Buffer Access Flag 0: No illegal Host access to Input Buffer occurred 1: Illegal Host access to Input Buffer occurred
8	EFA	Empty FIFO Access Flag 0: No access to empty FIFO occurred 1: Access to empty FIFO occurred
7	RFO	Receive FIFO Overrun Flag 0: No receive FIFO overrun detected 1: A receive FIFO overrun has been detected
6	AERR	Access error flag Flag 0: Access error is not detected. 1: Access error is detected.
5	CCL	CHI Command Locked Flag 0: CHI command not accepted 1: CHI command accepted
4	CCF	Clock Correction Failure Flag 0: No clock correction error 1: Clock correction failed
3	SFO	Sync Frame Overflow Flag 0: Number of received sync frames ≤ FLXAnFRGTUC2 1: More sync frames received than configured by FLXAnFRGTUC2
2	SFBM	Sync Frames Below Minimum Flag 0: Sync node: 1 or more sync frames received, non-sync node: 2 or more sync frames received 1: Less than the required minimum of sync frames received
1	CNA	Command Not Accepted Flag 0: CHI command accepted 1: CHI command not accepted
0	PEMC	POC Error Mode Changed Flag 0: Error mode has not changed 1: Error mode has changed

(1) FLXAnFREIR.TABB

Transmission Across Boundary Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

The flag indicates that a transmission across a slot boundary occurred for channel B.

(2) FLXAnFREIR.LTVB

Latest Transmit Violation Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

The flag indicates to the CPU that a latest transmission violation occurred on channel B.

(3) FLXAnFREIR.EDB

Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This bit is set whenever one of the flags FLXAnFRACS.SEDB, FLXAnFRACS.CEDB, FLXAnFRACS.CIB, FLXAnFRACS.SBVB changes from 0 to 1.

(4) FLXAnFREIR.TABA

Transmission Across Boundary Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

The flag indicates that a transmission across a slot boundary occurred for channel A.

(5) FLXAnFREIR.LTVA

Latest Transmit Violation Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

The flag indicates to the CPU that a latest transmit violation occurred on channel A.

(6) FLXAnFREIR.EDA

Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This bit is set whenever one of the flags FLXAnFRACS.SEDA, FLXAnFRACS.CEDA, FLXAnFRACS.CIA, FLXAnFRACS.SBVA changes from 0 to 1.

(7) FLXAnFREIR.MHF

Message Handler Constraints Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

The flag indicates a Message Handler constraint violation condition. It is set whenever one of the flags FLXAnFRMHDF.SNUA, FLXAnFRMHDF.SNUB, FLXAnFRMHDF.FNFA, FLXAnFRMHDF.FNFB, FLXAnFRMHDF.TBFA, FLXAnFRMHDF.TBFB, FLXAnFRMHDF.WAHP changes from 0 to 1.

(8) FLXAnFREIR.IOBA

Illegal Output buffer Access Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set by the CC when the Host requests the transfer of a message buffer from the Message RAM to the Output Buffer while FLXAnFROBCR.OBSYS is set to 1.

(9) FLXAnFREIR.IIBA

Illegal Input Buffer Access Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set by the CC when the Host wants to modify a message buffer via Input Buffer and one of the following conditions applies:

1. The CC is not in CONFIG or DEFAULT_CONFIG state and the Host writes to the Input Buffer Command Request register to modify the
   - Header section of message buffer 0 or 1 configured as a key slot
   - Header section of static message buffers with buffer number < FLXAnFRMRC.FDB while FLXAnFRMRC.SEC = "01_B"
   - Header section of any static or dynamic message buffer while FLXAnFRMRC.SEC = "$1x_B$"
   - Header and / or data section of any message buffer belonging to the receive FIFO
2. The Host writes to any register of the Input Buffer while FLXAnFRIBCR.IBSYH is set to 1.

(10) FLXAnFREIR.EFA

Empty FIFO Access Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set when the Host requests the transfer of a message from the receive FIFO via Output Buffer while the receive FIFO is empty.

(11) FLXAnFREIR.RFO

Receive FIFO Overrun Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

The flag is set by the CC when a receive FIFO overrun is detected. When a receive FIFO overrun occurs, the oldest message is overwritten with a new message message. The current state of the FIFO is monitored in register FLXAnFRFSR.

(12) FLXAnFREIR.AERR

Access error flag Flag

Writing 0 in this bit has no effect.

This bit is cleared when writing 1 to it.

Notifies of an access error.

When the AMR, ATBF1, or ATBF2 bit in the FLXAnFRMHDS register changes from 0 to 1, this bit is set to 1.

(13) FLXAnFREIR.CCL

CHI Command Locked Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

The flag indicates that the write access to the CHI command vector FLXAnFRSUCC1.CMD was not successful because the execution of the previous CHI command has not yet completed. In this case bit FLXAnFREIR.CNA is also set to 1.

(14) FLXAnFREIR.CCF

Clock Correction Failure Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set at the end of the communication cycle whenever one of the following errors occurs:

- Offset and / or rate correction incomplete
- Clock correction limit exceeded

The clock correction status is monitored in registers FLXAnFRCCEV and FLXAnFRSFS. A failure may occur during startup, therefore bit FLXAnFREIR.CCF should be set to 0 after the CC entered NORMAL_ACTIVE state.

(15) FLXAnFREIR.SFO

Sync Frame Overflow Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

It is set to 1 when either the number of sync frames received during the last communication cycle or the total number of different sync frame IDs received during the last two cycles exceeds the maximum number of sync frames as defined by FLXAnFRGTUC2.SNM.

(16) FLXAnFREIR.SFBM

Sync Frames Below Minimum Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set to 1 at the end of a cycle if the number of sync frames received during the last communication cycle was below the limit required for rate or offset correction term calculation (i.e. offset and / or rate correction incomplete). The clock correction status is monitored in FLXAnFRCCEV and FLXAnFRSFS.

This flag may be set to 1 during startup. Therefore this flag should be set to 0 by the Host after the CC entered NORMAL_ACTIVE state.

(17) FLXAnFREIR.CNA

Command Not Accepted Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

The flag indicates that the write access to the CHI command vector FLXAnFRSUCC1.CMD was not successful because the requested command was not valid in the actual POC state, or because the CHI command was locked (FLXAnFREIR.CCL = 1).

(18) FLXAnFREIR.PEMC

POC Error Mode Changed Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set to 1 whenever the error mode signaled by FLXAnFRCCEV.ERRM changes.

21.2.4.2 FLXAnFRSIR — FlexRay Status Interrupt Register

The flags are set when the CC detects one of the listed events. The flags remain set until cleared.

Access: FLXAnFRSIR can be read or written in 32-bit units.
FLXAnFRSIRL and FLXAnFRSIRH can be read or written in 16-bit units.
FLXAnFRSIRLL, FLXAnFRSIRLH, FLXAnFRSIRHL, and FLXAnFRSIRHH can be read or written in 8-bit units.

Address: FLXAnFRSIR: <FLXAn_base> + 0024_H,
FLXAnFRSIRL: <FLXAn_base> + 0024_H, FLXAnFRSIRH: <FLXAn_base> + 0026_H,
FLXAnFRSIRLL: <FLXAn_base> + 0024_H, FLXAnFRSIRLH: <FLXAn_base> + 0025_H,
FLXAnFRSIRHL: <FLXAn_base> + 0026_H, FLXAnFRSIRHH: <FLXAn_base> + 0027_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSB	WUPB	—	—	—	—	—	—	MTSA	WUPA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDS	MBSI	SUCS	SWE	TOBC	TIBC	TI1	TI0	NMVC	RFCL	RFNE	RXI	TXI	CYCS	CAS	WST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.16 FLXAnFRSIR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSB	MTS Received on Channel B Flag (vSS!ValidMTSB) 0: No MTS symbol received on channel B 1: MTS symbol received on channel B
24	WUPB	Wakeup Pattern Channel B Flag 0: No wakeup pattern received on channel B 1: Wakeup pattern received on channel B
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSA	MTS Received on Channel A Flag (vSS!ValidMTSA) 0: No MTS symbol received on channel A 1: MTS symbol received on channel A
16	WUPA	Wakeup Pattern Channel A Flag 0: No wakeup pattern received on channel A 1: Wakeup pattern received on channel A
15	SDS	Start of Dynamic Segment Flag 0: Dynamic segment not yet started 1: Dynamic segment started
14	MBSI	Message Buffer Status Interrupt Flag 0: No message buffer status change of message buffer with MBI = 1 1: Message buffer status of at least one message buffer with MBI = 1 has changed
13	SUCS	Startup Completed Successfully Flag 0: No startup completed successfully 1: Startup completed successfully
12	SWE	Stop Watch Event Flag 0: No Stop Watch Event 1: Stop Watch Event occurred

Table 21.16 FLXAnFRSIR Register Contents (2/2)

Bit Position	Bit Name	Function
11	TOBC	OBF Transfer Complete Flag 0: No transfer completed 1: Transfer between Message RAM and Output Buffer completed
10	TIBC	IBF Transfer Complete Flag 0: No transfer completed 1: Transfer between Input Buffer and Message RAM completed
9	TI1	Timer Interrupt 1 Flag 0: No timer interrupt 1 1: Timer interrupt 1 occurred
8	TI0	Timer Interrupt 0 Flag 0: No timer interrupt 0 1: Timer interrupt 0 occurred
7	NMVC	Network Management Vector Changed Flag 0: No change in the network management vector 1: Network management vector changed
6	RFCL	Receive FIFO Critical Level Flag 0: Receive FIFO below critical level 1: Receive FIFO critical level reached
5	RFNE	Receive FIFO Not Empty Flag 0: Receive FIFO is empty 1: Receive FIFO is not empty
4	RXI	Receive Interrupt Flag 0: No ND flag of a receive buffer with MBI = 1 has been set to 1 1: At least one ND flag of a receive buffer with MBI = 1 has been set to 1
3	TXI	Transmit Interrupt Flag 0: No frame transmitted from a transmit buffer with MBI = 1 1: At least one frame was transmitted from a transmit buffer with MBI = 1
2	CYCS	Cycle Start Interrupt Flag 0: No communication cycle started 1: Communication cycle started
1	CAS	Collision Avoidance Symbol Flag 0: No bit pattern matching the CAS symbol received 1: Bit pattern matching the CAS symbol received
0	WST	Wakeup Status Flag 0: Wakeup status unchanged 1: Wakeup status changed

(1) FLXAnFRSIR.MTSB

MTS Received on Channel B Flag (vSS!ValidMTSB)

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

Indicates that a Media Access Test Symbol was received on channel B during the preceding symbol window.

Updated by the CC for each channel at the end of the symbol window.

(2) FLXAnFRSIR.WUPB

Wakeup Pattern Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set to 1 when a wakeup pattern was received on channel B in either of the following states:

- WAKEUP
- READY
- STARTUP state

(3) FLXAnFRSIR.MTSA

MTS Received on Channel A Flag (vSS!ValidMTSA)

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

Indicates that a Media Access Test Symbol was received on channel A during the preceding symbol window.

Updated by the CC for each channel at the end of the symbol window.

(4) FLXAnFRSIR.WUPA

Wakeup Pattern Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set to 1 when a wakeup pattern was received on channel A in either of the following states:

- WAKEUP
- READY
- STARTUP state

(5) FLXAnFRSIR.SDS

Start of Dynamic Segment Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set by the CC when the dynamic segment starts.

(6) FLXAnFRSIR.MBSI

Message Buffer Status Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set by the CC when the message buffer status FLXAnFRMBS has changed and if bit MBI of that message buffer is 1 (see **Table 21.109**).

(7) FLXAnFRSIR.SUCS

Startup Completed Successfully Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set whenever a startup completed successfully and the CC entered NORMAL_ACTIVE state.

(8) FLXAnFRSIR.SWE

Stop Watch Event Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set after a stop watch activation when the current cycle counter and macrotick value are stored in the Stop Watch register (see **Section 21.2.5.4, FLXAnFRSTPW1 — FlexRay Stop Watch Register 1**).

(9) FLXAnFRSIR.TOBC

Transfer Output Buffer Completed Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set whenever a transfer from the Message RAM to the Output Buffer has completed and FLXAnFROBCR.OBSYS has been cleared by the Message Handler.

(10) FLXAnFRSIR.TIBC

Transfer Input Buffer Completed Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set whenever a transfer from the message RAM to the input buffer has completed and FLXAnFRIBCR.IBSYS has been cleared by the Message Handler.

(11) FLXAnFRSIR.TI1

Timer 1 Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set whenever timer 1 matches the conditions configured in register FLXAnFRT1C.

FlexRay timer 1 interrupt is generated when the T0IE bit in the FLXAnFROC register is effective.

(12) FLXAnFRSIR.TI0

Timer 0 Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set whenever timer 0 matches the conditions configured in register FLXAnFRT0C.

FlexRay timer 0 interrupt is generated when the T0IE bit in the FLXAnFROC register is effective.

(13) FLXAnFRSIR.NMVC

Network Management Vector Changed Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This is set when a change in the Network Management Vector occurs.

(14) FLXAnFRSIR.RFCL

Receive FIFO Critical Level Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set when the receive FIFO fill level FLXAnFRFSR.RFFL is equal to or greater than the critical level as configured by FLXAnFRFCL.CL.

(15) FLXAnFRSIR.RFNE

Receive FIFO Not Empty Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set by the CC when a received valid frame was stored into the empty receive FIFO. The current state of the receive FIFO is monitored in register FLXAnFRFSR.

(16) FLXAnFRSIR.RXI

Receive Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set by the CC whenever the set condition of a message buffers ND flag is fulfilled (see **Section 21.2.9.6, FLXAnFRNDATi — FlexRay New Data Register i (i = 1 to 4)**), and if bit MBI of that message buffer is set to 1 (see **Table 21.109**)

(17) FLXAnFRSIR.TXI

Transmit Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set by the CC at the end of frame transmission if bit MBI in the respective message buffer is set to 1 (see **Table 21.109**).

(18) FLXAnFRSIR.CYCS

Cycle Start Interrupt Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set by the CC when a communication cycle starts.

(19) FLXAnFRSIR.CAS

Collision Avoidance Symbol Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set by the CC during STARTUP state when a CAS or a potential CAS was received.

(20) FLXAnFRSIR.WST

Wakeup Status Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set when FLXAnFRCCSV.WSV changes to a value other than UNDEFINED.

21.2.4.3 FLXAnFREILS — FlexRay Error Interrupt Line Select Register

The FlexRay Error Interrupt Line Select register assigns an interrupt generated by a specific error interrupt flag from register FLXAnFREIR to one of the two modules interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt).

Access: FLXAnFREILS can be read or written in 32-bit units.
FLXAnFREILSL and FLXAnFREILSH can be read or written in 16-bit units.
FLXAnFREILSLL, FLXAnFREILSLH, FLXAnFREILSHL, and FLXAnFREILSHH can be read or written in 8-bit units.

Address: FLXAnFREILS: <FLXAn_base> + 0028_H,
FLXAnFREILSL: <FLXAn_base> + 0028_H, FLXAnFREILSH: <FLXAn_base> + $002A_H$,
FLXAnFREILSLL: <FLXAn_base> + 0028_H, FLXAnFREILSLH: <FLXAn_base> + 0029_H,
FLXAnFREILSHL: <FLXAn_base> + $002A_H$, FLXAnFREILSHH: <FLXAn_base> + $002B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBL	LTVBL	EDBL	—	—	—	—	—	TABAL	LTVAL	EDAL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFL	IOBAL	IIBAL	EFAL	RFOL	AERRL	CCLL	CCFL	SFOL	SFBML	CNAL	PEMCL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.17 FLXAnFREILS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABBL	Transmission Across Boundary Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
25	LTVBL	Latest Transmit Violation Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
24	EDBL	Error Detected on Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABAL	Transmission Across Boundary Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
17	LTVAL	Latest Transmit Violation Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
16	EDAL	Error Detected on Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MHFL	Message Handler Constraints Flag Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

Table 21.17 FLXAnFREILS Register Contents (2/2)

Bit Position	Bit Name	Function
10	IOBAL	Illegal Output Buffer Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
9	IIBAL	Illegal Input Buffer Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
8	EFAL	Empty FIFO Access Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
7	RFOL	Receive FIFO Overrun Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
6	AERRL	Access Error Interrupt Output Select Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
5	CCLL	CHI Command Locked Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
4	CCFL	Clock Correction Failure Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
3	SFOL	Sync Frame Overflow Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
2	SFBML	Sync Frames Below Minimum Interrupt Line Bit 0: Interrupt assigned to FlexRay 0interrupt 1: Interrupt assigned to FlexRay 1 interrupt
1	CNAL	Command Not Accepted Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
0	PEMCL	POC Error Mode Changed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

21.2.4.4 FLXAnFRSILS — FlexRay Status Interrupt Line Select Register

The FlexRay Status Interrupt Line Select register assign an interrupt generated by a specific status interrupt flag from register FLXAnFRSIR to one of the two module interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt).

Access: FLXAnFRSILS can be read or written in 32-bit units.
FLXAnFRSILSL and FLXAnFRSILSH can be read or written in 16-bit units.
FLXAnFRSILSLL, FLXAnFRSILSLH, FLXAnFRSILSHL, and FLXAnFRSILSHH can be read or written in 8-bit units.

Address: FLXAnFRSILS: <FLXAn_base> + $002C_H$,
FLXAnFRSILSL: <FLXAn_base> + $002C_H$, FLXAnFRSILSH: <FLXAn_base> + $002E_H$,
FLXAnFRSILSLL: <FLXAn_base> + $002C_H$, FLXAnFRSILSLH: <FLXAn_base> + $002D_H$,
FLXAnFRSILSHL: <FLXAn_base> + $002E_H$, FLXAnFRSILSHH: <FLXAn_base> + $002F_H$

Value after reset: $0303\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBL	WUPBL	—	—	—	—	—	—	MTSAL	WUPAL
Value after reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSL	MBSIL	SUCSL	SWEL	TOBCL	TIBCL	TI1L	TI0L	NMVCL	RFCLL	RFNEL	RXIL	TXIL	CYCSL	CASL	WSTL
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.18 FLXAnFRSILS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSBL	Media Access Test Symbol Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
24	WUPBL	Wakeup Pattern Channel B Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSAL	Media Access Test Symbol Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
16	WUPAL	Wakeup Pattern Channel A Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
15	SDSL	Start of Dynamic Segment Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
14	MBSIL	Message Buffer Status Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
13	SUCSL	Startup Completed Successfully Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

Table 21.18 FLXAnFRSILS Register Contents (2/2)

Bit Position	Bit Name	Function
12	SWEL	Stop Watch Event Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
11	TOBCL	Transfer Output Buffer Completed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
10	TIBCL	Transfer Input Buffer Completed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
9	TI1L	Timer 1 Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
8	TI0L	Timer 0 Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
7	NMVCL	Network Management Vector Changed Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
6	RFCLL	Receive FIFO Critical Level Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
5	RFNEL	Receive FIFO Not Empty Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
4	RXIL	Receive Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
3	TXIL	Transmit Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
2	CYCSL	Cycle Start Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
1	CASL	Collision Avoidance Symbol Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt
0	WSTL	Wakeup Status Interrupt Line Bit 0: Interrupt assigned to FlexRay 0 interrupt 1: Interrupt assigned to FlexRay 1 interrupt

21.2.4.5 FLXAnFREIES — FlexRay Error Interrupt Enable Set Register

The settings in the FlexRay Error Interrupt Enable Set register (FLXAnFREIES) and FlexRay Error Interrupt Enable Reset (FLXAnFREIER) register determine which status changes in the FlexRay Error Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFREIES and reset by writing to FLXAnFREIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 sets the interrupt enable bit.

Access: FLXAnFREIES can be read or written in 32-bit units.
FLXAnFREIESL and FLXAnFREIESH can be read or written in 16-bit units.
FLXAnFREIESLL, FLXAnFREIESLH, FLXAnFREIESHL, and FLXAnFREIESHH can be read or written in 8-bit units.

Address: FLXAnFREIES: <FLXAn_base> + 0030_H,
FLXAnFREIESL: <FLXAn_base> + 0030_H, FLXAnFREIESH: <FLXAn_base> + 0032_H,
FLXAnFREIESLL: <FLXAn_base> + 0030_H, FLXAnFREIESLH: <FLXAn_base> + 0031_H,
FLXAnFREIESHL: <FLXAn_base> + 0032_H, FLXAnFREIESHH: <FLXAn_base> + 0033_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBE	LTVBE	EDBE	—	—	—	—	—	TABAE	LTVAE	EDAE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFE	IOBAE	IIBAE	EFAE	RFOE	AERRE	CCLE	CCFE	SFOE	SFBME	CNAE	PEMCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.19 FLXAnFREIES Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABBE	Transmission Across Boundary Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
25	LTVBE	Latest Transmit Violation Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
24	EDBE	Error Detected on Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABAE	Transmission Across Boundary Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
17	LTVAE	Latest Transmit Violation Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 21.19 FLXAnFREIES Register Contents (2/2)

Bit Position	Bit Name	Function
16	EDAE	Error Detected on Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MHFE	Message Handler Constraints Flag Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
10	IOBAE	Illegal Output Buffer Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
9	IIBAE	Illegal Input Buffer Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
8	EFAE	Empty FIFO Access Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
7	RFOE	Receive FIFO Overrun Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
6	AERRE	Access Error Interrupt Enable Bit 0: Interrupt is disabled. 1: Interrupt is enabled.
5	CCLE	CHI Command Locked Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
4	CCFE	Clock Correction Failure Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
3	SFOE	Sync Frame Overflow Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
2	SFBME	Sync Frames Below Minimum Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CNAE	Command Not Accepted Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
0	PEMCE	POC Error Mode Changed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

21.2.4.6 FLXAnFREIER — FlexRay Error Interrupt Enable Reset Register

The settings in the FlexRay Error Interrupt Enable Set register (FLXAnFREIES) and FlexRay Error Interrupt Enable Reset (FLXAnFREIER) register determine which status changes in the FlexRay Error Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFREIES and reset by writing to FLXAnFREIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 clears the interrupt enable bit.

Access: FLXAnFREIER can be read or written in 32-bit units.
FLXAnFREIERL and FLXAnFREIERH can be read or written in 16-bit units.
FLXAnFREIERLL, FLXAnFREIERLH, FLXAnFREIERHL, and FLXAnFREIERHH can be read or written in 8-bit units.

Address: FLXAnFREIER: <FLXAn_base> + 0034_H,
FLXAnFREIERL: <FLXAn_base> + 0034_H, FLXAnFREIERH: <FLXAn_base> + 0036_H,
FLXAnFREIERLL: <FLXAn_base> + 0034_H, FLXAnFREIERLH: <FLXAn_base> + 0035_H,
FLXAnFREIERHL: <FLXAn_base> + 0036_H, FLXAnFREIERHH: <FLXAn_base> + 0037_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TABBD	LTVBD	EDBD	—	—	—	—	—	TABAD	LTVAD	EDAD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MHFD	IOBAD	IIBAD	EFAD	RFOD	AERRD	CCLD	CCFD	SFOD	SFBMD	CNAD	PEMCD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.20 FLXAnFREIER Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	TABBD	Transmission Across Boundary Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
25	LTVBD	Latest Transmit Violation Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
24	EDBD	Error Detected on Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	TABAD	Transmission Across Boundary Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
17	LTVAD	Latest Transmit Violation Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 21.20 FLXAnFREIER Register Contents (2/2)

Bit Position	Bit Name	Function
16	EDAD	Error Detected on Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MHFD	Message Handler Constraints Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
10	IOBAD	Illegal Output Buffer Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
9	IIBAD	Illegal Input Buffer Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
8	EFAD	Empty FIFO Access Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
7	RFOD	Receive FIFO Overrun Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
6	AERRD	Access Error Interrupt Disable Bit 0: Interrupt is disabled. 1: Interrupt is enabled.
5	CCLD	CHI Command Locked Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
4	CCFD	Clock Correction Failure Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
3	SFOD	Sync Frame Overflow Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
2	SFBMD	Sync Frames Below Minimum Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CNAD	Command Not Accepted Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
0	PEMCD	POC Error Mode Changed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

21.2.4.7 FLXAnFRSIES — FlexRay Status Interrupt Enable Set Register

The settings in the FlexRay Status Interrupt Enable Set register (FLXAnFRSIES) and FlexRay Status Interrupt Enable Reset (FLXAnFRSIER) register determine which status changes in the FlexRay Status Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFRSIES and reset by writing to FLXAnFRSIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 sets the interrupt enable bit.

Access: FLXAnFRSIES can be read or written in 32-bit units.
FLXAnFRSIESL and FLXAnFRSIESH and can be read or written in 16-bit units.
FLXAnFRSIESLL, FLXAnFRSIESLH, FLXAnFRSIESHL, and FLXAnFRSIESHH can be read or written in 8-bit units.

Address: FLXAnFRSIES: <FLXAn_base> + 0038_H,
FLXAnFRSIESL: <FLXAn_base> + 0038_H, FLXAnFRSIESH: <FLXAn_base> + $003A_H$,
FLXAnFRSIESLL: <FLXAn_base> + 0038_H, FLXAnFRSIESLH: <FLXAn_base> + 0039_H,
FLXAnFRSIESHL: <FLXAn_base> + $003A_H$, FLXAnFRSIESHH: <FLXAn_base> + $003B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBE	WUPBE	—	—	—	—	—	—	MTSAE	WUPAE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSE	MBSIE	SUCSE	SWEE	TOBCE	TIBCE	TI1E	TI0E	NMVCE	RFCLE	RFNEE	RXIE	TXIE	CYCSE	CASE	WSTE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.21 FLXAnFRSIES Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSBE	MTS Received on Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
24	WUPBE	Wakeup Pattern Channel B Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSAE	MTS Received on Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
16	WUPAE	Wakeup Pattern Channel A Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
15	SDSE	Start of Dynamic Segment Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 21.21 FLXAnFRSIES Register Contents (2/2)

Bit Position	Bit Name	Function
14	MBSIE	Message Buffer Status Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
13	SUCSE	Startup Completed Successfully Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
12	SWEE	Stop Watch Event Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
11	TOBCE	Transfer Output Buffer Completed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
10	TIBCE	Transfer Input Buffer Completed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
9	TI1E	Timer 1 Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
8	TI0E	Timer 0 Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
7	NMVCE	Network Management Vector Changed Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
6	RFCLE	Receive FIFO Critical Level Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
5	RFNEE	Receive FIFO Not Empty Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
4	RXIE	Receive Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
3	TXIE	Transmit Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
2	CYCSE	Cycle Start Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CASE	Collision Avoidance Symbol Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled
0	WSTE	Wakeup Status Interrupt Enable Bit 0: Interrupt disabled 1: Interrupt enabled

21.2.4.8 FLXAnFRSIER — FlexRay Status Interrupt Disable Register

The settings in the FlexRay Status Interrupt Enable Set register (FLXAnFRSIES) and FlexRay Status Interrupt Disable (FLXAnFRSIER) register determine which status changes in the FlexRay Status Interrupt Register will result in an interrupt.

The enable bits are set by writing to FLXAnFRSIES and reset by writing to FLXAnFRSIER. Reading from both addresses will result in the same value.

Writing 0 has no effect on the bit value.

Writing a 1 clears the interrupt enable bit.

Access: FLXAnFRSIER can be read or written in 32-bit units.
FLXAnFRSIERL and FLXAnFRSIERH can be read or written in 16-bit units.
FLXAnFRSIERLL, FLXAnFRSIERLH, FLXAnFRSIERHL, and FLXAnFRSIERHH can be read or written in 8-bit units.

Address: FLXAnFRSIER: <FLXAn_base> + $003C_H$,
FLXAnFRSIERL: <FLXAn_base> + $003C_H$, FLXAnFRSIERH: <FLXAn_base> + $003E_H$,
FLXAnFRSIERLL: <FLXAn_base> + $003C_H$, FLXAnFRSIERLH: <FLXAn_base> + $003D_H$,
FLXAnFRSIERHL: <FLXAn_base> + $003E_H$, FLXAnFRSIERHH: <FLXAn_base> + $003F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	MTSBD	WUPBD	—	—	—	—	—	—	MTSAD	WUPAD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDSD	MBSID	SUCSD	SWED	TOBCD	TIBCD	TI1D	TI0D	NMVCD	RFCLD	RFNED	RXID	TXID	CYCSD	CASD	WSTD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.22 FLXAnFRSIER Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	MTSBD	MTS Received on Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
24	WUPBD	Wakeup Pattern Channel B Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
23 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	MTSAD	MTS Received on Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
16	WUPAD	Wakeup Pattern Channel A Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
15	SDSD	Start of Dynamic Segment Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

Table 21.22 FLXAnFRSIER Register Contents (2/2)

Bit Position	Bit Name	Function
14	MBSID	Message Buffer Status Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
13	SUCSD	Startup Completed Successfully Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
12	SWED	Stop Watch Event Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
11	TOBCD	Transfer Output Buffer Completed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
10	TIBCD	Transfer Input Buffer Completed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
9	TI1D	Timer Interrupt 1 Disable Bit 0: Interrupt disabled 1: Interrupt enabled
8	TI0D	Timer Interrupt 0 Disable Bit 0: Interrupt disabled 1: Interrupt enabled
7	NMVCD	Network Management Vector Changed Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
6	RFCLD	Receive FIFO Critical Level Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
5	RFNED	Receive FIFO Not Empty Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
4	RXID	Receive Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
3	TXID	Transmit Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
2	CYCSD	Cycle Start Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
1	CASD	Collision Avoidance Symbol Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled
0	WSTD	Wakeup Status Interrupt Disable Bit 0: Interrupt disabled 1: Interrupt enabled

21.2.4.9 FLXAnFRILE — FlexRay Interrupt Line Enable Register

Each of the two module interrupt lines (FlexRay 0 interrupt, FlexRay 1 interrupt) can be enabled / disabled separately by programming bit FLXAnFRILE.EINT0 and FLXAnFRILE.EINT1.

Access: FLXAnFRILE can be read or written in 32-bit units.
FLXAnFRILEL can be read or written in 16-bit units.
FLXAnFRILELL can be read or written in 8-bit units.

Address: FLXAnFRILE: <FLXAn_base> + 0040_H,
FLXAnFRILEL: <FLXAn_base> + 0040_H,
FLXAnFRILELL: <FLXAn_base> + 0040_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EINT1	EINT0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 21.23 FLXAnFRILE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	EINT1	Enable FlexRay 1 Interrupt Line Bit 0: FlexRay 1 interrupt disabled 1: FlexRay 1 interrupt enabled
0	EINT0	Enable FlexRay 0 Interrupt Line Bit 0: FlexRay 0 interrupt disabled 1: FlexRay 0 interrupt enabled

21.2.5 FlexRay Timer Registers

21.2.5.1 FLXAnFRT0C — FlexRay Timer 0 Configuration Register

This register is an absolute timer. It specifies the point in time when a FlexRay timer 0 interrupt occurs as the values of cycle count and macrotick (MT). When the FlexRay timer 0 passes, FLXAnFRSIR.TI0 and FLXAnFROS.T0IS are set to 1. A timer 0 interrupt then occurs while the FLXAnFROC.T0IE bit is effective.

CAUTION

The configuration of timer 0 is compared against the macrotick counter value, there is no separate counter for timer 0.

Access: FLXAnFRT0C can be read or written in 32-bit units.
FLXAnFRT0CL and FLXAnFRT0CH can be read or written in 16-bit units.
FLXAnFRT0CLL, FLXAnFRT0CLH, FLXAnFRT0CHL, and FLXAnFRT0CHH can be read or written in 8-bit units.

Address: FLXAnFRT0C: <FLXAn_base> + 0044_H,
FLXAnFRT0CL: <FLXAn_base> + 0044_H, FLXAnFRT0CH: <FLXAn_base> + 0046_H,
FLXAnFRT0CLL: <FLXAn_base> + 0044_H, FLXAnFRT0CLH: <FLXAn_base> + 0045_H,
FLXAnFRT0CHL: <FLXAn_base> + 0046_H, FLXAnFRT0CHH: <FLXAn_base> + 0047_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T0MO[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	T0CC[6:0]							—	—	—	—	—	—	T0MS	T0RC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 21.24 FLXAnFRT0C Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	T0MO[13:0]	Timer 0 Macrotick Offset Bit Timer 0 Macrotick Offset
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	T0CC[6:0]	Timer 0 Cycle Code Bit Timer 0 Cycle Code
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	T0MS	Timer 0 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T0RC	Timer 0 Run Control Bit 0: Timer 0 halted 1: Timer 0 running

(1) FLXAnFRT0C.T0MO

Timer 0 Macrotick Offset Bit

Before reconfiguration of the timer, the timer has to be halted first by clearing FLXAnFRT0C.T0RC to 0.

Configures the macrotick offset from the beginning of the communication cycle where the interrupt is to occur. The FlexRay timer 0 interrupt occurs at this offset for each cycle of the cycle set.

(2) FLXAnFRT0C.T0CC

Timer 0 Cycle Code Bit

Before reconfiguration of the timer, the timer has to be halted first by clearing FLXAnFRT0C.T0RC to 0.

The 7-bit timer 0 cycle code determines the cycle set used for generation of the FlexRay timer 0 interrupt. For details about the configuration of the cycle code see **Section 21.3.8.2, Cycle Counter Filtering**.

(3) FLXAnFRT0C.T0MS

Timer 0 Mode Select Bit

Before reconfiguration of the timer, the timer has to be halted first by clearing FLXAnFRT0C.T0RC to 0.

Configures the timer run mode. In Single-shot mode the timer is deactivated when the timer configuration matches the configured cycle counter and macrotick value.

(4) FLXAnFRT0C.T0RC

Timer 0 Run Control Bit

Timer 0 can be activated (set FLXAnFRT0C.T0RC to 1) when the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state.

Timer 0 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

21.2.5.2 FLXAnFRT1C — FlexRay Timer 1 Configuration Register

This register is a relative timer. After the specified number of macroticks (MT) has expired, a FlexRay timer 1 interrupt is asserted. When the FlexRay timer 1 passes, FLXAnFRSIR.TI1 and FLXAnFROS.T1IS are set to 1. A timer 1 interrupt then occurs while the FLXAnFROC.T1IE bit is effective.

Access: FLXAnFRT1C can be read or written in 32-bit units.
FLXAnFRT1CL and FLXAnFRT1CH can be read or written in 16-bit units.
FLXAnFRT1CLL, FLXAnFRT1CHL, and FLXAnFRT1CHH can be read or written in 8-bit units.

Address: FLXAnFRT1C: <FLXAn_base> + 0048_H,
FLXAnFRT1CL: <FLXAn_base> + 0048_H, FLXAnFRT1CH: <FLXAn_base> + $004A_H$,
FLXAnFRT1CLL: <FLXAn_base> + 0048_H, FLXAnFRT1CHL: <FLXAn_base> + $004A_H$,
FLXAnFRT1CHH: <FLXAn_base> + $004B_H$

Value after reset: $0002\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T1MC[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	T1MS	T1RC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 21.25 FLXAnFRT1C Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	T1MC[13:0]	Timer 1 Macrotick Count Bit
15 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	T1MS	Timer 1 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T1RC	Timer 1 Run Control Bit 0: Timer 1 halted 1: Timer 1 running

(1) FLXAnFRT1C.T1MC

Timer 1 Macrotick Count Bit

Before reconfiguration of the timer, the timer has to be halted first by writing 0 to FLXAnFRT1C.T1RC.

Valid values are 2 to 16383 MT in continuous mode

Valid values are 1 to 16383 MT in single-shot mode

When the configured macrotick count is reached the FlexRay timer 1 interrupt is generated.

(2) FLXAnFRT1C.T1MS

Timer 1 Mode Select Bit

Before reconfiguration of the timer, the timer has to be halted first by writing 0 to FLXAnFRT1C.T1RC.

Configures the timer run mode. In Single-shot mode the timer is deactivated when the timer configuration matches the configured cycle counter and macrotick value.

(3) FLXAnFRT1C.T1RC

Timer 1 Run Control Bit

Timer 1 can be activated (set FLXAnFRT1C.T1RC to 1) as long as the POC is either in NORMAL_ACTIVE state or in NORMAL_PASSIVE state.

Timer 1 is deactivated when leaving NORMAL_ACTIVE state or NORMAL_PASSIVE state except for transitions between the two states.

21.2.5.3 FLXAnFRT2C — FlexRay Timer 2 Configuration Register

Access: FLXAnFRT2C can be read or written in 32-bit units.
FLXAnFRT2CL and FLXAnFRT2CH can be read or written in 16-bit units.
FLXAnFRT2CLL, FLXAnFRT2CLH, FLXAnFRT2CHL, and FLXAnFRT2CHH can be read or written in 8-bit units.

Address: FLXAnFRT2C: <FLXAn_base> + 0844_H
FLXAnFRT2CL: <FLXAn_base> + 0844_H, FLXAnFRT2CH: <FLXAn_base> + 0846_H,
FLXAnFRT2CLL: <FLXAn_base> + 0844_H, FLXAnFRT2CLH: <FLXAn_base> + 0845_H,
FLXAnFRT2CHL: <FLXAn_base> + 0846_H, FLXAnFRT2CHH: <FLXAn_base> + 0847_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T2MO[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	T2CC[6:0]							—	—	—	—	—	—	T2MS	T2RC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 21.26 FLXAnFRT2C Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	T2MO[13:0]	Timer 2 Macrotick Offset Bit Timer 2 Macrotick Offset
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	T2CC[6:0]	Timer 2 Cycle Code Bit Timer 2 Cycle Code
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	T2MS	Timer 2 Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	T2RC	Timer 2 Run Control Bit 0: Timer running 1: Timer halted

(1) FLXAnFRT2C.T2MO

Timer 2 Macrotick Offset Bit

The user can only write to these bits when FLXAnFRT2C.T2RC is 0.

Allowed range is 0 to FLXAnFRGTUC2.MPC.

These bits indicate the timer macrotick value defining the timer expiration condition.

(2) FLXAnFRT2C.T2CC

Timer 2 Cycle Code Bit

The user can only write to these bits when FLXAnFRT2C.T2RC is 0.

This bit indicates the cycle counter filter code defining the timer expiration condition. See **Section 21.3.8.2, Cycle Counter Filtering** for cycle filtering.

(3) FLXAnFRT2C.T2MS

Timer 2 Mode Select Bit

The user can only write to these bits when FLXAnFRT2C.T2RC is 0.

This bit indicates the operation mode of the Timer 2.

0: Single-shot mode

The timer is operating in the non-repetitive (single shot) mode. Once the configured expiration criteria are matching the timer will be automatically halted.

1: Continuous mode

The timer is operating in the repetitive (continuous) mode. The timer will expire every time the configured expiration criteria are matching. The timer is not halted.

(4) FLXAnFRT2C.T2RC

Timer 2 Run Control Bit

The user can only set this bit to 1 when the POC is in NORMAL_ACTIVE or NORMAL_PASSIVE state.

This bit indicates the activation state of the Timer 2.

When the expiration criteria are matching for a single shot timer, then this bit is cleared automatically and the Timer 2 is halted.

[Setting condition]

This bit is set by writing 1 to it when the POC is in NORMAL_ACTIVE or NORMAL_PASSIVE state.

[Clearing condition]

This bit is cleared when the POC state leaves the NORMAL_ACTIVE or NORMAL_PASSIVE state except for transitions between the two states.
This bit is cleared when the timer is operating in the non-repetitive (single shot) mode (FLXAnFRT2C.T2MS is 0) and the expiration criteria are matching.
This bit is cleared by writing 0 to it.

21.2.5.4 FLXAnFRSTPW1 — FlexRay Stop Watch Register 1

The stop watch is activated by the following trigger events.

- FlexRay 0 interrupt or FlexRay 1 interrupt
- Writing bit FLXAnFRSTPW1.SSWT to 1

At the first MT counter increment after the stop watch starts, the actual cycle counter and macrotick values are captured in register FLXAnFRSTPW1 while the slot counter values for channel A and B are captured in register FLXAnFRSTPW2.

Access: FLXAnFRSTPW1 can be read or written in 32-bit units.
FLXAnFRSTPW1L can be read or written in 16-bit units.
FLXAnFRSTPW1H is a read-only register that can be read in 16-bit units.
FLXAnFRSTPW1LL can be read or written in 8-bit units.
FLXAnFRSTPW1LH, FLXAnFRSTPW1HL, and FLXAnFRSTPW1HH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRSTPW1: <FLXAn_base> + $004C_H$,
FLXAnFRSTPW1L: <FLXAn_base> + $004C_H$, FLXAnFRSTPW1H: <FLXAn_base> + $004E_H$,
FLXAnFRSTPW1LL: <FLXAn_base> + $004C_H$, FLXAnFRSTPW1LH: <FLXAn_base> + $004D_H$,
FLXAnFRSTPW1HL: <FLXAn_base> + $004E_H$, FLXAnFRSTPW1HH: <FLXAn_base> + $004F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SMTV[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SCCV[5:0]						—	EINT1	EINT0	EETP	SSWT	EDGE	SWMS	ESWT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.27 FLXAnFRSTPW1 Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	SMTV[13:0]	Stop Watch Captured Macrotick Value
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	SCCV[5:0]	Stop Watch Captured Cycle Counter Value
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	EINT1	Enable FlexRay Interrupt 1 Trigger Bit 0: Stop watch trigger by FlexRay 1 interrupt disabled 1: FlexRay 1 interrupt event triggers stop watch
5	EINT0	Enable FlexRay 0 Interrupt Trigger Bit 0: Stop watch trigger by FlexRay 0 interrupt disabled 1: FlexRay interrupt 0 event triggers stop watch
4	EETP	Enable External Trigger Pin 0: Stop watch trigger via MCU input pin (FLXA0STPWT) disabled 1: A rising or falling edge on MCU input pin (FLXA0STPWT) trigger stop watch
3	SSWT	Software Stop Watch Trigger Bit 0: Software trigger reset 1: Stop watch activated by software trigger

Table 21.27 FLXAnFRSTPW1 Register Contents (2/2)

Bit Position	Bit Name	Function
2	EDGE	Stop Watch Trigger Edge Select Bit 0: Falling edge 1: Rising edge
1	SWMS	Stop Watch Mode Select Bit 0: Single-shot mode 1: Continuous mode
0	ESWT	Enable Hardware Stop Watch Trigger Bit 0: Stop watch trigger disabled 1: Stop watch trigger enabled

(1) FLXAnFRSTPW1.SMTV

Stop Watch Captured Macrotick Value

Indicates the state of the macrotick counter when the stop watch event occurred.

(2) FLXAnFRSTPW1.SCCV

Stop Watch Captured Cycle Counter Value

Indicates the state of the cycle counter when the stop watch event occurred.

(3) FLXAnFRSTPW1.EINT1

Enable FlexRay 1 Interrupt Trigger Bit

Enables stop watch trigger by FlexRay 1 interrupt when FLXAnFRSTPW1.ESWT = 1.

(4) FLXAnFRSTPW1.EINT0

Enable FlexRay 0 Interrupt Trigger Bit

Enables stop watch trigger by FlexRay 0 interrupt when FLXAnFRSTPW1.ESWT = 1.

(5) FLXAnFRSTPW1.EETP

Enable External Trigger Pin

Enables stop watch trigger event via MCU pin FLXAnSTPWT if FLXAnFRSTPW1.ESWT = ’1’.

(6) FLXAnFRSTPW1.SSWT

Software Stop Watch Trigger Bit

Bits FLXAnFRSTPW1.ESWT and FLXAnFRSTPW1.SSWT cannot be set to 1 simultaneously. In this case the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.

Writing 1 in this bit activates the stop watch. This bit is reset to 0 after the cycle count and slot count, and macrotick (MT) value are stored in the FlexRay stop watch register.

(7) FLXAnFRSTPW1.EDGE

Stop Watch Trigger Edge Select Bit

(8) FLXAnFRSTPW1.SWMS

Stop Watch Mode Select Bit

(9) FLXAnFRSTPW1.ESWT

Enable Stop Watch Trigger Bit

Bits FLXAnFRSTPW1.ESWT and FLXAnFRSTPW1.SSWT cannot be set to 1 simultaneously. In this case the write access to the register is ignored, and both bits keep their previous values. Either the external stop watch trigger or the software stop watch trigger may be used.

If enabled, a FlexRay 0 interrupt event or a FlexRay 1 interrupt event activates the stop watch.

In single-shot mode, this bit is reset to 0 after the cycle count and slot count, and macrotick (MT) value are stored in the FlexRay stop watch register.

21.2.5.5 FLXAnFRSTPW2 — FlexRay Stop Watch Register 2

Access: FLXAnFRSTPW2 is a read-only register that can be read in 32-bit units.
FLXAnFRSTPW2L and FLXAnFRSTPW2H are the read-only registers that can be read in 16-bit units.
FLXAnFRSTPW2LL, FLXAnFRSTPW2LH, FLXAnFRSTPW2HL, and FLXAnFRSTPW2HH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRSTPW2: <FLXAn_base> + 0050_H,
FLXAnFRSTPW2L: <FLXAn_base> + 0050_H, FLXAnFRSTPW2H: <FLXAn_base> + 0052_H,
FLXAnFRSTPW2LL: <FLXAn_base> + 0050_H, FLXAnFRSTPW2LH: <FLXAn_base> + 0051_H,
FLXAnFRSTPW2HL: <FLXAn_base> + 0052_H, FLXAnFRSTPW2HH: <FLXAn_base> + 0053_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SSCVB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SSCVA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.28 FLXAnFRSTPW2 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned.
26 to 16	SSCVB[10:0]	Stop Watch Captured Slot Counter Value Channel B
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	SSCVA[10:0]	Stop Watch Captured Slot Counter Value Channel A

(1) FLXAnFRSTPW2.SSCVB

Stop Watch Captured Slot Counter Value Channel B

Indicates the state of the slot counter value for channel B when the stop watch event occurred.

(2) FLXAnFRSTPW2.SSCVA

Stop Watch Captured Slot Counter Value Channel A

Indicates the state of the slot counter value for channel A when the stop watch event occurred.

21.2.6 CC Control Registers

This section describes the registers provided by the CC (Communication Controller) to allow the Host to control the operation of the CC. The FlexRay protocol specification requires the Host to write application configuration data in CONFIG state only. Please consider that the configuration registers are not locked for writing in DEFAULT_CONFIG state.

The configuration data is reset when DEFAULT_CONFIG state is entered after reset. To change POC state from DEFAULT_CONFIG to CONFIG state the Host has to apply CHI command CONFIG. If the Host wants the CC to leave CONFIG state, the Host has to execute the lock release sequence as described in **Section 21.2.3.3, FLXAnFRLCK — FlexRay Lock Register**.

21.2.6.1 FLXAnFRSUCC1 — FlexRay SUC Configuration Register 1

Access: FLXAnFRSUCC1 can be read or written in 32-bit units.
FLXAnFRSUCC1L and FLXAnFRSUCC1H can be read or written in 16-bit units.
FLXAnFRSUCC1LL, FLXAnFRSUCC1LH, FLXAnFRSUCC1HL, and FLXAnFRSUCC1HH can be read or written in 8-bit units.

Address: FLXAnFRSUCC1: <FLXAn_base> + 0080_H,
FLXAnFRSUCC1L: <FLXAn_base> + 0080_H, FLXAnFRSUCC1H: <FLXAn_base> + 0082_H,
FLXAnFRSUCC1LL: <FLXAn_base> + 0080_H, FLXAnFRSUCC1LH: <FLXAn_base> + 0081_H,
FLXAnFRSUCC1HL: <FLXAn_base> + 0082_H, FLXAnFRSUCC1HH: <FLXAn_base> + 0083_H

Value after reset: $0C40\ 1080_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	CCHB	CCHA	MTSB	MTSA	HCSE	TSM	WUCS	PTA[4:0]				
Value after reset	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSA[4:0]					—	TXSY	TXST	PBSY	—	—	—	CMD[3:0]			
Value after reset	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 21.29 FLXAnFRSUCC1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27	CCHB	Connected to Channel B Bit Configures pChannels 0: Not connected to channel B 1: Node connected to channel B (default after reset)
26	CCHA	Connected to Channel A Bit Configures pChannels 0: Not connected to channel A 1: Node connected to channel A (default after reset)
25	MTSB	Select Channel B for MTS Transmission Bit 0: Channel B disabled for MTS transmission 1: Channel B selected for MTS transmission
24	MTSA	Select Channel A for MTS Transmission Bit 0: Channel A disabled for MTS transmission 1: Channel A selected for MTS transmission

Table 21.29 FLXAnFRSUCC1 Register Contents (2/2)

Bit Position	Bit Name	Function
23	HCSE	Halt due to Clock Sync Error Bit Configures pAllowHaltDueToClock 0: CC will enter / remain in NORMAL_PASSIVE 1: CC will enter HALT state
22	TSM	Transmission Slot Mode Bit Configures pSingleSlotEnabled 0: ALL Slot Mode 1: SINGLE Slot Mode (value after hard reset)
21	WUCS	Wakeup Channel Select Bit Configures pWakeupChannel 0: Send wakeup pattern on channel A 1: Send wakeup pattern on channel B
20 to 16	PTA[4:0]	Passive to Active Bit Configures pAllowPassiveToActive
15 to 11	CSA[4:0]	Cold Start Attempts Bit Configures gColdStartAttempts
10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	TXSY	Transmit Sync Frame in Key Slot Bit Configures pKeySlotUsedForSync 0: No sync frame transmission in key slot, node is neither sync nor coldstart node 1: Key slot used to transmit sync frame, node is sync node
8	TXST	Transmit Startup Frame in Key Slot Bit Configures pKeySlotUsedForStartup 0: No startup frame transmission in key slot, node is non-coldstarter 1: Key slot used to transmit startup frame, node is leading or following coldstarter
7	PBSY	POC Busy Flag 0: POC not busy, FLXAnFRSUCC1.CMD writeable 1: POC is busy, FLXAnFRSUCC1.CMD locked
6 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	CMD[3:0]	CHI Command Vector Bit 0000_B: command_not_accepted 0001_B: CONFIG 0010_B: READY 0011_B: WAKEUP 0100_B: RUN 0101_B: ALL_SLOTS 0110_B: HALT 0111_B: FREEZE 1000_B: SEND_MTS 1001_B: ALLOW_COLDSTART 1010_B: RESET_STATUS_INDICATORS 1011_B: MONITOR_MODE 1100_B: CLEAR_RAMS others: reserved

(1) FLXAnFRSUCC1.CCHB

Connected to Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Configures whether the node is connected to channel B (pChannels).

(2) FLXAnFRSUCC1.CCHA

Connected to Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Configures whether the node is connected to channel A (pChannels).

(3) FLXAnFRSUCC1.MTSB

Select Channel B for MTS Transmission Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

FLXAnFRSUCC1.MTSB may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to FLXAnFRSUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in **Section 21.2.3.3, FLXAnFRLCK — FlexRay Lock Register**. This may be combined with CHI command SEND_MTS. If both bits FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB are set to 1, an MTS symbol will be transmitted on both channels when requested by writing FLXAnFRSUCC1.CMD = "1000_B".

The bit selects channel B for MTS symbol transmission.

(4) FLXAnFRSUCC1.MTSA

Select Channel A for MTS Transmission Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

FLXAnFRSUCC1.MTSA may also be changed outside DEFAULT_CONFIG or CONFIG state when the write to FLXAnFRSUCC1 register is directly preceded by the unlock sequence for the Configuration Lock Key as described in **Section 21.2.3.3, FLXAnFRLCK — FlexRay Lock Register**. This may be combined with CHI command SEND_MTS. If both bits FLXAnFRSUCC1.MTSA and FLXAnFRSUCC1.MTSB are set to 1, an MTS symbol will be transmitted on both channels when requested by writing FLXAnFRSUCC1.CMD = "1000_B".

The bit selects channel A for MTS symbol transmission.

(5) FLXAnFRSUCC1.HCSE

Halt due to Clock Sync Error Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Controls the transition to HALT state due to a clock synchronization error (pAllowHaltDueToClock).

(6) FLXAnFRSUCC1.TSM

Transmission Slot Mode Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Sets the value after transmission slot mode reset (pSingleSlotEnabled).

In SINGLE slot mode the CC may only transmit in the preconfigured key slot. The key slot ID is configured in the header section of message buffer 0 or message buffers 0 and 1 depending on bit FLXAnFRMRC.SPLM.

If FLXAnFRSUCC1.TSM = 1, message buffer 0 or message buffers 0 and 1 can be (re)configured in DEFAULT_CONFIG or CONFIG state only. In ALL slot mode the CC may transmit in all slots.

FLXAnFRSUCC1.TSM is a configuration bit which can only be set / reset by the Host.

The CC changes to ALL slot mode when the Host successfully applied the ALL_SLOTS command by writing FLXAnFRSUCC1.CMD = "0101_B" in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. The actual slot mode is monitored by FLXAnFRCCSV.SLM.

(7) FLXAnFRSUCC1.WUCS

Wakeup Channel Select Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

With this bit the Host selects the channel on which the CC sends the Wakeup pattern (pWakeupChannel).

(8) FLXAnFRSUCC1.PTA

Passive to Active Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 31 even / odd cycle pairs.

Defines the number of consecutive even / odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state (pAllowPassiveToActive).

If set to "00000_B" the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

(9) FLXAnFRSUCC1.CSA

Cold Start Attempts Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Must be identical in all nodes of a cluster.

Valid values are 2 to 31.

Configures the maximum number of attempts that a cold starting node is permitted to try to start up the network without receiving any valid response from another node (gColdStartAttempts).

(10) FLXAnFRSUCC1.TXSY

Transmit Sync Frame in Key Slot Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Defines whether the key slot is used to transmit sync frames (pKeySlotUsedForSync).

CAUTION

The protocol requires that both bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY are set for coldstart nodes.

(11) FLXAnFRSUCC1.TXST

Transmit Startup Frame in Key Slot Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Defines whether the key slot is used to transmit startup frames (pKeySlotUsedForStartup).

CAUTION

The protocol requires that both bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY are set for coldstart nodes.

(12) FLXAnFRSUCC1.PBSY

POC Busy Flag

Signals that the POC is busy and cannot accept a command from the Host. FLXAnFRSUCC1.CMD is locked against write accesses.

Set to 1 after reset during initialization of internal RAM blocks.

(13) FLXAnFRSUCC1.CMD

CHI Command Vector Bit

The Host may write any CHI command at any time, but certain commands are enabled only in certain POC states. If a command is not enabled, it will not be executed, the CHI command vector FLXAnFRSUCC1.CMD will be reset to "0000_B" = command_not_accepted, and flag FLXAnFREIR.CNA will be set to 1.

In general the Host must check FLXAnFRSUCC1.PBSY before writing a new CHI command.

In case the previous CHI command has not yet completed, FLXAnFREIR.CCL is set to 1 together with FLXAnFREIR.CNA; the CHI command needs to be repeated.

Except for HALT state, a POC state change command applied while the CC is already in the requested POC state neither causes a state change nor will FLXAnFREIR.CNA be set.

Reading FLXAnFRSUCC1.CMD shows whether the last CHI command was accepted. The actual POC state is monitored by FLXAnFRCCSV.POCS.

- command_not_accepted

  FLXAnFRSUCC1.CMD is reset to "0000_B" due to one of the following conditions:

  – Illegal command applied by the Host
  – Host applied command to leave CONFIG state without preceding config lock key
  – Host applied new command while execution of the previous Host command has not completed
  – Host writes command_not_accepted

When FLXAnFRSUCC1.CMD is cleared to "0000_B", FLXAnFREIR.CNA is set to 1, and if enabled an interrupt is generated.

Commands which are not accepted are not executed.

CONFIG command

Go to POC state CONFIG when called in POC states DEFAULT_CONFIG, or READY. When called in HALT state the CC transits to POC state DEFAULT_CONFIG. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to "0000_B" = command_not_accepted.

READY command

Go to POC state READY when called in POC states CONFIG, NORMAL_ACTIVE, NORMAL_PASSIVE, STARTUP, or WAKEUP. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to "0000_B" = command_not_accepted.

WAKEUP command

Go to POC state WAKEUP when called in POC state READY. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to "0000_B" = command_not_accepted.

RUN command

Go to POC state STARTUP when called in POC state READY. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to "0000_B" = command_not_accepted.

ALL_SLOTS command

Leave SINGLE slot mode and go to ALL-SLOTS mode after successful startup / integration at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to "0000_B" = command_not_accepted.

HALT command

Set halt request FLXAnFRCCSV.HRQ to 1 and go to POC state HALT at the next end of cycle when called in POC states NORMAL_ACTIVE or NORMAL_PASSIVE. When called in any other state, FLXAnFRSUCC1.CMD will be cleared to "0000_B" = command_not_accepted.

FREEZE command

Set the freeze status indicator FLXAnFRCCSV.FSI to 1 and go to POC state HALT immediately. Can be called from any state.

SEND_MTS command

Send single MTS symbol during the next following symbol window on the channel configured by FLXAnFRSUCC1.MTSA, FLXAnFRSUCC1.MTSB, when called in POC state NORMAL_ACTIVE after CC entered ALL slot mode (FLXAnFRCCSV.SLM = "11_B"). When called in any other state, or when called while a previously requested MTS has not yet been transmitted, FLXAnFRSUCC1.CMD will be reset to "0000_B" = command_not_accepted.

ALLOW_COLDSTART command

The command clears FLXAnFRCCSV.CSI to enable the coldstart of the node. When called in states DEFAULT_CONFIG, CONFIG, or HALT, FLXAnFRSUCC1.CMD will be cleared to "0000_B" = command_not_accepted. To enable the coldstart it is also required that both FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY are set.

RESET_STATUS_INDICATORS command

Clears status flags FLXAnFRCCSV.CSNI, FLXAnFRCCSV.CSAI, and FLXAnFRCCSV.WSV to their values after reset. May be called in POC states READY and STARTUP. When called in any other state, FLXAnFRSUCC1.CMD will be reset to "0000_B" = command_not_accepted.

CLEAR_RAMS command

Sets FLXAnFRMHDS.CRAM to 1 when called in DEFAULT_CONFIG or CONFIG state. When called in any other state, FLXAnFRSUCC1.CMD will be reset to "0000_B" = command_not_accepted.

FLXAnFRMHDS.CRAM is also set to 1 after reset. By setting FLXAnFRMHDS.CRAM all internal RAM blocks are initialized to zero. During the initialization of the RAMs, FLXAnFRSUCC1.PBSY will show POC busy. Access to the configuration and status registers is possible during execution of CHI command CLEAR_RAMS.

The initialization of the internal message RAM requires 2048 peripheral bus clock cycles. There should be no Host access to IBF or OBF during initialization of the internal RAM blocks after reset or after assertion of CHI command CLEAR_RAMS.

Before asserting CHI command CLEAR_RAMS, make sure that no transfer between Message RAM and IBF/OBF or Message RAM and Temporary Buffer RAM is ongoing and that the data transfer handler has no effect (FLXAnFRITS.ITS = 0 and FLXAnFROTS.OTS = 0). This command also clears the Message Buffer Status registers FLXAnFRMHDS, FLXAnFRLDTS, FLXAnFRFSR, FLXAnFRMHDF, FLXAnFRTXRQ1/2/3/4, FLXAnFRNDAT1/2/3/4, and FLXAnFRMBSC1/2/3/4.

CAUTIONS

1. All accepted commands with exception of CLEAR_RAMS and SEND_MTS will cause a change of the POC state in the FlexRay sample clock domain after at most 8 cycles of the slower of the two clocks "bus clock" and "FlexRay sample clock", assuming that POC was not busy when the command was applied and that no POC state change was forced by bus activity in that time frame. Reading register FLXAnFRCCSV will show data that is additionally delayed by synchronization from the FlexRay sample clock domain to the bus clock domain. The maximum additional delay is 12 cycles of the slower of the two clocks "bus clock" and "FlexRay sample clock".
2. When the communication is stopped by the FREEZE or READY command and the communication is restarted as Leading Coldstart node, the startup frame may not be transmitted on cycle 0 depending on the internal condition of the FlexRay module. This case occurs when the Startup frame is set in one of slot 1 to slot 7.
   This does not occur in ColdStart after a hardware reset.
   Even if this occurs, the second trial of ColdStart will succeed. ColdStart time becomes longer, but ColdStart will not be affected by the occurrence.
   To avoid this, allocate the Startup/Sync frame in the static slot 8 or higher.

Table 21.30 below references the CHI commands from the FlexRay Protocol Specification (Section 2.1.1.1, Table 2.2) to the FlexRay CHI command vector FLXAnFRSUCC1.CMD.

Table 21.30 Reference to CHI Host Command Summary from FlexRay Protocol Specification

CHI command	Where processed (POC States)	CHI Command Vector CMD
ALL_SLOTS	POC: normal active, POC: normal passive	ALL_SLOTS
ALLOW_COLDSTART	All except POC: default config, POC: config, POC: halt	ALLOW_COLDSTART
CONFIG	POC: default config, POC: ready	CONFIG
CONFIG_COMPLETE	POC: config	Unlock sequence & READY
DEFAULT_CONFIG	POC: halt	CONFIG
FREEZE	All	FREEZE
HALT	POC: normal active, POC: normal passive	HALT
READY	All except POC: default config, POC: config, POC: ready, POC: halt	READY
RUN	POC: ready	RUN
WAKEUP	POC: ready	WAKEUP

21.2.6.2 FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2

Access: FLXAnFRSUCC2 can be read or written in 32-bit units.
FLXAnFRSUCC2L and FLXAnFRSUCC2H can be read or written in 16-bit units.
FLXAnFRSUCC2LL, FLXAnFRSUCC2LH, FLXAnFRSUCC2HL, and FLXAnFRSUCC2HH can be read or written in 8-bit units.

Address: FLXAnFRSUCC2: <FLXAn_base> + 0084_H
FLXAnFRSUCC2L: <FLXAn_base> + 0084_H, FLXAnFRSUCC2H: <FLXAn_base> + 0086_H,
FLXAnFRSUCC2LL: <FLXAn_base> + 0084_H, FLXAnFRSUCC2LH: <FLXAn_base> + 0085_H,
FLXAnFRSUCC2HL: <FLXAn_base> + 0086_H, FLXAnFRSUCC2HH: <FLXAn_base> + 0087_H

Value after reset: 0100 0504_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	LTN[3:0]				—	—	—	LT[20:16]				
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LT[15:0]															
Value after reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.31 FLXAnFRSUCC2 Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
27 to 24	LTN[3:0]	Listen Timeout Noise Bit Configures (gListenNoise - 1)
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 0	LT[20:0]	Listen Timeout Bit Configures pdListenTimeout

(1) FLXAnFRSUCC2.LTN

Listen Timeout Noise Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

The range for gListenNoise is 2 to 16.

FLXAnFRSUCC2.LTN must be configured identical in all nodes of a cluster.

Configures the upper limit for startup and wakeup listen timeout in the presence of noise expressed as a multiple of pdListenTimeout.

CAUTION

The wakeup / startup noise timeout is calculated as follows:
pdListenTimeout × gListenNoise = FLXAnFRSUCC2.LT × (FLXAnFRSUCC2.LTN + 1)

(2) FLXAnFRSUCC2.LT

Listen Timeout Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

The range for pdListenTimeout is 1284 to 1283846 µT.

Configures wakeup / startup listen time out in µT.

21.2.6.3 FLXAnFRSUCC3 — FlexRay SUC Configuration Register 3

Access: FLXAnFRSUCC3 can be read or written in 32-bit units.
FLXAnFRSUCC3L can be read or written in 16-bit units.
FLXAnFRSUCC3LL can be read or written in 8-bit units.

Address: FLXAnFRSUCC3: <FLXAn_base> + 0088_H,
FLXAnFRSUCC3L: <FLXAn_base> + 0088_H,
FLXAnFRSUCC3LL: <FLXAn_base> + 0088_H

Value after reset: $0000\ 0011_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	WCF[3:0]				WCP[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.32 FLXAnFRSUCC3 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 4	WCF[3:0]	Maximum Without Clock Correction Fatal Bit (transition to HALT state) Configures gMaxWithoutClockCorrectionFatal
3 to 0	WCP[3:0]	Maximum Without Clock Correction Passive Bit (transition to NORMAL_PASSIVE state) Configures gMaxWithoutClockCorrectionPassive

(1) FLXAnFRSUCC3.WCF

Maximum Without Clock Correction Fatal Bit (transition to HALT state)

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 15.

Must be identical in all nodes of a cluster.

Defines the number of consecutive even / odd cycle pairs with missing clock correction conditions that will cause a transition from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

CAUTION

The transition to HALT state is prevented if FLXAnFRSUCC1.HCSE is not set.

(2) FLXAnFRSUCC3.WCP

Maximum Without Clock Correction Passive Bit (transition to NORMAL_PASSIVE state)

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 15.

Must be identical in all nodes of a cluster.

Defines the number of consecutive even / odd cycle pairs with missing clock correction conditions that will cause a transition from NORMAL_ACTIVE to NORMAL_PASSIVE state.

21.2.6.4 FLXAnFRNEMC — FlexRay NEM Configuration Register

Access: FLXAnFRNEMC can be read or written in 32-bit units.
FLXAnFRNEMCL can be read or written in 16-bit units.
FLXAnFRNEMCLL can be read or written in 8-bit units.

Address: FLXAnFRNEMC: <FLXAn_base> + $008C_H$,
FLXAnFRNEMCL: <FLXAn_base> + $008C_H$,
FLXAnFRNEMCLL: <FLXAn_base> + $008C_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	NML[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 21.33 FLXAnFRNEMC Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	NML[3:0]	Network Management Vector Length Bit Configures gNetworkManagementVectorLength

(1) FLXAnFRNEMC.NML

Network Management Vector Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 12 bytes.

The configured length must be identical in all nodes of a cluster.

These bits configure the length of the NM vector in bytes.

21.2.6.5 FLXAnFRPRTC1 — FlexRay PRT Configuration Register 1

Access: FLXAnFRPRTC1 can be read or written in 32-bit units.
FLXAnFRPRTC1L and FLXAnFRPRTC1H can be read or written in 16-bit units.
FLXAnFRPRTC1LL, FLXAnFRPRTC1LH, FLXAnFRPRTC1HL, and FLXAnFRPRTC1HH can be read or written in 8-bit units.

Address: FLXAnFRPRTC1: <FLXAn_base> + 0090_H,
FLXAnFRPRTC1L: <FLXAn_base> + 0090_H, FLXAnFRPRTC1H: <FLXAn_base> + 0092_H,
FLXAnFRPRTC1LL: <FLXAn_base> + 0090_H, FLXAnFRPRTC1LH: <FLXAn_base> + 0091_H,
FLXAnFRPRTC1HL: <FLXAn_base> + 0092_H, FLXAnFRPRTC1HH: <FLXAn_base> + 0093_H

Value after reset: $084C\ 0633_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RWP[5:0]						—	RXW[8:0]								
Value after reset	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRP[1:0]		SPP[1:0]		—	CASM[6:0]							TSST[3:0]			
Value after reset	0	0	0	0	0	1	1	0	0	0	1	1	0	0	1	1
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.34 FLXAnFRPRTC1 Register Contents

Bit Position	Bit Name	Function
31 to 26	RWP[5:0]	Repetitions of Tx Wakeup Pattern Bit Configures pWakeupPattern
25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24 to 16	RXW[8:0]	Wakeup Symbol Receive Window Length Bit Configures gdWakeupSymbolRxWindow
15, 14	BRP[1:0]	Baud Rate Prescaler Bit Configures gdSampleClockPeriod and pSamplesPerMicrotick 00_B = 10 Mbps 01_B = 5 Mbps 10_B = 2.5 Mbps 11_B = 2.5 Mbps
13, 12	SPP[1:0]	Strobe Point Position Bit Configures Strobe point position 00_B = Sample 5 01_B = Sample 4 10_B = Sample 6 11_B = Sample 5
11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 4	CASM[6:0]	Collision Avoidance Symbol Max Bit Configures gdCASRxLowMax
3 to 0	TSST[3:0]	Transmission Start Sequence Transmitter Bit Configures gdTSSTransmitter

(1) FLXAnFRPRTC1.RWP

Repetitions of Tx Wakeup Pattern Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 63.

Configures the number of repetitions (sequences) of the Tx wakeup symbol.

(2) FLXAnFRPRTC1.RXW

Wakeup Symbol Receive Window Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 76 to 301.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the received wakeup pattern.

(3) FLXAnFRPRTC1.BRP

Baud Rate Prescaler Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

The Baud Rate Prescaler configures the baud rate on the FlexRay bus. The baud rates listed below are valid with a sample clock set to 80 MHz. One bit time always consists of 8 samples independent of the configured baud rate.

00_B = 10 MBit/s

gdSampleClockPeriod = 12.5 ns = 1 × “sample clock”

pSamplesPerMicrotick = 2 (1 µT = 25 ns)

01_B = 5 MBit/s

gdSampleClockPeriod = 25 ns = 2 × “sample clock”

pSamplesPerMicrotick = 1 (1 µT = 25 ns)

10_B, 11_B = 2.5 MBit/s

gdSampleClockPeriod = 50 ns = 4 × “sample clock”

pSamplesPerMicrotick = 1 (1 µT = 50 ns)

(4) FLXAnFRPRTC1.SPP

Strobe Point Position Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Defines the sample count value for strobing. The strobed bit value is set to the voted value when the sample count is incremented to the value configured by FLXAnFRPRTC1.SPP.

CAUTION

The current revision 2.1 of the FlexRay protocol requires that FLXAnFRPRTC1.SPP = "00_B". The alternate strobe point positions could be used to compensate for asymmetries in the physical layer.

(5) FLXAnFRPRTC1.CASM

Collision Avoidance Symbol Max Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

CASM[6] is fixed to 1.

Valid values are 67 to 99.

Configures the upper limit of the acceptance window for a collision avoidance symbol (CAS).

(6) FLXAnFRPRTC1.TSST

Transmission Start Sequence Transmitter Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 3 to 15.

Must be identical in all nodes of a cluster.

Configures the duration of the Transmission Start Sequence (TSS) in terms of bit times (1 bit time = 4 µT = 100ns @ 10Mbps).

21.2.6.6 FLXAnFRPRTC2 — FlexRay PRT Configuration Register 2

Access: FLXAnFRPRTC2 can be read or written in 32-bit units.
FLXAnFRPRTC2L and FLXAnFRPRTC2H can be read or written in 16-bit units.
FLXAnFRPRTC2LL, FLXAnFRPRTC2LH, FLXAnFRPRTC2HL, and FLXAnFRPRTC2HH can be read or written in 8-bit units.

Address: FLXAnFRPRTC2:<FLXAn_base> + 0094_H,
FLXAnFRPRTC2L: <FLXAn_base> + 0094_H, FLXAnFRPRTC2H: <FLXAn_base> + 0096_H,
FLXAnFRPRTC2LL: <FLXAn_base> + 0094_H, FLXAnFRPRTC2LH: <FLXAn_base> + 0095_H,
FLXAnFRPRTC2HL: <FLXAn_base> + 0096_H, FLXAnFRPRTC2HH: <FLXAn_base> + 0097_H

Value after reset: $0F2D\ 0A0E_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	TXL[5:0]						TXI[7:0]							
Value after reset	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RXL[5:0]						—	—	RXI[5:0]					
Value after reset	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.35 FLXAnFRPRTC2 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 24	TXL[5:0]	Wakeup Symbol Transmit Low Bit Configures gdWakeupSymbolTxLow
23 to 16	TXI[7:0]	Wakeup Symbol Transmit Idle Bit Configures gdWakeupSymbolTxIdle
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	RXL[5:0]	Wakeup Symbol Receive Low Bit Configures gdWakeupSymbolRxLow
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RXI[5:0]	Wakeup Symbol Rx Idle Bit Configures gdWakeupSymbolRxIdle

(1) FLXAnFRPRTC2.TXL

Wakeup Symbol Transmit Low Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 15 to 60.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to transmit the low phase of the wakeup symbol.

(2) FLXAnFRPRTC2.TXI

Wakeup Symbol Transmit Idle Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 45 to 180.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to transmit the idle phase of the wakeup symbol.

(3) FLXAnFRPRTC2.RXL

Wakeup Symbol Receive Low Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 10 to 55.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the low phase of the received wakeup symbol.

(4) FLXAnFRPRTC2.RXI

Wakeup Symbol Rx Idle Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 14 to 59.

Must be identical in all nodes of a cluster.

Configures the number of bit times used by the node to test the duration of the idle phase of the received wakeup symbol.

21.2.6.7 FLXAnFRMHDC — FlexRay MHD Configuration Register

Access: FLXAnFRMHDC can be read or written in 32-bit units.
FLXAnFRMHDCL and FLXAnFRMHDCH can be read or written in 16-bit units.
FLXAnFRMHDCLL, FLXAnFRMHDCHL, and FLXAnFRMHDCHH can be read or written in 8-bit units.

Address: FLXAnFRMHDC: <FLXAn_base> + 0098_H,
FLXAnFRMHDCL: <FLXAn_base> + 0098_H, FLXAnFRMHDCH: <FLXAn_base> + $009A_H$,
FLXAnFRMHDCLL: <FLXAn_base> + 0098_H, FLXAnFRMHDCHL: <FLXAn_base> + $009A_H$,
FLXAnFRMHDCHH: <FLXAn_base> + $009B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	SLT[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SFDL[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.36 FLXAnFRMHDC Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 16	SLT[12:0]	Start of Latest Transmit Bit Configures pLatestTx
15 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	SFDL[6:0]	Static Frame Data Length Bit Configures gPayloadLengthStatic

(1) FLXAnFRMHDC.SLT

Start of Latest Transmit Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7981.

Configures the latest maximum minislot value allowed before inhibiting frame transmission in the dynamic segment of the cycle. There is no transmission in dynamic segment if FLXAnFRMHDC.SLT is set to zero.

(2) FLXAnFRMHDC.SFDL

Static Frame Data Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 127.

The payload length must be identical in all nodes of a cluster.

Configures the cluster-wide payload length for all frames sent in the static segment in double bytes.

21.2.6.8 FLXAnFRGTUC1 — FlexRay GTU Configuration Register 1

Access: FLXAnFRGTUC1 can be read or written in 32-bit units.
FLXAnFRGTUC1L and FLXAnFRGTUC1H can be read or written in 16-bit units.
FLXAnFRGTUC1LL, FLXAnFRGTUC1LH, and FLXAnFRGTUC1HL can be read or written in 8-bit units.

Address: FLXAnFRGTUC1: <FLXAn_base> + $00A0_H$,
FLXAnFRGTUC1L: <FLXAn_base> + $00A0_H$, FLXAnFRGTUC1H: <FLXAn_base> + $00A2_H$,
FLXAnFRGTUC1LL: <FLXAn_base> + $00A0_H$, FLXAnFRGTUC1LH: <FLXAn_base> + $00A1_H$,
FLXAnFRGTUC1HL: <FLXAn_base> + $00A2_H$

Value after reset: $0000\ 0280_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	UT[19:16]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UT[15:0]															
Value after reset	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.37 FLXAnFRGTUC1 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 0	UT[19:0]	Setting of Communication Cycle in Microticks Bit Configures pMicroPerCycle

(1) FLXAnFRGTUC1.UT

Setting of Communication Cycle in Microticks Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 640 to 640000 µT.

Configures the duration of the communication cycle in microticks.

21.2.6.9 FLXAnFRGTUC2 — FlexRay GTU Configuration Register 2

Access: FLXAnFRGTUC2 can be read or written in 32-bit units.
FLXAnFRGTUC2L and FLXAnFRGTUC2H can be read or written in 16-bit units.
FLXAnFRGTUC2LL, FLXAnFRGTUC2LH, and FLXAnFRGTUC2HL can be read or written in 8-bit units.

Address: FLXAnFRGTUC2: <FLXAn_base> + $00A4_H$,
FLXAnFRGTUC2L: <FLXAn_base> + $00A4_H$, FLXAnFRGTUC2H: <FLXAn_base> + $00A6_H$,
FLXAnFRGTUC2LL: <FLXAn_base> + $00A4_H$, FLXAnFRGTUC2LH: <FLXAn_base> + $00A5_H$,
FLXAnFRGTUC2HL: <FLXAn_base> + $00A6_H$

Value after reset: $0002\ 000A_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	SNM[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MPC[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.38 FLXAnFRGTUC2 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 16	SNM[3:0]	Sync Node Max Bit
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	MPC[13:0]	Setting of Communication Cycle in Macrotick Bit Configures gMacroPerCycle

(1) FLXAnFRGTUC2.SNM

Sync Node Max Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 15.

Must be identical in all nodes of a cluster.

Indicates the maximum number of frames with sync frame indicator bit SYN set to 1.

(2) FLXAnFRGTUC2.MPC

Setting of Communication Cycle in Macrotick Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 10 to 16000 MT.

The cycle length must be identical in all nodes of a cluster.

Configures the duration of one communication cycle in macroticks.

21.2.6.10 FLXAnFRGTUC3 — FlexRay GTU Configuration Register 3

Access: FLXAnFRGTUC3 can be read or written in 32-bit units.
FLXAnFRGTUC3L and FLXAnFRGTUC3H can be read or written in 16-bit units.
FLXAnFRGTUC3LL, FLXAnFRGTUC3LH, FLXAnFRGTUC3HL, and FLXAnFRGTUC3HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC3: <FLXAn_base> + $00A8_H$,
FLXAnFRGTUC3L: <FLXAn_base> + $00A8_H$, FLXAnFRGTUC3H: <FLXAn_base> + $00AA_H$,
FLXAnFRGTUC3LL: <FLXAn_base> + $00A8_H$, FLXAnFRGTUC3LH: <FLXAn_base> + $00A9_H$,
FLXAnFRGTUC3HL: <FLXAn_base> + $00AA_H$, FLXAnFRGTUC3HH: <FLXAn_base> + $00AB_H$

Value after reset: $0202\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MIOB[6:0]							—	MIOA[6:0]						
Value after reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UIOB[7:0]								UIOA[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.39 FLXAnFRGTUC3 Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 24	MIOB[6:0]	Macrotick Initial Offset Channel B Bit Configures pMacroInitialOffset[B]
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	MIOA[6:0]	Macrotick Initial Offset Channel A Bit Configures pMacroInitialOffset[A]
15 to 8	UIOB[7:0]	Microtick Initial Offset Channel B Bit Configures pMicroInitialOffset[B]
7 to 0	UIOA[7:0]	Microtick Initial Offset Channel A Bit Configures pMicroInitialOffset[A]

(1) FLXAnFRGTUC3.MIOB

Macrotick Initial Offset Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 72 MT.

Must be identical in all nodes of a cluster.

Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration.

(2) FLXAnFRGTUC3.MIOA

Macrotick Initial Offset Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 72 MT.

Must be identical in all nodes of a cluster.

Configures the number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point based on the nominal macrotick duration.

(3) FLXAnFRGTUC3.UIOB

Microtick Initial Offset Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 240 µT.

Configures the number of microticks between the actual time reference point on channel B and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on pDelayCompensation [B] and therefore has to be set for each channel independently.

(4) FLXAnFRGTUC3.UIOA

Microtick Initial Offset Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 240 µT.

Configures the number of microticks between the actual time reference point on channel A and the subsequent macrotick boundary of the secondary time reference point. The parameter depends on pDelayCompensation [A] and therefore has to be set for each channel independently.

21.2.6.11 FLXAnFRGTUC4 — FlexRay GTU Configuration Register 4

For details about configuration of FLXAnFRGTUC4.NIT and FLXAnFRGTUC4.OCS see **Section 21.3.2.5, Configuration of NIT Start and Offset Correction Start**.

Access: FLXAnFRGTUC4 can be read or written in 32-bit units.
FLXAnFRGTUC4L and FLXAnFRGTUC4H can be read or written in 16-bit units.
FLXAnFRGTUC4LL, FLXAnFRGTUC4LH, FLXAnFRGTUC4HL, and FLXAnFRGTUC4HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC4: <FLXAn_base> + $00AC_H$,
FLXAnFRGTUC4L: <FLXAn_base> + $00AC_H$, FLXAnFRGTUC4H: <FLXAn_base> + $00AE_H$,
FLXAnFRGTUC4LL: <FLXAn_base> + $00AC_H$, FLXAnFRGTUC4LH: <FLXAn_base> + $00AD_H$,
FLXAnFRGTUC4HL: <FLXAn_base> + $00AE_H$, FLXAnFRGTUC4HH: <FLXAn_base> + $00AF_H$

Value after reset: $0008\ 0007_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	OCS[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	NIT[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.40 FLXAnFRGTUC4 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 16	OCS[13:0]	Offset Correction Start Bit Configures (gOffsetCorrectionStart - 1)
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	NIT[13:0]	Network Idle Time Start Bit Configures (gMacroPerCycle -gdNIT - 1)

(1) FLXAnFRGTUC4.OCS

Offset Correction Start Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 8 to 15998 MT.

For cluster consisting of E-Ray implementations only, it is sufficient to program FLXAnFRGTUC4.OCS = FLXAnFRGTUC4.NIT + 1.

Must be identical in all nodes of a cluster.

Determines the start position of the offset correction within the NIT phase, calculated from start of cycle.

(2) FLXAnFRGTUC4.NIT

Network Idle Time Start Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 7 to 15997 MT.

Must be identical in all nodes of a cluster.

Configures the starting point of the Network Idle Time NIT from the beginning of the communication cycle expressed in terms of macroticks from the beginning of the cycle. The start of NIT is recognized if Macrotick = gMacroPerCycle -gdNIT -1 and the increment pulse of Macrotick is set.

21.2.6.12 FLXAnFRGTUC5 — FlexRay GTU Configuration Register 5

Access: FLXAnFRGTUC5 can be read or written in 32-bit units.
FLXAnFRGTUC5L and FLXAnFRGTUC5H can be read or written in 16-bit units.
FLXAnFRGTUC5LL, FLXAnFRGTUC5LH, FLXAnFRGTUC5HL, and FLXAnFRGTUC5HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC5: <FLXAn_base> + $00B0_H$,
FLXAnFRGTUC5L: <FLXAn_base> + $00B0_H$, FLXAnFRGTUC5H: <FLXAn_base> + $00B2_H$,
FLXAnFRGTUC5LL: <FLXAn_base> + $00B0_H$, FLXAnFRGTUC5LH: <FLXAn_base> + $00B1_H$,
FLXAnFRGTUC5HL: <FLXAn_base> + $00B2_H$, FLXAnFRGTUC5HH: <FLXAn_base> + $00B3_H$

Value after reset: $0E00\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DEC[7:0]								—	—	—	CDD[4:0]				
Value after reset	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCB[7:0]								DCA[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.41 FLXAnFRGTUC5 Register Contents

Bit Position	Bit Name	Function
31 to 24	DEC[7:0]	Decoding Correction Bit Configures pDecodingCorrection
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	CDD[4:0]	Cluster Drift Damping Bit Configures pClusterDriftDamping
15 to 8	DCB[7:0]	Delay Compensation Channel B Bit Configures pDelayCompensation[B]
7 to 0	DCA[7:0]	Delay Compensation Channel A Bit Configures pDelayCompensation[A]

(1) FLXAnFRGTUC5.DEC

Decoding Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 14 to 143 µT.

Configures the decoding correction value in microticks used to determine the primary time reference point.

(2) FLXAnFRGTUC5.CDD

Cluster Drift Damping Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 20 µT.

Configures the cluster drift damping value in microticks used in clock synchronization to minimize accumulation of rounding errors.

(3) FLXAnFRGTUC5.DCB

Delay Compensation Channel B Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 200 µT.

Used to compensate for reception delays on channel B. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05µs. In practice, the minimum of the propagation delays of all sync nodes should be applied.

(4) FLXAnFRGTUC5.DCA

Delay Compensation Channel A Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 200 µT.

Used to compensate for reception delays on channel A. This covers assumed propagation delay up to cPropagationDelayMax for microticks in the range of 0.0125 to 0.05µs. In practice, the minimum of the propagation delays of all sync nodes should be applied.

21.2.6.13 FLXAnFRGTUC6 — FlexRay GTU Configuration Register 6

Access: FLXAnFRGTUC6 can be read or written in 32-bit units.
FLXAnFRGTUC6L and FLXAnFRGTUC6H can be read or written in 16-bit units.
FLXAnFRGTUC6LL, FLXAnFRGTUC6LH, FLXAnFRGTUC6HL, and FLXAnFRGTUC6HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC6: <FLXAn_base> + $00B4_H$,
FLXAnFRGTUC6L: <FLXAn_base> + $00B4_H$, FLXAnFRGTUC6H: <FLXAn_base> + $00B6_H$,
FLXAnFRGTUC6LL: <FLXAn_base> + $00B4_H$, FLXAnFRGTUC6LH: <FLXAn_base> + $00B5_H$,
FLXAnFRGTUC6HL: <FLXAn_base> + $00B6_H$, FLXAnFRGTUC6HH: <FLXAn_base> + $00B7_H$

Value after reset: $0002\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MOD[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ASR[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.42 FLXAnFRGTUC6 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 16	MOD[10:0]	Maximum Oscillator Drift Bit Configures pdMaxDrift
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	ASR[10:0]	Accepted Startup Range Bit Configures pdAcceptedStartupRange

(1) FLXAnFRGTUC6.MOD

Maximum Oscillator Drift Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1923 µT.

Configures the maximum drift offset between two nodes that operate with unsynchronized clocks over one communication cycle in µT.

(2) FLXAnFRGTUC6.ASR

Accepted Startup Range Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 1875 µT.

Configures the number of microticks constituting the expanded range of measured deviation for startup frames during integration.

21.2.6.14 FLXAnFRGTUC7 — FlexRay GTU Configuration Register 7

Access: FLXAnFRGTUC7 can be read or written in 32-bit units.
FLXAnFRGTUC7L and FLXAnFRGTUC7H can be read or written in 16-bit units.
FLXAnFRGTUC7LL, FLXAnFRGTUC7LH, FLXAnFRGTUC7HL, and FLXAnFRGTUC7HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC7: <FLXAn_base> + $00B8_H$,
FLXAnFRGTUC7L: <FLXAn_base> + $00B8_H$, FLXAnFRGTUC7H: <FLXAn_base> + $00BA_H$,
FLXAnFRGTUC7LL: <FLXAn_base> + $00B8_H$, FLXAnFRGTUC7LH: <FLXAn_base> + $00B9_H$,
FLXAnFRGTUC7HL: <FLXAn_base> + $00BA_H$, FLXAnFRGTUC7HH: <FLXAn_base> + $00BB_H$

Value after reset: $0002\ 0004_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	NSS[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SSL[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.43 FLXAnFRGTUC7 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25 to 16	NSS[9:0]	Number of Static Slots Bit Configures gNumberOfStaticSlots
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9 to 0	SSL[9:0]	Static Slot Length Bit Configures gdStaticSlot

(1) FLXAnFRGTUC7.NSS

Number of Static Slots Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1023.

The number of static slots must be identical in all nodes of a cluster.

Configures the number of static slots in a cycle.

(2) FLXAnFRGTUC7.SSL

Static Slot Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 4 to 659 MT.

The static slot length must be identical in all nodes of a cluster.

Configures the length of a static slot in macroticks.

21.2.6.15 FLXAnFRGTUC8 — FlexRay GTU Configuration Register 8

Access: FLXAnFRGTUC8 can be read or written in 32-bit units.
FLXAnFRGTUC8L and FLXAnFRGTUC8H can be read or written in 16-bit units.
FLXAnFRGTUC8LL, FLXAnFRGTUC8HL, and FLXAnFRGTUC8HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC8: <FLXAn_base> + $00BC_H$,
FLXAnFRGTUC8L: <FLXAn_base> + $00BC_H$,FLXAnFRGTUC8H: <FLXAn_base> + $00BE_H$,
FLXAnFRGTUC8LL: <FLXAn_base> + $00BC_H$,FLXAnFRGTUC8HL: <FLXAn_base> + $00BE_H$,
FLXAnFRGTUC8HH: <FLXAn_base> + $00BF_H$

Value after reset: $0000\ 0002_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	NMS[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	MSL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.44 FLXAnFRGTUC8 Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28 to 16	NMS[12:0]	Number of Minislots Bit Configures gNumberOfMinislots
15 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	MSL[5:0]	Minislot Length Bit Configures gdMinislot

(1) FLXAnFRGTUC8.NMS

Number of Minislots Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7986.

The number of minislots must be identical in all nodes of a cluster.

Configures the number of minislots within the dynamic segment of a cycle.

(2) FLXAnFRGTUC8.MSL

Minislot Length Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 63 MT.

The minislot length must be identical in all nodes of a cluster.

Configures the length of a minislot in macroticks.

21.2.6.16 FLXAnFRGTUC9 — FlexRay GTU Configuration Register 9

Access: FLXAnFRGTUC9 can be read or written in 32-bit units.
FLXAnFRGTUC9L and FLXAnFRGTUC9H can be read or written in 16-bit units.
FLXAnFRGTUC9LL, FLXAnFRGTUC9LH, and FLXAnFRGTUC9HL can be read or written in 8-bit units.

Address: FLXAnFRGTUC9: <FLXAn_base> + $00C0_H$,
FLXAnFRGTUC9L: <FLXAn_base> + $00C0_H$, FLXAnFRGTUC9H: <FLXAn_base> + $00C2_H$,
FLXAnFRGTUC9LL: <FLXAn_base> + $00C0_H$, FLXAnFRGTUC9LH: <FLXAn_base> + $00C1_H$,
FLXAnFRGTUC9HL: <FLXAn_base> + $00C2_H$

Value after reset: $0000\ 0101_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DSI[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MAPO[4:0]					—	—	APO[5:0]					
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.45 FLXAnFRGTUC9 Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17, 16	DSI[1:0]	Dynamic Slot Idle Phase Bit Configures gdDynamicSlotIdlePhase
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 8	MAPO[4:0]	Minislot Action Point Offset Bit Configures gdMinislotActionPointOffset
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	APO[5:0]	Action Point Offset Bit Configures gdActionPointOffset

(1) FLXAnFRGTUC9.DSI

Dynamic Slot Idle Phase Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 2.

Must be identical in all nodes of a cluster.

Configures the duration of the dynamic slot idle phase in the number of minislots. The duration has to be greater or equal than the idle detection time.

(2) FLXAnFRGTUC9.MAPO

Minislot Action Point Offset Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 31 MT.

Must be identical in all nodes of a cluster.

Configures the action point offset in macroticks within the minislots of the dynamic segment.

(3) FLXAnFRGTUC9.APO

Action Point Offset Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 1 to 63 MT.

Must be identical in all nodes of a cluster.

Configures the action point offset in macroticks within static slots and symbol window.

21.2.6.17 FLXAnFRGTUC10 — FlexRay GTU Configuration Register 10

Access: FLXAnFRGTUC10 can be read or written in 32-bit units.
FLXAnFRGTUC10L and FLXAnFRGTUC10H can be read or written in 16-bit units.
FLXAnFRGTUC10LL, FLXAnFRGTUC10LH, FLXAnFRGTUC10HL, and FLXAnFRGTUC10HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC10: <FLXAn_base> + $00C4_H$,
FLXAnFRGTUC10L: <FLXAn_base> + $00C4_H$, FLXAnFRGTUC10H: <FLXAn_base> + $00C6_H$,
FLXAnFRGTUC10LL: <FLXAn_base> + $00C4_H$, FLXAnFRGTUC10LH: <FLXAn_base> + $00C5_H$,
FLXAnFRGTUC10HL: <FLXAn_base> + $00C6_H$, FLXAnFRGTUC10HH: <FLXAn_base> + $00C7_H$

Value after reset: $0002\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	MRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MOC[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.46 FLXAnFRGTUC10 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 16	MRC[10:0]	Maximum Rate Correction Bit Configures pRateCorrectionOut
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 0	MOC[13:0]	Maximum Offset Correction Bit Configures pOffsetCorrectionOut

(1) FLXAnFRGTUC10.MRC

Maximum Rate Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 2 to 1923 µT.

Configures the maximum permitted rate correction value to be applied by the internal clock synchronization algorithm. The CC checks the internal rate correction value against the maximum rate correction value (absolute value).

(2) FLXAnFRGTUC10.MOC

Maximum Offset Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 5 to 15266 µT.

Configures the maximum permitted offset correction value (absolute value) to be applied by the internal clock synchronization algorithm (absolute value). The CC checks the internal offset correction value against the maximum offset correction value.

21.2.6.18 FLXAnFRGTUC11 — FlexRay GTU Configuration Register 11

Access: FLXAnFRGTUC11 can be read or written in 32-bit units.
FLXAnFRGTUC11L and FLXAnFRGTUC11H can be read or written in 16-bit units.
FLXAnFRGTUC11LL, FLXAnFRGTUC11LH, FLXAnFRGTUC11HL, and FLXAnFRGTUC11HH can be read or written in 8-bit units.

Address: FLXAnFRGTUC11: <FLXAn_base> + $00C8_H$,
FLXAnFRGTUC11L: <FLXAn_base> + $00C8_H$, FLXAnFRGTUC11H: <FLXAn_base> + $00CA_H$,
FLXAnFRGTUC11LL: <FLXAn_base> + $00C8_H$, FLXAnFRGTUC11LH: <FLXAn_base> + $00C9_H$,
FLXAnFRGTUC11HL: <FLXAn_base> + $00CA_H$, FLXAnFRGTUC11HH: <FLXAn_base> + $00CB_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ERC[2:0]			—	—	—	—	—	EOC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ERCC[1:0]		—	—	—	—	—	—	EOCC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 21.47 FLXAnFRGTUC11 Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26 to 24	ERC[2:0]	External Rate Correction Bit Configures pExternRateCorrection
23 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18 to 16	EOC[2:0]	External Offset Correction Bit Configures pExternOffsetCorrection
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9, 8	ERCC[1:0]	External Rate Correction Control Bit Configures vExternRateControl 00_B: External rate correction is prohibited. 01_B: External rate correction is prohibited. 10_B: Subtract 11_B: Add
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	EOCC[1:0]	External Offset Correction Control Bit Configures vExternOffsetControl 00_B: External offset correction is prohibited. 01_B: External offset correction is prohibited. 10_B: Subtract 11_B: Add

(1) FLXAnFRGTUC11.ERC

External Rate Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7 µT.

Configures the external rate correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted / added from / to the calculated rate correction value. The value is applied during NIT.

(2) FLXAnFRGTUC11.EOC

External Offset Correction Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Valid values are 0 to 7 µT.

Configures the external offset correction value in microticks to be applied by the internal clock synchronization algorithm. The value is subtracted / added from / to the calculated offset correction value. The value is applied during NIT.

(3) FLXAnFRGTUC11.ERCC

External Rate Correction Control Bit

Should be modified only outside NIT (Network Idle Time).

By writing to FLXAnFRGTUC11.ERCC the external rate correction is enabled as specified below.

00_B = External rate correction is prohibited.

01_B = External rate correction is prohibited.

10_B = Subtract

External rate correction value subtracted from calculated rate correction value

11_B = Add

External rate correction value added to calculated rate correction value

(4) FLXAnFRGTUC11.EOCC

External Offset Correction Control Bit

Should be modified only outside NIT (Network Idle Time).

By writing to FLXAnFRGTUC11.EOCC the external offset correction is enabled as specified below.

00_B = External offset correction is prohibited.

01_B = External offset correction is prohibited.

10_B = Subtract

External offset correction value subtracted from calculated offset correction value

11_B = Add

External offset correction value added to calculated offset correction value

21.2.7 CC Status Registers

During 8/16-bit accesses to status variables coded with more than 8/16-bit, the variable might be updated by the CC between two accesses (non-atomic read accesses).

21.2.7.1 FLXAnFRCCSV — FlexRay CC Status Vector Register

Access: FLXAnFRCCSV is a read-only register that can be read in 32-bit units.
FLXAnFRCCSVL and FLXAnFRCCSVH are the read-only registers that can be read in 16-bit units.
FLXAnFRCCSVLL, FLXAnFRCCSVLH, FLXAnFRCCSVHL, and FLXAnFRCCSVHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRCCSV: <FLXAn_base> + 0100_H,
FLXAnFRCCSVL: <FLXAn_base> + 0100_H, FLXAnFRCCSVH: <FLXAn_base> + 0102_H,
FLXAnFRCCSVLL: <FLXAn_base> + 0100_H, FLXAnFRCCSVLH: <FLXAn_base> + 0101_H,
FLXAnFRCCSVHL: <FLXAn_base> + 0102_H, FLXAnFRCCSVHH: <FLXAn_base> + 0103_H

Value after reset: $0010\ 4000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	PSL[5:0]						RCA[4:0]					WSV[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	CSI	CSAI	CSNI	—	—	SLM[1:0]		HRQ	FSI	POCS[5:0]					
Value after reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.48 FLXAnFRCCSV Register Contents (1/2)

Bit	Symbol	Function
31, 30	Reserved	When read, the value after reset is returned.
29 to 24	PSL[5:0]	POC Status Log Flag Status of FLXAnFRCCSV.POCS immediately before entering HALT state.
23 to 19	RCA[4:0]	Remaining Coldstart Attempts Flag Indicates vRemainingColdstartAttempts
18 to 16	WSV[2:0]	Wakeup Status Flag Indicates vPOC!WakeupStatus 000_B: UNDEFINED 001_B: RECEIVED_HEADER 010_B: RECEIVED_WUP 011_B: COLLISION_HEADER 100_B: COLLISION_WUP 101_B: COLLISION_UNKNOWN 110_B: TRANSMITTED 111_B: Reserved
15	Reserved	When read, the value after reset is returned.
14	CSI	Cold Start Inhibit Flag Indicates vColdStartInhibit 0: Cold starting of node enabled 1: Cold starting of node disabled
13	CSAI	Coldstart Abort Indicator Flag
12	CSNI	Coldstart Noise Indicator Flag Indicates vPOC!ColdstartNoise
11, 10	Reserved	When read, the value after reset is returned.

Table 21.48 FLXAnFRCCSV Register Contents (2/2)

Bit	Symbol	Function
9, 8	SLM[1:0]	Slot Mode Flag Indicates vPOC!SlotMode 00_B: SINGLE 01_B: reserved 10_B: ALL_PENDING 11_B: ALL
7	HRQ	Halt Request Flag Indicates vPOC!CHIHaltRequest
6	FSI	Freeze Status Indicator Flag Indicates vPOC!Freeze
5 to 0	POCS[5:0]	Protocol Operation Control Status Flag

(1) FLXAnFRCCSV.PSL

POC Status Log Flag

Set the value of FLXAnFRCCSV.POCS immediately before entering HALT state.

Set to HALT when FREEZE command is applied during HALT state and FLXAnFRCCSV.FSI is not already set i.e. the HALT state was not reached by FREEZE command.

Reset to "000000_B" when leaving HALT state.

(2) FLXAnFRCCSV.RCA

Remaining Coldstart Attempts Flag

Indicates the number of remaining coldstart attempts (vRemainingColdstartAttempts).

The value after a reset of FLXAnFRCCSV.RCA during CONFIG and DEFAULT_CONFIG state is also FLXAnFRSUCC1.CSA.

The RUN command resets this counter to the maximum number of coldstart attempts as configured by FLXAnFRSUCC1.CSA.

(3) FLXAnFRCCSV.WSV

Wakeup Status Flag

Indicates the status of the current wakeup attempt (vPOC!WakeupStatus).

Reset to 0 when entering Wakeup state, by CHI command RESET_STATUS_INDICATORS, or by transition from DEFAULT_CONFIG to CONFIG state

000_B = UNDEFINED

Wakeup not yet executed by the CC.

001_B = RECEIVED_HEADER

Set when the CC finishes wakeup due to the reception of a frame header without coding violation on either channel in WAKEUP_LISTEN state.

010_B = RECEIVED_WUP

Set when the CC finishes wakeup due to the reception of a valid wakeup pattern on the configured wakeup channel in WAKEUP_LISTEN state.

011_B = COLLISION_HEADER

Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid header on either channel.

100_B = COLLISION_WUP

Set when the CC stops wakeup due to a detected collision during wakeup pattern transmission by receiving a valid wakeup pattern on the configured wakeup channel.

101_B = COLLISION_UNKNOWN

Set when the CC stops wakeup by leaving WAKEUP_DETECT state after expiration of the wakeup timer without receiving a valid wakeup pattern or a valid frame header.

110_B = TRANSMITTED

Set when the CC has successfully completed the transmission of the wakeup pattern.

111_B = reserved

(4) FLXAnFRCCSV.CSI

Cold Start Inhibit Flag

Indicates whether the node is disabled from cold starting (vColdStartInhibit).

The flag is set to 1 whenever the POC enters READY state due to CHI command READY.

The flag has to be cleared under control of the Host by CHI command ALLOW_COLDSTART (FLXAnFRSUCC1.CMD = "1001_B").

(5) FLXAnFRCCSV.CSAI

Coldstart Abort Indicator Flag

Indicates that coldstart was aborted.

Cleared by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

(6) FLXAnFRCCSV.CSNI

Coldstart Noise Indicator Flag

Indicates that the cold start procedure occurred under noisy conditions (vPOC!ColdstartNoise).

Cleared by CHI command RESET_STATUS_INDICATORS or by transition from HALT to DEFAULT_CONFIG state or from READY to STARTUP state.

(7) FLXAnFRCCSV.SLM

Slot Mode Flag

Indicates the actual slot mode of the POC (vPOC!SlotMode) in states READY, WAKEUP, STARTUP, NORMAL_ACTIVE, and NORMAL_PASSIVE.

Default value is SINGLE. Changes to ALL, depending on FLXAnFRSUCC1.TSM.

In NORMAL_ACTIVE or NORMAL_PASSIVE state the CHI command ALL_SLOTS will change the slot mode from SINGLE over ALL_PENDING to ALL.

Set FLXAnFRSUCC1.TSM to SINGLE except for NORMAL_ACTIVE or NORMAL_PASSIVE.

(8) FLXAnFRCCSV.HRQ

Halt Request Flag

Indicates that a request from the Host has been received to halt the POC at the end of the communication cycle (vPOC!CHIHaltRequest).

Cleared by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(9) FLXAnFRCCSV.FSI

Freeze Status Indicator Flag

Indicates that the POC has entered the HALT state due to CHI command FREEZE (CMD = 0111_B) or due to an error condition requiring an immediate POC halt (vPOC!Freeze).

Reset by transition from HALT to DEFAULT_CONFIG state.

(10) FLXAnFRCCSV.POCS

Protocol Operation Control Status Flag

Indicates the current state of operation of the CC Protocol Operation Control

$00\ 0000_B$ = DEFAULT_CONFIG state

$00\ 0001_B$ = READY state

$00\ 0010_B$ = NORMAL_ACTIVE state

$00\ 0011_B$ = NORMAL_PASSIVE state

$00\ 0100_B$ = HALT state

$00\ 1111_B$ = CONFIG state

Indicates the current state of operation of the POC in the wakeup path

$01\ 0000_B$ = WAKEUP_STANDBY state

$01\ 0001_B$ = WAKEUP_LISTEN state

$01\ 0010_B$ = WAKEUP_SEND state

$01\ 0011_B$ = WAKEUP_DETECT state

Indicates the current state of operation of the POC in the startup path

$10\ 0000_B$ = STARTUP_PREPARE state

$10\ 0001_B$ = COLDSTART_LISTEN state

$10\ 0010_B$ = COLDSTART_COLLISION_RESOLUTION state

$10\ 0011_B$ = COLDSTART_CONSISTENCY_CHECK state

$10\ 0100_B$ = COLDSTART_GAP state

$10\ 0101_B$ = COLDSTART_JOIN State

$10\ 0110_B$ = INTEGRATION_COLDSTART_CHECK state

$10\ 0111_B$ = INTEGRATION_LISTEN state

$10\ 1000_B$ = INTEGRATION_CONSISTENCY_CHECK state

$10\ 1001_B$ = INITIALIZE_SCHEDULE state

$10\ 1010_B$ = ABORT_STARTUP state

$10\ 1011_B$ = STARTUP_SUCCESS state

others = reserved

21.2.7.2 FLXAnFRCCEV — FlexRay CC Error Vector Register

Access: FLXAnFRCCEV is a read-only register that can be read in 32-bit units.
FLXAnFRCCEVL is a read-only register that can be read in 16-bit units.
FLXAnFRCCEVLL and FLXAnFRCCEVLH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRCCEV: <FLXAn_base> + 0104_H,
FLXAnFRCCEVL: <FLXAn_base> + 0104_H,
FLXAnFRCCEVLL: <FLXAn_base> + 0104_H, FLXAnFRCCEVLH: <FLXAn_base> + 0105_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PTAC[4:0]					ERRM[1:0]		—	—	CCFC[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.49 FLXAnFRCCEV Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned.
12 to 8	PTAC[4:0]	Passive to Active Count Flag Indicates vAllowPassiveToActive
7, 6	ERRM[1:0]	Error Mode Flag Indicates vPOC!ErrorMode 00_B: ACTIVE 01_B: PASSIVE 10_B: COMM_HALT 11_B: reserved
5, 4	Reserved	When read, the value after reset is returned.
3 to 0	CCFC[3:0]	Clock Correction Failed Counter Indicates vClockCorrectionFailed

(1) FLXAnFRCCEV.PTAC

Passive to Active Count Flag

Indicates the number of consecutive even / odd cycle pairs that have passed with valid rate and offset correction terms, while the node is waiting to transit from NORMAL_PASSIVE state to NORMAL_ACTIVE state. The transition takes place when FLXAnFRCCEV.PTAC equals FLXAnFRSUCC1.PTA -1.

Cleared by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(2) FLXAnFRCCEV.ERRM

Error Mode Flag

Indicates the current error mode of the POC (vPOC!ErrorMode).

Cleared by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

(3) FLXAnFRCCEV.CCFC

Clock Correction Failed Counter

Indicates the clock correction failed counter value of the POC (vClockCorrectionFailed).

The Clock Correction Failed Counter is incremented by one at the end of any odd communication cycle where either the missing offset correction error or missing rate correction error are active.

The Clock Correction Failed Counter is reset to 0 at the end of an odd communication cycle if neither the offset correction failed nor the rate correction failed errors are active.

The Clock Correction Failed Counter stops at 15.

Cleared by transition from HALT to DEFAULT_CONFIG state or when entering READY state.

21.2.7.3 FLXAnFRSCV — FlexRay Slot Counter Value Register

Access: FLXAnFRSCV is a read-only register that can be read in 32-bit units.
FLXAnFRSCVL and FLXAnFRSCVH are the read-only registers that can be read in 16-bit units.
FLXAnFRSCVLL, FLXAnFRSCVLH, FLXAnFRSCVHL, and FLXAnFRSCVHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRSCV: <FLXAn_base> + 0110_H,
FLXAnFRSCVL: <FLXAn_base> + 0110_H, FLXAnFRSCVH: <FLXAn_base> + 0112_H,
FLXAnFRSCVLL: <FLXAn_base> + 0110_H, FLXAnFRSCVLH: <FLXAn_base> + 0111_H,
FLXAnFRSCVHL: <FLXAn_base> + 0112_H, FLXAnFRSCVHH: <FLXAn_base> + 0113_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SCCB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SCCA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.50 FLXAnFRSCV Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned.
26 to 16	SCCB[10:0]	Slot Counter Channel B Indicates vSlotCounter[B]
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	SCCA[10:0]	Slot Counter Channel A Indicates vSlotCounter[A]

(1) FLXAnFRSCV.SCCB

Slot Counter Channel B

Indicates the current slot counter value on channel B (vSlotCounter[B]). The value is incremented by the CC and initialized at the start of a communication cycle.

Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRSCV.SCCA

Slot Counter Channel A

Indicates the current slot counter value on channel A (vSlotCounter[A]). The value is incremented by the CC and initialized at the start of a communication cycle.

Cleared when leaving CONFIG state or when entering STARTUP state.

21.2.7.4 FLXAnFRMTCCV — FlexRay Macrotick and Cycle Counter Value Register

Access: FLXAnFRMTCCV is a read-only register that can be read in 32-bit units.
FLXAnFRMTCCVL and FLXAnFRMTCCVH are the read-only registers that can be read in 16-bit units.
FLXAnFRMTCCVLL, FLXAnFRMTCCVLH, and FLXAnFRMTCCVHL are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRMTCCV: <FLXAn_base> + 0114_H,
FLXAnFRMTCCVL: <FLXAn_base> + 0114_H, FLXAnFRMTCCVH: <FLXAn_base> + 0116_H,
FLXAnFRMTCCVLL: <FLXAn_base> + 0114_H,FLXAnFRMTCCVLH: <FLXAn_base> + 0115_H,
FLXAnFRMTCCVHL: <FLXAn_base> + 0116_H,

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	CCV[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	MTV[13:0]													
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.51 FLXAnFRMTCCV Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 16	CCV[5:0]	Cycle Counter Value Indicates vCycleCounter
15, 14	Reserved	When read, the value after reset is returned.
13 to 0	MTV[13:0]	Macrotick Value Indicates vMacrotick

(1) FLXAnFRMTCCV.CCV

Cycle Counter Value

Indicates the current cycle counter value (vCycleCounter). The value is incremented by the CC at the start of a communication cycle.

Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRMTCCV.MTV

Macrotick Value

Indicates the current macrotick value (vMacrotick). The value is incremented by the CC and reset at the start of a communication cycle.

Cleared when leaving CONFIG state or when entering STARTUP state.

21.2.7.5 FLXAnFRRCV — FlexRay Rate Correction Value Register

Access: FLXAnFRRCV is a read-only register that can be read in 32-bit units.
FLXAnFRRCVL is a read-only register that can be read in 16-bit units.
FLXAnFRRCVLL and FLXAnFRRCVLH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRRCV: <FLXAn_base> + 0118_H,
FLXAnFRRCVL: <FLXAn_base> + 0118_H,
FLXAnFRRCVLL: <FLXAn_base> + 0118_H, FLXAnFRRCVLH: <FLXAn_base> + 0119_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	RCV[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.52 FLXAnFRRCV Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned.
11 to 0	RCV[11:0]	Rate Correction Value Flag Indicates vRateCorrection

(1) FLXAnFRRCV.RCV

Rate Correction Value Flag

Indicates internal rate correction value (vRateCorrection/ two's complement) before limitation. If the FLXAnFRRCV.RCV value exceeds the limits defined by FLXAnFRGTUC10.MRC, flag FLXAnFRSFS.RCLR is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

CAUTION

The amount by which this value exceeded the limits is added to the external rate correction value.

21.2.7.6 FLXAnFROCV — FlexRay Offset Correction Value Register

Access: FLXAnFROCV is a read-only register that can be read in 32-bit units.
FLXAnFROCVL and FLXAnFROCVH are the read-only registers that can be read in 16-bit units.
FLXAnFROCVLL, FLXAnFROCVLH, and FLXAnFROCVHL are the read-only registers that can be read in 8-bit units.

Address: FLXAnFROCV: <FLXAn_base> + $011C_H$,
FLXAnFROCVL: <FLXAn_base> + $011C_H$, FLXAnFROCVH: <FLXAn_base> + $011E_H$,
FLXAnFROCVLL: <FLXAn_base> + $011C_H$, FLXAnFROCVLH: <FLXAn_base> + $011D_H$,
FLXAnFROCVHL: <FLXAn_base> + $011E_H$,

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	OCV[18:16]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OCV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.53 FLXAnFROCV Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned.
18 to 0	OCV[18:0]	Offset Correction Value Flag Indicates vOffsetCorrection

(1) FLXAnFROCV.OCV

Offset Correction Value Flag

Indicates the internal offset correction value (vOffsetCorrection/ two's complement) before limitation. If the FLXAnFROCV.OCV value exceeds the limits defined by FLXAnFRGTUC10.MOC, flag FLXAnFRSFS.OCLR is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

CAUTION

The amount by which this value exceeded the limits is added to the external offset correction value.

21.2.7.7 FLXAnFRSFS — FlexRay Sync Frame Status Register

Access: FLXAnFRSFS is a read-only register that can be read in 32-bit units.
FLXAnFRSFSL and FLXAnFRSFSH are the read-only registers that can be read in 16-bit units.
FLXAnFRSFSLL, FLXAnFRSFSLH, and FLXAnFRSFSHL are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRSFS: <FLXAn_base> + 0120_H
FLXAnFRSFSL: <FLXAn_base> + 0120_H, FLXAnFRSFSH: <FLXAn_base> + 0122_H
FLXAnFRSFSLL: <FLXAn_base> + 0120_H, FLXAnFRSFSLH: <FLXAn_base> + 0121_H,
FLXAnFRSFSHL: <FLXAn_base> + 0122_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	RCLR	MRCS	OCLR	MOCS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VSBO[3:0]				VSBE[3:0]				VSAO[3:0]				VSAE[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.54 FLXAnFRSFS Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned.
19	RCLR	Rate Correction Limit Reached Flag 0: Rate correction below limit 1: Rate correction limit reached
18	MRCS	Missing Rate Correction Signal Flag 0: Rate correction signal valid 1: Missing rate correction signal
17	OCLR	Offset Correction Limit Reached Flag 0: Offset correction below limit 1: Offset correction limit reached
16	MOCS	Missing Offset Correction Signal Flag 0: Offset correction signal valid 1: Missing offset correction signal
15 to 12	VSBO[3:0]	Valid Sync Frames Channel B, odd communication cycle
11 to 8	VSBE[3:0]	Valid Sync Frames Channel B, even communication cycle
7 to 4	VSAO[3:0]	Valid Sync Frames Channel A, odd communication cycle
3 to 0	VSAE[3:0]	Valid Sync Frames Channel A, even communication cycle

(1) FLXAnFRSFS.RCLR

Rate Correction Limit Reached Flag

The Rate Correction Limit Reached flag indicates that the rate correction value has exceeded its limit as defined by FLXAnFRGTUC10.MRC10 - MRC0. The flag is updated by the CC at start of offset correction phase.

Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(2) FLXAnFRSFS.MRCS

Missing Rate Correction Signal Flag

The Missing Rate Correction flag indicates that no rate correction calculation can be performed because no pairs of even / odd sync frames were received. The flag is updated by the CC at start of offset correction phase.

Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(3) FLXAnFRSFS.OCLR

Offset Correction Limit Reached Flag

The Offset Correction Limit Reached flag indicates that the offset correction value has exceeded its limit as defined by FLXAnFRGTUC10.MOC. The flag is updated by the CC at start of offset correction phase.

Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(4) FLXAnFRSFS.MOCS

Missing Offset Correction Signal Flag

The Missing Offset Correction flag indicates that no offset correction calculation can be performed because no sync frames were received. The flag is updated by the CC at start of offset correction phase.

Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(5) FLXAnFRSFS.VSBO

Valid Sync Frames Channel B, odd communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHB is 1.

Indicates the number of valid sync frames received on channel B in the odd communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(6) FLXAnFRSFS.VSBE

Valid Sync Frames Channel B, even communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHB is 1.

Indicates the number of valid sync frames received on channel B in the even communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each even communication cycle.

Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(7) FLXAnFRSFS.VSAO

Valid Sync Frames Channel A, odd communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHA is 1.

Indicates the number of valid sync frames received on channel A in the odd communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each odd communication cycle.

Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

(8) FLXAnFRSFS.VSAE

Valid Sync Frames Channel A, even communication cycle

These bits are only valid when FLXAnFRSUCC1.CCHA is 1.

Indicates the number of valid sync frames received on channel A in the even communication cycle. If transmission of sync frames is enabled by FLXAnFRSUCC1.TXSY the value is incremented by one. The value is updated during the NIT of each even communication cycle.

Cleared when leaving CONFIG state or when entering STARTUP state, INTEGRATION_COLDSTART_CHECK state or INTEGRATION_CONSISTENCY_CHECK state.

21.2.7.8 FLXAnFRSWNIT — FlexRay Symbol Window and NIT Status Register

Indicates the symbol windows related status. Symbol window related status information is updated by the CC at the end of the symbol window for each channel. NIT related status information is updated by the CC at the end of the NIT for each channel.

During startup the status data is not updated.

Access: FLXAnFRSWNIT is a read-only register that can be read in 32-bit units.
FLXAnFRSWNITL is a read-only register that can be read in 16-bit units.
FLXAnFRSWNITLL and FLXAnFRSWNITLH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRSWNIT: <FLXAn_base> + 0124_H,
FLXAnFRSWNITL: <FLXAn_base> + 0124_H,
FLXAnFRSWNITLL: <FLXAn_base> + 0124_H, FLXAnFRSWNITLH: <FLXAn_base> + 0125_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SBNB	SENB	SBNA	SENA	MTSB	MTSA	TCSB	SBSB	SESB	TCSA	SBSA	SESA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.55 FLXAnFRSWNIT Register Contents (1/2)

Bit	Symbol	Function
31 to 12	Reserved	When read, the value after reset is returned.
11	SBNB	Slot Boundary Violation during NIT Channel B Flag 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel B
10	SENB	Syntax Error during NIT Channel B Flag 0: No syntax error detected 1: Syntax error during NIT detected on channel B
9	SBNA	Slot Boundary Violation during NIT Channel A Flag 0: No slot boundary violation detected 1: Slot boundary violation during NIT detected on channel A
8	SENA	Syntax Error during NIT Channel A Flag 0: No syntax error detected 1: Syntax error during NIT detected on channel A
7	MTSB	MTS Received on Channel B Flag 0: No MTS symbol received on channel B 1: MTS symbol received on channel B
6	MTSA	MTS Received on Channel A Flag 0: No MTS symbol received on channel A 1: MTS symbol received on channel A
5	TCSB	Transmission Conflict in Symbol Window Channel B Flag 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel B
4	SBSB	Slot Boundary Violation in Symbol Window Channel B Flag 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel B

Table 21.55 FLXAnFRSWNIT Register Contents (2/2)

Bit	Symbol	Function
3	SESB	Syntax Error in Symbol Window Channel B Flag 0: No syntax error detected 1: Syntax error during symbol window detected on channel B
2	TCSA	Transmission Conflict in Symbol Window Channel A Flag 0: No transmission conflict detected 1: Transmission conflict in symbol window detected on channel A
1	SBSA	Slot Boundary Violation in Symbol Window Channel A Flag 0: No slot boundary violation detected 1: Slot boundary violation during symbol window detected on channel A
0	SESA	Syntax Error in Symbol Window Channel A Flag 0: No syntax error detected 1: Syntax error during symbol window detected on channel A

(1) FLXAnFRSWNIT.SBNB

Indicates a Slot Boundary Violation during NIT Channel B Flag (vSS!BViolationB).

Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRSWNIT.SENB

Indicates a Syntax Error during NIT Channel B Flag (vSS!SyntaxErrorB).

Cleared when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRSWNIT.SBNA

Indicates a Slot Boundary Violation during NIT Channel A Flag (vSS!BViolationA).

Cleared when leaving CONFIG state or when entering STARTUP state.

(4) FLXAnFRSWNIT.SENA

Indicates a Syntax Error during NIT Channel A Flag (vSS!SyntaxErrorA).

Cleared when leaving CONFIG state or when entering STARTUP state.

(5) FLXAnFRSWNIT.MTSB

Indicates a MTS Received on Channel B Flag (vSS!ValidMTSB).

Media Access Test Symbol received on channel B during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.

When this bit is set to 1, also interrupt flag FLXAnFRSIR.MTSB is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

(6) FLXAnFRSWNIT.MTSA

Indicates a MTS Received on Channel A Flag (vSS!ValidMTSA).

Media Access Test Symbol received on channel A during the preceding symbol window. Updated by the CC for each channel at the end of the symbol window.

When this bit is set to 1, also interrupt flag FLXAnFRSIR.MTSA is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

(7) FLXAnFRSWNIT.TCSB

Indicates a Transmission Conflict in Symbol Window Channel B Flag (vSS!TxConflictB).

Cleared when leaving CONFIG state or when entering STARTUP state.

(8) FLXAnFRSWNIT.SBSB

Indicates a Slot Boundary Violation in Symbol Window Channel B Flag (vSS!BViolationB).

Cleared when leaving CONFIG state or when entering STARTUP state.

(9) FLXAnFRSWNIT.SESB

Indicates a Syntax Error in Symbol Window Channel B Flag (vSS!SyntaxErrorB).

Cleared when leaving CONFIG state or when entering STARTUP state.

(10) FLXAnFRSWNIT.TCSA

Indicates a Transmission Conflict in Symbol Window Channel A Flag (vSS!TxConflictA).

Cleared when leaving CONFIG state or when entering STARTUP state.

(11) FLXAnFRSWNIT.SBSA

Indicates a Slot Boundary Violation in Symbol Window Channel A Flag (vSS!BViolationA).

Cleared when leaving CONFIG state or when entering STARTUP state.

(12) FLXAnFRSWNIT.SESA

Indicates a Syntax Error in Symbol Window Channel A Flag (vSS!SyntaxErrorA).

Cleared when leaving CONFIG state or when entering STARTUP state.

21.2.7.9 FLXAnFRACS — FlexRay Aggregated Channel Status Register

The aggregated channel status provides the Host with an accrued status of channel activity for all communication slots regardless of whether they are assigned for transmission or subscribed for reception.

The aggregated channel status also includes status data from the symbol window and the network idle time.

The status data is updated (set) after each slot and aggregated until it is cleared by the Host.

During startup the status data is not updated.

Access: FLXAnFRACS can be read or written in 32-bit units.
FLXAnFRACSL can be read or written in 16-bit units.
FLXAnFRACSLL and FLXAnFRACSLH can be read or written in 8-bit units.

Address: FLXAnFRACS: <FLXAn_base> + 0128_H,
FLXAnFRACSL: <FLXAn_base> + 0128_H,
FLXAnFRACSLL: <FLXAn_base> + 0128_H, FLXAnFRACSLH: <FLXAn_base> + 0129_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SBVB	CIB	CEDB	SEDB	VFRB	—	—	—	SBVA	CIA	CEDA	SEDA	VFRA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 21.56 FLXAnFRACS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	SBVB	Slot Boundary Violation on Channel B Flag 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel B
11	CIB	Communication Indicator Channel B Flag 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel B in slots containing any additional communication
10	CEDB	Content Error Detected on Channel B Flag 0: No frame with content error received 1: Frame(s) with content error received on channel B
9	SEDB	Syntax Error Detected on Channel B Flag 0: No syntax error observed 1: Syntax error(s) observed on channel B
8	VFRB	Valid Frame Received on Channel B Flag 0: No valid frame received 1: Valid frame(s) received on channel B
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	SBVA	Slot Boundary Violation on Channel A Flag 0: No slot boundary violation observed 1: Slot boundary violation(s) observed on channel A

Table 21.56 FLXAnFRACS Register Contents (2/2)

Bit Position	Bit Name	Function
3	CIA	Communication Indicator Channel A Flag 0: No valid frame(s) received in slots containing any additional communication 1: Valid frame(s) received on channel A in slots containing any additional communication
2	CEDA	Content Error Detected on Channel A Flag 0: No frame with content error received 1: Frame(s) with content error received on channel A
1	SEDA	Syntax Error Detected on Channel A Flag 0: No syntax error observed 1: Syntax error(s) observed on channel A
0	VFRA	Valid Frame Received on Channel A Flag 0: No valid frame received 1: Valid frame(s) received on channel A

(1) FLXAnFRACS.SBVB

Slot Boundary Violation on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

One or more slot boundary violations were observed on channel B at any time during the observation period (static or dynamic slots, symbol window, and NIT).

When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDB is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRACS.CIB

Communication Indicator Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

Indicates that one or more valid frames were received on channel B in slots that also contained additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation.

CAUTION

The set condition of the flag FLXAnFRACS.CIB is also fulfilled if there is only one single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase.
When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDB is set to 1.
Cleared when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRACS.CEDB

Content Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

Indicates that one or more frames with a content error were received on channel B in any static or dynamic slot during the observation period.

When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDB is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

(4) FLXAnFRACS.SEDB

Syntax Error Detected on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel B.

When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDB is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

(5) FLXAnFRACS.VFRB

Valid Frame Received on Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

One or more valid frames were received on channel B in any static or dynamic slot during the observation period.

Cleared when leaving CONFIG state or when entering STARTUP state.

(6) FLXAnFRACS.SBVA

Slot Boundary Violation on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

Slot boundary violations were observed on channel A at any time during the observation period (static or dynamic slots, symbol window, and NIT).

When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDA is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

(7) FLXAnFRACS.CIA

Communication Indicator Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

One or more valid frames were received on channel A in slots that also contained any additional communication during the observation period, i.e. one or more slots received a valid frame AND had any combination of either syntax error OR content error OR slot boundary violation.

CAUTION

The set condition of the flag FLXAnFRACS.CIA is also fulfilled if there is only one single frame in the slot and the slot boundary at the end of the slot is reached during the frames channel idle recognition phase.

When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDA is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

(8) FLXAnFRACS.CEDA

Content Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

One or more frames with a content error were received on channel A in any static or dynamic slot during the observation period.

When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDA is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

(9) FLXAnFRACS.SEDA

Syntax Error Detected on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

One or more syntax errors in static or dynamic slots, symbol window, and NIT were observed on channel A.

When this flag changes from 0 to 1, interrupt flag FLXAnFREIR.EDA is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

(10) FLXAnFRACS.VFRA

Valid Frame Received on Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

One or more valid frames were received on channel A in any static or dynamic slot during the observation period.

Cleared when leaving CONFIG state or when entering STARTUP state.

21.2.7.10 FLXAnFRESIDm — FlexRay Even Sync ID Register m (m = 1 to 15)

Registers FLXAnFRESID1 to FLXAnFRESID15 hold the frame IDs of the sync frames received in even communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXAnFRESID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an even communication cycle, register FLXAnFRESID1 holds the respective sync frame ID as configured in message buffer 0 and flags FLXAnFRESID1.RXEA, FLXAnFRESID1.RXEB are set. The value is updated during the NIT of each even communication cycle.

Access: FLXAnFRESIDm is a read-only register that can be read in 32-bit units.
FLXAnFRESIDmL is a read-only register that can be read in 16-bit units.
FLXAnFRESIDmLL and FLXAnFRESIDmLH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRESIDm: <FLXAn_base> + 0130_H + (m - 1) × 4_H,
FLXAnFRESIDmL: <FLXAn_base> + 0130_H + (m - 1) × 4_H,
FLXAnFRESIDmLL: <FLXAn_base> + 0130_H + (m - 1) × 4_H,
FLXAnFRESIDmLH: <FLXAn_base> + 0130_H + (m - 1) × 4_H + 1_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXEB	RXEA	—	—	—	—	EID[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.57 FLXAnFRESIDn (n=1 to 15) Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	RXEB	Received / Configured Even Sync ID on Channel B Flag 0: No sync frame received on channel B / node not configured to transmit sync frames 1: Sync frame received on channel B / node configured to transmit sync frames
14	RXEA	Received / Configured Even Sync ID on Channel A Flag 0: No sync frame received on channel A / node not configured to transmit sync frames 1: Sync frame received on channel A / node configured to transmit sync frames
13 to 10	Reserved	When read, the value after reset is returned.
9 to 0	EID[9:0]	Even Sync ID (vsSyncIDListA,B even Flag)

(1) FLXAnFRESIDn.RXEB

Received / Configured Even Sync ID on Channel B Flag

Signals that a sync frame corresponding to the stored even sync ID was received on channel B or that the node is configured to be a sync node with key slot = FLXAnFRESID1.EID.

Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFRESIDn.RXEA

Received / Configured Even Sync ID on Channel A Flag

Signals that a sync frame corresponding to the stored even sync ID was received on channel A or that the node is configured to be a sync node with key slot = FLXAnFRESID1.EID.

Cleared when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFRESIDn.EID

Even Sync ID Flag (vsSyncIDListA,B even)

Indicates the sync frame ID received in an even communication cycle.

Cleared when leaving CONFIG state or when entering STARTUP state.

21.2.7.11 FLXAnFROSIDm — FlexRay Odd Sync ID Register m (m = 1 to 15)

Registers FLXAnFROSID1 to FLXAnFROSID15 hold the frame IDs of the sync frames received in odd communication cycles used for clock synchronization up to the limit of gSyncNodeMax. The values are sorted in ascending order, with register FLXAnFROSID1 holding the lowest received sync frame ID. If the node itself transmits a sync frame in an odd communication cycle, register FLXAnFROSID1 holds the respective sync frame ID as configured in message buffer 0 and flags FLXAnFROSID1.RXOA, FLXAnFROSID1.RXOB are set. The value is updated during the NIT of each odd communication cycle.

Access: FLXAnFROSIDm is a read-only register that can be read in 32-bit units.
FLXAnFROSIDmL is a read-only register that can be read in 16-bit units.
FLXAnFROSIDmLL and FLXAnFROSIDmLH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFROSIDm: <FLXAn_base> + 0170_H + (m - 1) × 4_H,
FLXAnFROSIDmL: <FLXAn_base> + 0170_H + (m - 1) × 4_H,
FLXAnFROSIDmLL: <FLXAn_base> + 0170_H + (m - 1) × 4_H,
FLXAnFROSIDmLH: <FLXAn_base> + 0170_H + (m - 1) × 4_H + 1_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RXOB	RXOA	—	—	—	—	OID[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.58 FLXAnFROSIDn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	RXOB	Received / Configured Odd Sync ID on Channel B Flag 0: No sync frame received on channel B / node not configured to transmit sync frames 1: Sync frame received on channel B / node configured to transmit sync frames
14	RXOA	Received / Configured Odd Sync ID on Channel A Flag 0: No sync frame received on channel A / node not configured to transmit sync frames 1: Sync frame received on channel A / node configured to transmit sync frames
13 to 10	Reserved	When read, the value after reset is returned.
9 to 0	OID[9:0]	Odd Sync ID Flag (vsSyncIDListA,B odd)

(1) FLXAnFROSIDn.RXOB

Received / Configured Odd Sync ID on Channel B Flag

Signals that a sync frame corresponding to the stored odd sync ID was received on channel B or that the node is configured to be a sync node with key slot = FLXAnFROSID1.OID.

Cleared when leaving CONFIG state or when entering STARTUP state.

(2) FLXAnFROSIDn.RXOA

Received / Configured Odd Sync ID on Channel A Flag

Signals that a sync frame corresponding to the stored odd sync ID was received on channel A or that the node is configured to be a sync node with key slot = FLXAnFROSID1.OID.

Cleared when leaving CONFIG state or when entering STARTUP state.

(3) FLXAnFROSIDn.OID

Odd Sync ID Flag (vsSyncIDListA,B odd)

Indicates the sync frame ID odd communication cycle.

Cleared when leaving CONFIG state or when entering STARTUP state.

21.2.7.12 FLXAnFRNMVm — FlexRay Network Management Vector Register m (m = 1 to 3)

The three network management registers hold the accrued NM vector (see **Section 21.3.7, Network Management**).

The CC updates the NM vector at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state.

NMVn-bytes exceeding the configured NM vector length are not valid.

For information about the byte alignment of the received NM vector in this register see **Section 21.3.17, Byte Alignment**.

Access: FLXAnFRNMVm is a read-only register that can be read in 32-bit units.
FLXAnFRNMVmL and FLXAnFRNMVmH are the read-only registers that can be read in 16-bit units.
FLXAnFRNMVmLL, FLXAnFRNMVmLH, FLXAnFRNMVmHL, and FLXAnFRNMVmHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRNMVm: <FLXAn_base> + $01B0_H$ + (m - 1) × 4_H,
FLXAnFRNMVmL: <FLXAn_base> + $01B0_H$ + (m - 1) × 4_H,
FLXAnFRNMVmH: <FLXAn_base> + $01B0_H$ + (m - 1) × 4_H + 2_H,
FLXAnFRNMVmLL: <FLXAn_base> + $01B0_H$ + (m - 1) × 4_H,
FLXAnFRNMVmLH: <FLXAn_base> + $01B0_H$ + (m - 1) × 4_H + 1_H,
FLXAnFRNMVmHL: <FLXAn_base> + $01B0_H$ + (m - 1) × 4_H + 2_H,
FLXAnFRNMVmHH: <FLXAn_base> + $01B0_H$ + (m - 1) × 4_H + 3_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NM[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.59 FLXAnFRNMVm Register Contents

Bit Position	Bit Name	Function
31 to 0	NM[31:0]	NM Vector The three network management vector registers hold the accrued NM vector (configurable 0 to 12 bytes). The NM vector to be held is generated by bit-wise logic OR for each NM vector received on each channel (valid static frames with PPI = 1) (see **Section 21.3.7, Network Management**). For information about the byte alignment of the received NM vector in this register see **Section 21.3.17, Byte Alignment**. NMVn-bytes exceeding the configured NM vector length are not valid. The Register Contents are updated at the end of each communication cycle as long as the CC is either in NORMAL_ACTIVE or NORMAL_PASSIVE state. These bits are cleared when leaving CONFIG state or when entering STARTUP state.

21.2.8 Message Buffer Control Registers

21.2.8.1 FLXAnFRMRC — FlexRay Message RAM Configuration Register

The Message RAM Configuration register defines the number of message buffers assigned to the static segment, dynamic segment, and FIFO.

The Message RAM can be divided into up three different areas; Static Buffer area, Static and Dynamic Buffer area, FIFO area. If present, the Static Buffer area is starting at Message Buffer 0.

The start of the Static and Dynamic Buffer area is configured by FLXAnFRMRC.FDB. FLXAnFRMRC.FDB defines the end of the Static Buffer area. If no Static Buffer area is present, the Static and Dynamic Buffer area starts at Message Buffer 0.

The start of the FIFO area is configured by FLXAnFRMRC.FFB. FLXAnFRMRC.FFB defines the end of the previous area, which can be either the Static Buffer area or the Static and Dynamic Buffer area. If no Static Buffer area and no Static and Dynamic Buffer area is present, the FIFO area starts at Message Buffer 0.

The end of the last configured area, which can be the Static Buffer area, the Static and Dynamic buffer area, or the FIFO area, is configured by FLXAnFRMRC.LCB.

Figure 21.2 shows an example configuration of the Message RAM where all there area are configured.

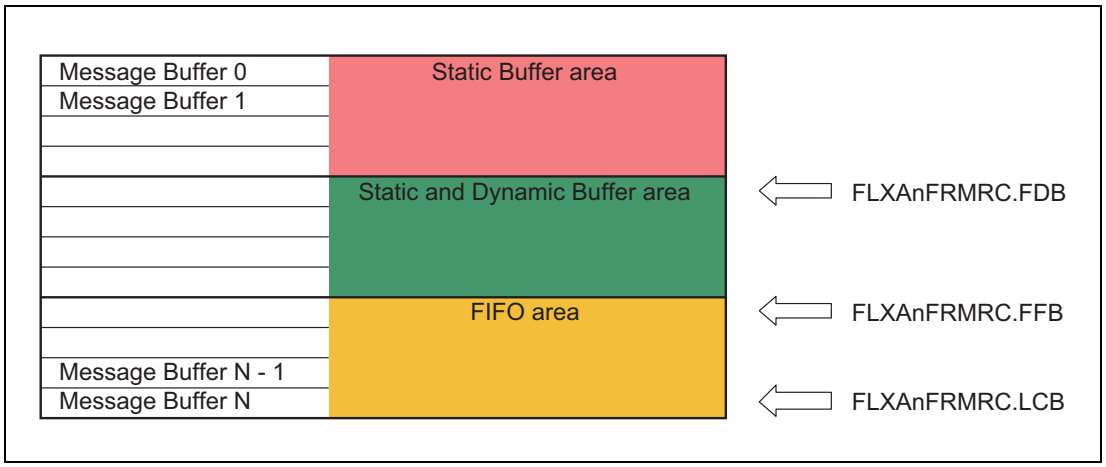


Figure 21.2 Message RAM Organization

CAUTIONS

1. **If the node is configured as sync node (FLXAnFRSUCC1.TXSY = 1) or for single slot mode operation (FLXAnFRSUCC1.TSM = 1), message buffer 0 or 1 is reserved for sync frames or single slot frames and have to be configured with the node-specific key slot ID. In case the node is neither configured as sync node nor for single slot operation message buffer 0 or 1 is treated like all other message buffers.**
2. **The maximum number of header sections is 128. This means a maximum of 128 message buffers can be configured. The maximum length of a data section is 254 bytes. The length of the data section may be configured differently for each message buffer. For details see Section 21.3.13, Message RAM.**
3. **If two or more message buffers are assigned to slot 1 by use of cycle filtering, all of them must be located either in the “Static Buffers” or at the beginning of the**

"Static + Dynamic Buffers" section.

4. **The FlexRay protocol specification requires that each node has to send a frame in its key slot. Therefore at least message buffer 0 is reserved for transmission in the key slot. Due to this requirement a maximum number of 127 message buffers can be assigned to the FIFO. Nevertheless, a non protocol conform configuration without a transmission slot in the static segment would still be operational.**
5. **The payload length configured and the length of the data section need to be configured identical for all message buffers belonging to the FIFO via FLXAnFRWRHS2.PL and FLXAnFRWRHS3.DP. When the CC is not in DEFAULT_CONFIG or CONFIG state reconfiguration of message buffers belonging to the FIFO is locked.**

Access: FLXAnFRMRC can be read or written in 32-bit units.
FLXAnFRMRCL and FLXAnFRMRCH can be read or written in 16-bit units.
FLXAnFRMRCLL, FLXAnFRMRCLH, FLXAnFRMRCHL and FLXAnFRMRCHH can be read or written in 8-bit units.

Address: FLXAnFRMRC: <FLXAn_base> + 0300_H,
FLXAnFRMRCL: <FLXAn_base> + 0300_H, FLXAnFRMRCH: <FLXAn_base> + 0302_H,
FLXAnFRMRCLL: <FLXAn_base> + 0300_H, FLXAnFRMRCLH: <FLXAn_base> + 0301_H,
FLXAnFRMRCHL: <FLXAn_base> + 0302_H, FLXAnFRMRCHH: <FLXAn_base> + 0303_H

Value after reset: $0180\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SPLM	SEC[1:0]		LCB[7:0]							
Value after reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FFB[7:0]								FDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.60 FLXAnFRMRC Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
26	SPLM	Sync Frame Payload Multiplex Bit 0: Only message buffer 0 locked against reconfiguration 1: Both message buffers 0 and 1 are locked against reconfiguration
25, 24	SEC[1:0]	Secure Buffers Bit 00_B: all buffers unlocked 01_B: static buffers locked, FIFO locked, limited transmission 10_B: all buffers locked 11_B: all buffers locked, limited transmission
23 to 16	LCB[7:0]	Last Configured Buffer Bit 0 to 127: Number of message buffers is FLXAnFRMRF.LCB + 1 128: No message buffer assigned to the FIFO
15 to 8	FFB[7:0]	First Buffer of FIFO Bit 0: All message buffers assigned to the FIFO 1 to 127: Message buffers from FLXAnFRMRC.FFB to FLXAnFRMRC.LCB assigned to the FIFO 128: No message buffer configured

Table 21.60 FLXAnFRMRC Register Contents

Bit Position	Bit Name	Function
7 to 0	FDB[7:0]	First Dynamic Buffer Bit 0: No group of message buffers exclusively for the static segment configured 1 to 127: Message buffers 0 to FLXAnFRMRC.FDB - 1 reserved for static segment 128: No dynamic message buffers configured

(1) FLXAnFRMRC.SPLM

Sync Frame Payload Multiplex Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

This bit is only evaluated if the node is configured as sync node (FLXAnFRSUCC1.TXSY = 1) or for single slot mode operation (FLXAnFRSUCC1.TSM = 1).

When this bit is set to 1 message buffers 0 and 1 are dedicated for sync frame transmission with different payload data on channel A and B.

When this bit is set to 0, sync frames are transmitted from message buffer 0 with the same payload data on all channels configured. Note that the channel filter configuration for message buffer 0 or message buffer 1 has to be chosen accordingly.

(2) FLXAnFRMRC.SEC

Secure Buffers Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Not evaluated when the CC is in DEFAULT_CONFIG or CONFIG state.

For temporary unlocking, see **Section 21.3.13.4, Host Handling of Access Errors**.

00_B = all buffers unlocked

Reconfiguration of message buffers enabled with numbers < FLXAnFRMRC.FFB enabled

Exception: In nodes configured for sync frame transmission or for single slot mode operation message buffer 0 (and if FLXAnFRMRC.SPLM = 1, also message buffer 1) is always locked

01_B = static buffers locked, FIFO locked, limited transmission

Reconfiguration of message buffers with numbers < FLXAnFRMRC.FDB and with numbers ≥ FLXAnFRMRC.FFB locked and transmission of message buffers for static segment with numbers ≥ FLXAnFRMRC.FDB disabled

10_B = all buffers locked

Reconfiguration of all message buffers locked

11_B = all buffers locked, limited transmission

Reconfiguration of all message buffers locked and transmission of message buffers for static segment with numbers ≥ FLXAnFRMRC.FDB disabled

(3) FLXAnFRMRC.LCB

Last Configured Buffer Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

When a Static and Dynamic Buffer area is configured (FLXAnFRMRC.FDB < 128), the user should configure FLXAnFRMRC.LCB ≥ FLXAnFRMRC.FDB.

When a FIFO area is configured (FLXAnFRMRC.FFB < 128), the user should configure FLXAnFRMRC.LCB ≥ FLXAnFRMRC.FFB.

(4) FLXAnFRMRC.FFB

First Buffer of FIFO Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

When a Static and Dynamic Buffer area is configured (FLXAnFRMRC.FDB < 128), the user should configure FLXAnFRMRC.FFB > FLXAnFRMRC.FDB.

(5) FLXAnFRMRC.FDB

First Dynamic Buffer Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

21.2.8.2 FLXAnFRFRF — FlexRay FIFO Rejection Filter Register

The FIFO Rejection Filter defines a user specified sequence of bits to which channel, frame ID, and cycle count of the incoming frames are compared. Together with the FIFO Rejection Filter Mask this register determines whether a message is rejected by the FIFO.

Access: FLXAnFRFRF can be read or written in 32-bit units.
FLXAnFRFRFL and FLXAnFRFRFH can be read or written in 16-bit units.
FLXAnFRFRFLL, FLXAnFRFRFLH, FLXAnFRFRFLH, FLXAnFRFRFHL, and FLXAnFRFRFHH can be read or written in 8-bit units.

Address: FLXAnFRFRF: <FLXAn_base> + 0304_H,
FLXAnFRFRFL: <FLXAn_base> + 0304_H, FLXAnFRFRFH: <FLXAn_base> + 0306_H,
FLXAnFRFRFLL: <FLXAn_base> + 0304_H, FLXAnFRFRFLH: <FLXAn_base> + 0305_H,
FLXAnFRFRFHL: <FLXAn_base> + 0306_H, FLXAnFRFRFHH: <FLXAn_base> + 0307_H

Value after reset: $0180\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	RNF	RSS	CYF[6:0]						
Value after reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FID[10:0]											CH[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.61 FLXAnFRFRF Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
24	RNF	Reject Null Frames Bit 0: Null frames are stored in the FIFO 1: Reject all null frames
23	RSS	Reject in Static Segment Bit 0: FIFO also used for static segment 1: Reject messages in static segment
22 to 16	CYF[6:0]	Cycle Counter Filter Bit Cycle Counter Filter
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 2	FID[10:0]	Frame ID Filter Bit 0 to 2047: Frame ID filter values
1, 0	CH[1:0]	Channel Filter Bit 00_B: receive on both channels 01_B: receive only on channel B 10_B: receive only on channel A 11_B: no reception

(1) FLXAnFRFRF.RNF

Reject Null Frames Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

If this bit is set to 1, received null frames are not stored in the FIFO.

(2) FLXAnFRFRF.RSS

Reject in Static Segment Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

If this bit is set to 1, the FIFO is used only for the dynamic segment.

(3) FLXAnFRFRF.CYF

Cycle Counter Filter Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FLXAnFRFRF.CYF, all frames are rejected. For details about the configuration of the cycle counter filter see **Section 21.3.8.2, Cycle Counter Filtering**.

(4) FLXAnFRFRF.FID

Frame ID Filter Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

Determines the frame ID to be rejected by the FIFO. With the additional configuration of register FLXAnFRFRFM, the corresponding frame ID filter bits are ignored, which results in further rejected frame IDs. When FLXAnFRFRFM.MFID is zero, a frame ID filter value of zero means that no frame ID is rejected.

(5) FLXAnFRFRF.CH

Channel Filter Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

If reception on both channels is configured, also in static segment always both frames (from channel A and B) are stored in the FIFO, even if they are identical.

21.2.8.3 FLXAnFRFRFM — FlexRay FIFO Rejection Filter Mask Register

The FlexRay FIFO Rejection Filter Mask specifies which of the corresponding frame ID filter bits are relevant for rejection filtering. If a bit is set to 1, it indicates that the corresponding bit in the FLXAnFRFRF register will not be considered for rejection filtering.

Access: FLXAnFRFRFM can be read or written in 32-bit units.
FLXAnFRFRFML can be read or written in 16-bit units.
FLXAnFRFRFMLL and FLXAnFRFRFMLH can be read or written in 8-bit units.

Address: FLXAnFRFRFM: <FLXAn_base> + 0308_H,
FLXAnFRFRFML: <FLXAn_base> + 0308_H,
FLXAnFRFRFMLL: <FLXAn_base> + 0308_H, FLXAnFRFRFMLH: <FLXAn_base> + 0309_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	MFID[10:0]											—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 21.62 FLXAnFRFRFM Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 2	MFID[10:0]	Mask Frame ID Filter Bit 0: Corresponding frame ID filter bit is used for rejection filtering 1: Ignore corresponding frame ID filter bit.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(1) FLXAnFRFRFM.MFID

Mask Frame ID Filter Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

21.2.8.4 FLXAnFRFCL — FlexRay FIFO Critical Level Register

Access: FLXAnFRFCL can be read or written in 32-bit units.
FLXAnFRFCLL can be read or written in 16-bit units.
FLXAnFRFCLLL can be read or written in 8-bit units.

Address: FLXAnFRFCL: <FLXAn_base> + $030C_H$,
FLXAnFRFCLL: <FLXAn_base> + $030C_H$,
FLXAnFRFCLLL: <FLXAn_base> + $030C_H$

Value after reset: $0000\ 0080_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CL[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.63 FLXAnFRFCL Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	CL[7:0]	Critical Level Bit Critical Level

(1) FLXAnFRFCL.CL

Critical Level Bit

The user can only write to these bits when FLXAnFRCCSV.POCS is DEFAULT_CONFIG or CONFIG.

When the receive FIFO fill level (FLXAnFRFSR.RFFL) is equal or greater than the critical level configured by FLXAnFRFCL.CL, the receive FIFO critical level flag FLXAnFRFSR.RFCL is set to 1.

If FLXAnFRFCL.CL is programmed to values > 128, bit FLXAnFRFSR.RFCL is never set to 1.

21.2.9 Message Buffer Status Registers

21.2.9.1 FLXAnFRMHDS — FlexRay Message Handler Status Register

Access: FLXAnFRMHDS can be read or written in 32-bit units.
FLXAnFRMHDSL can be read or written in 16-bit units.
FLXAnFRMHDSH is a read-only register that can be read in 16-bit units.
FLXAnFRMHDSLL can be read or written in 8-bit units.
FLXAnFRMHDSLH, FLXAnFRMHDSHL, and FLXAnFRMHDSHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRMHDS: <FLXAn_base> + 0310_H,
FLXAnFRMHDSL: <FLXAn_base> + 0310_H, FLXAnFRMHDSH: <FLXAn_base> + 0312_H,
FLXAnFRMHDSLL: <FLXAn_base> + 0310_H, FLXAnFRMHDSLH: <FLXAn_base> + 0311_H,
FLXAnFRMHDSHL: <FLXAn_base> + 0312_H, FLXAnFRMHDSHH: <FLXAn_base> + 0313_H

Value after reset: $0000\ 0080_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	MBU[6:0]							—	MBT[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	FMB[6:0]							CRAM	MFMB	FMBD	ATBF2	ATBF1	AMR	—	—
Value after reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R

Table 21.64 FLXAnFRMHDS Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 24	MBU[6:0]	Message Buffer Updated Flag
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	MBT[6:0]	Message Buffer Transmitted Flag
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	FMB[6:0]	Faulty Message Buffer Number Flag
7	CRAM	Clear all internal RAM's Flag 0: No execution of the CHI command CLEAR_RAMS 1: Execution of the CHI command CLEAR_RAMS ongoing
6	MFMB	Multiple Faulty Message Buffer Detection Flag 0: No additional faulty message buffer. 1: Additional faulty message buffer was detected while the FLXAnFRMHDS.FMBD flag is set to 1.
5	FMBD	Faulty Message Buffer Detection Flag 0: No faulty message buffer. 1: Message buffer referenced by FLXAnFRMHDS.FMBD holds faulty data with a parity error.
4	ATBF2	Nonresident Buffer RAM B Access Error Flag 0: No access error 1: Access error occurred when reading the RAM B.
3	ATBF1	Nonresident Buffer RAM A Access Error Flag 0: No access error. 1: Access error occurred when reading the RAM A.

Table 21.64 FLXAnFRMHDS Register Contents

Bit Position	Bit Name	Function
2	AMR	Message RAM Access Error Flag 0 = No access error 1 = Access error occurred when reading the Message RAM.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(1) FLXAnFRMHDS.MBU

Message Buffer Updated Flag

Number of message buffer that was updated last by the CC. For this message buffer the respective ND and / or MBC flag in the FLXAnFRNDAT1/2/3/4 registers and the FLXAnFRMBSC1/2/3/4 registers are also set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(2) FLXAnFRMHDS.MBT

Message Buffer Transmitted Flag

Number of last successfully transmitted message buffer.

If the message buffer is configured for single-shot mode, the respective TXR flag in the FLXAnFRTXRQ1/2/3/4 registers was reset to 0.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(3) FLXAnFRMHDS.FMB

Faulty Message Buffer Number Flag

This flag indicates that an access error occurred when reading from the message buffer referenced by FLXAnFRMHDS.FMB.

The value of this flag is only valid when FLXAnFRMHDS.AMR and flag FLXAnFRMHDS.FMBD are set to 1.

This flag is not updated while the FLXAnFRMHDS.FMBD flag is 1.

This flag is cleared by the CHI command CLEAR_RAMS.

(4) FLXAnFRMHDS.CRAM

Internal RAM Clear Flag

This flag indicates that the CHI command CLEAR_RAMS is ongoing (all bits of the message RAM, input buffer, output buffer and nonresident buffer are written to 0).

This flag is set by the CHI command CLEAR_RAMS.

(5) FLXAnFRMHDS.MFMB

Multiple Faulty Message Buffer Detection Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This bit indicates that an additional faulty message buffer was detected while the FMBD flag is set.

This bit is cleared by the CHI command CLEAR_RAMS.

(6) FLXAnFRMHDS.FMBD

Faulty Message Buffer Detection Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This bit indicates that the message buffer holds faulty data due to an access error.

This bit is cleared by the CHI command CLEAR_RAMS.

(7) FLXAnFRMHD.ATBF2

Nonresident Buffer RAM B Access Error Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This flag indicates that an access error occurred when reading the RAM B.

CAUTION

When this flag changes from 0 to 1, the AERR bit in the FLXAnFREIR register is set to 1. This flag can be reset by the CHI command CLEAR_RAMS.

(8) FLXAnFRMHD.ATBF1

Nonresident Buffer RAM A Access Error Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This flag indicates that an access error occurred when reading the RAM A.

CAUTION

When this flag changes from 0 to 1, the AERR bit in the FLXAnFREIR register is set to 1. This flag can be reset by the CHI command CLEAR_RAMS.

(9) FLXAnFRMHDS.AMR

Message RAM Access Error Flag

Writing 0 has no effect on the bit value.

This bit is cleared when writing 1 to it.

This flag indicates that an access error occurred when reading the Message RAM.

CAUTION

When this flag changes from 0 to 1, the AERR bit in the FLXAnFREIR register is set to 1. This flag can be reset by the CHI command CLEAR_RAMS.

21.2.9.2 FLXAnFRLDTS — FlexRay Last Dynamic Transmit Slot Register

Access: FLXAnFRLDTS is a read-only register that can be read in 32-bit units.
FLXAnFRLDTSL and FLXAnFRLDTSH are the read-only registers that can be read in 16-bit units.
FLXAnFRLDTSLL, FLXAnFRLDTSLH, FLXAnFRLDTSHL, and FLXAnFRLDTSHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRLDTS: <FLXAn_base> + 0314_H,
FLXAnFRLDTSL: <FLXAn_base> + 0314_H, FLXAnFRLDTSH: <FLXAn_base> + 0316_H,
FLXAnFRLDTSLL: <FLXAn_base> + 0314_H, FLXAnFRLDTSLH: <FLXAn_base> + 0315_H,
FLXAnFRLDTSHL: <FLXAn_base> + 0316_H, FLXAnFRLDTSHH: <FLXAn_base> + 0317_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	LDTB[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	LDTA[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.65 FLXAnFRLDTS Register Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned.
26 to 16	LDTB[10:0]	Last Dynamic Transmission Channel B Flag
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	LDTA[10:0]	Last Dynamic Transmission Channel A Flag

(1) FLXAnFRLDTS.LDTB

Last Dynamic Transmission Channel B Flag

Stores the value of vSlotCounter[B] at the time of the last frame transmission on channel B in the dynamic segment of this node.

It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS of channel B.

(2) FLXAnFRLDTS.LDTA

Last Dynamic Transmission Channel A Flag

Stores the value of vSlotCounter[A] at the time of the last frame transmission on channel A in the dynamic segment of this node.

It is updated at the end of the dynamic segment and is reset to zero if no frame was transmitted during the dynamic segment.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS of channel A.

21.2.9.3 FLXAnFRFSR — FlexRay FIFO Status Register

Access: FLXAnFRFSR is a read-only register that can be read in 32-bit units.
FLXAnFRFSRL is a read-only register that can be read in 16-bit units.
FLXAnFRFSRLL and FLXAnFRFSRLH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRFSR: <FLXAn_base> + 0318_H,
FLXAnFRFSRL: <FLXAn_base> + 0318_H,
FLXAnFRFSRLL: <FLXAn_base> + 0318_H, FLXAnFRFSRLH: <FLXAn_base> + 0319_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFFL[7:0]								—	—	—	—	—	RFO	RFCL	RFNE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.66 FLXAnFRFSR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 8	RFFL[7:0]	Receive FIFO Fill Level Flag
7 to 3	Reserved	When read, the value after reset is returned.
2	RFO	Receive FIFO Overrun Flag 0: No receive FIFO overrun detected 1: A receive FIFO overrun has been detected
1	RFCL	Receive FIFO Critical Level Flag 0: Receive FIFO below critical level 1: Receive FIFO critical level reached
0	RFNE	Receive FIFO Not Empty Flag 0: Receive FIFO is empty 1: Receive FIFO is not empty

(1) FLXAnFRFSR.RFFL

Receive FIFO Fill Level Flag

Indicates the number of FIFO buffers filled with new data not yet read by the Host. Maximum value is 128.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(2) FLXAnFRFSR.RFO

Receive FIFO Overrun Flag

The flag is set to 1 by the CC when a receive FIFO overrun is detected.

When a receive FIFO overrun occurs, the oldest message is overwritten with the actual received message. In addition, interrupt flag FLXAnFREIR.RFO is set to 1.

The flag is cleared by the next FIFO read access issued by the Host.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(3) FLXAnFRFSR.RFCL

Receive FIFO Critical Level Flag

This flag is set to 1 when the receive FIFO fill level FLXAnFRFSR.RFFL is equal or greater than the critical level as configured by FLXAnFRFCL.CL.

When FLXAnFRFSR.RFCL changes from 0 to 1 bit FLXAnFRSIR.RFCL is set to 1, and if enabled, an interrupt is generated.

The flag is cleared by the CC as soon as FLXAnFRFSR.RFFL drops below FLXAnFRFCL.CL.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(4) FLXAnFRFSR.RFNE

Receive FIFO Not Empty Flag

This flag is set to 1 by the CC when a received valid frame (data or null frame depending on rejection mask) was stored in the FIFO. In addition, interrupt flag FLXAnFRSIR.RFNE is set to 1.

The bit is reset to 0 after the Host has read all messages from the FIFO.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

21.2.9.4 FLXAnFRMHDF — FlexRay Message Handler Constraints Flags Register

Some constraints exist for the Message Handler regarding bus clock frequency, Message RAM configuration, and FlexRay bus traffic. To simplify software development, constraints violations are reported by setting flags in the FLXAnFRMHDF.

Access: FLXAnFRMHDF can be read or written in 32-bit units.
FLXAnFRMHDFL can be read or written in 16-bit units.
FLXAnFRMHDFLL and FLXAnFRMHDFLH can be read or written in 8-bit units.

Address: FLXAnFRMHDF: <FLXAn_base> + $031C_H$,
FLXAnFRMHDFL: <FLXAn_base> + $031C_H$,
FLXAnFRMHDFLL: <FLXAn_base> + $031C_H$, FLXAnFRMHDFLH: <FLXAn_base> + $031D_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	WAHP	TNSB	TNSA	TBFB	TBFA	FNFB	FNFA	SNUB	SNUA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.67 FLXAnFRMHDF Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8	WAHP	Write Attempt to Header Partition Flag 0: No write attempt to header partition 1: Write attempt to header partition
7	TNSB	Transmission Not Started Channel B Flag 0: No transmission not started on channel B 1: Transmission not started on channel B
6	TNSA	Transmission Not Started Channel A Flag 0: No transmission not started on channel A 1: Transmission not started on channel A
5	TBFB	Temporary buffer Access Failure B Flag 0: No TBF B access failure 1: TBF B access failure
4	TBFA	Temporary buffer Access Failure A Flag 0: No TBF A access failure 1: TBF A access failure
3	FNFB	Find Sequence Not Finished Channel B Flag 0: No find sequence not finished for channel B 1: Find sequence not finished for channel B
2	FNFA	Find Sequence Not Finished Channel A Flag 0: No find sequence not finished for channel A 1: Find sequence not finished for channel A
1	SNUB	Status Not Updated Channel B Flag 0: No overload condition occurred when updating MBS for channel B 1: MBS for channel B not updated
0	SNUA	Status Not Updated Channel A Flag 0: No overload condition occurred when updating MBS for channel A 1: MBS for channel A not updated

(1) FLXAnFRMHDF.WAHP

Write Attempt to Header Partition Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

Outside DEFAULT_CONFIG and CONFIG state this flag is set to 1 by the CC when the message handler tries to write message data into the header partition of the Message RAM due to faulty configuration of a message buffer. The write attempt is not executed, to protect the header partition from unintended write accesses.

When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(2) FLXAnFRMHDF.TNSB

Transmission Not Started Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set to 1 by the CC when the Message Handler was not ready to start a scheduled transmission on channel B at the action point of the configured slot.

When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(3) FLXAnFRMHDF.TNSA

Transmission Not Started Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set by the CC when the Message Handler was not ready to start a scheduled transmission on channel A at the action point of the configured slot.

When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(4) FLXAnFRMHDF.TBFB

Temporary buffer Access Failure B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set to 1 by the CC when a read or write access to TBF B requested by PRT (Protocol controller) B could not complete within the available time.

When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(5) FLXAnFRMHDF.TBFA

Temporary buffer Access Failure A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set to 1 by the CC when a read or write access to TBF A requested by PRT A could not complete within the available time.

When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(6) FLXAnFRMHDF.FNFB

Find Sequence Not Finished Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer).

When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(7) FLXAnFRMHDF.FNFA

Find Sequence Not Finished Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to finish a find sequence (scan of Message RAM for matching message buffer).

When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(8) FLXAnFRMHDF.SNUB

Status Not Updated Channel B Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to update a message buffer's status (FLXAnFRMBS).

When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

(9) FLXAnFRMHDF.SNUA

Status Not Updated Channel A Flag

Writing 0 has no effect on the bit value.

This bit is cleared by writing 1 to it.

This flag is set to 1 by the CC when the Message Handler, due to overload condition, was not able to update a message buffer's status (FLXAnFRMBS).

When this flag changes from 0 to 1 in addition interrupt flag FLXAnFREIR.MHF is set to 1.

Cleared when leaving CONFIG state or when entering STARTUP state.

Reset by the CHI command CLEAR_RAMS.

21.2.9.5 FLXAnFRTXRQi — FlexRay Transmission Request i (i = 1 to 4)

The four registers reflect the state of the TXR flags of all configured message buffers. The flags are valid for transmit buffers only. If the number of configured message buffers is less than 128, the unused TXR flags are invalid.

Access: FLXAnFRTXRQi is a read-only register that can be read in 32-bit units.
FLXAnFRTXRQiL and FLXAnFRTXRQiH are the read-only registers that can be read in 16-bit units.
FLXAnFRTXRQiLL, FLXAnFRTXRQiLH, FLXAnFRTXRQiHL, and FLXAnFRTXRQiHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRTXRQi: <FLXAn_base> + 0320_H + (i - 1) × 4_H,
FLXAnFRTXRQiL: <FLXAn_base> + 0320_H + (i - 1) × 4_H,
FLXAnFRTXRQiH: <FLXAn_base> + 0320_H + (i - 1) × 4_H + 2_H,
FLXAnFRTXRQiLL: <FLXAn_base> + 0320_H + (i - 1) × 4_H,
FLXAnFRTXRQiLH: <FLXAn_base> + 0320_H + (i - 1) × 4_H + 1_H,
FLXAnFRTXRQiHL: <FLXAn_base> + 0320_H + (i - 1) × 4_H + 2_H,
FLXAnFRTXRQiHH: <FLXAn_base> + 0320_H + (i - 1) × 4_H + 3_H,

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TXRo (o = (i - 1) × 32 + 31 to (i - 1) × 32 + 16)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXRo (o = (i - 1) × 32 + 15 to (i - 1) × 32 + 0)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.68 FLXAnFRTXRQi Register Contents

Bit Position	Bit Name	Function
31 to 0	TXRo	Transmission Request Flag o

(1) FLXAnFRTXRQi.TXRo (o = (i-1) × 32 to i × 32-1)

Transmission Request Flag o

If the flag is set to 1, the respective message buffer is ready for transmission or transmission of this message buffer is in progress.

In single-shot mode the flags are reset to 0 after transmission has completed.

This bit is cleared by the CHI command CLEAR_RAMS.

21.2.9.6 FLXAnFRNDATi — FlexRay New Data Register i (i = 1 to 4)

The four registers reflect the state of the ND flags of all configured message buffers. ND flags belonging to transmit buffers have no meaning. If the number of configured message buffers is less than 128, unused ND flags are invalid.

Access: FLXAnFRNDATi is a read-only register that can be read in 32-bit units.
FLXAnFRNDATiL and FLXAnFRNDATiH are the read-only registers that can be read in 16-bit units.
FLXAnFRNDATiLL, FLXAnFRNDATiLH, FLXAnFRNDATiHL, and FLXAnFRNDATiHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRNDATi: <FLXAn_base> + 0330_H + (i - 1) × 4_H,
FLXAnFRNDATiL: <FLXAn_base> + 0330_H + (i - 1) × 4_H,
FLXAnFRNDATiH: <FLXAn_base> + 0330_H + (i - 1) × 4_H + 2_H,
FLXAnFRNDATiLL: <FLXAn_base> + 0330_H + (i - 1) × 4_H,
FLXAnFRNDATiLH: <FLXAn_base> + 0330_H + (i - 1) × 4_H + 1_H,
FLXAnFRNDATiHL: <FLXAn_base> + 0330_H + (i - 1) × 4_H + 2_H,
FLXAnFRNDATiHH: <FLXAn_base> + 0330_H + (i - 1) × 4_H + 3_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	NDm (m = (i - 1) × 32 + 31 to (i - 1) × 32 + 16)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NDm (m = (i - 1) × 32 + 15 to (i - 1) × 32 + 0)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.69 FLXAnFRNDATi Register Contents

Bit Position	Bit Name	Function
31 to 0	NDm	New Data Flag m

(1) FLXAnFRNDATi.NDm (m = (i-1) × 32 to i × 32-1)

New Data Flag m

The flags are set to 1 when a valid received data frame matches the message buffer's filter configuration, independent of the payload length received or the payload length configured for that message buffer.

The flags are not set to 1 after reception of null frames except for message buffers belonging to the receive FIFO.

An ND flag is reset to 0 when the header section of the corresponding message buffer is reconfigured or when the data section has been transferred to the Output Buffer.

Cleared when leaving CONFIG state or when entering STARTUP state.

This bit is cleared by the CHI command CLEAR_RAMS.

21.2.9.7 FLXAnFRMBSCi — FlexRay Message Buffer Status Changed Register i (i = 1 to 4)

The four registers reflect the state of the MBC flags of all configured message buffers. If the number of configured message buffers is less than 128, unused MBC flags are invalid.

Access: FLXAnFRMBSCi is a read-only register that can be read in 32-bit units.
FLXAnFRMBSCiL and FLXAnFRMBSCiH are the read-only registers that can be read in 16-bit units.
FLXAnFRMBSCiLL, FLXAnFRMBSCiLH, FLXAnFRMBSCiHL, and FLXAnFRMBSCiHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRMBSCi: <FLXAn_base> + 0340_H + (i - 1) × 4_H,
FLXAnFRMBSCiL: <FLXAn_base> + 0340_H + (i - 1) × 4_H,
FLXAnFRMBSCiH: <FLXAn_base> + 0340_H + (i - 1) × 4_H + 2_H,
FLXAnFRMBSCiLL: <FLXAn_base> + 0340_H + (i - 1) × 4_H,
FLXAnFRMBSCiLH: <FLXAn_base> + 0340_H+ (i - 1) × 4_H + 1_H,
FLXAnFRMBSCiHL: <FLXAn_base> + 0340_H + (i - 1) × 4_H + 2_H,
FLXAnFRMBSCiHH: <FLXAn_base> + 0340_H + (i - 1) × 4_H + 3_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MBCm (m = (i - 1) × 32 + 31 to (i - 1) × 32 + 16)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MBCm (m = (i - 1) × 32 + 15 to (i - 1) × 32 + 0)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.70 FLXAnFRMBSCi (i=1 to 4) Register Contents

Bit Position	Bit Name	Function
31 to 0	MBCm	Message Buffer Status Changed Flag m

(1) FLXAnFRMBSCi.MBCm (m = (i-1) × 32 to i × 32-1)

Message Buffer Status Changed Flag m

Indicates whether the Message Handler has changed one of the status flags VFRA, VFRB, SEOA, SEOB, CEOA, CEOB, SVOA, SVOB, TCIA, TCIB, ESA, ESB, MLST, FTA, FTB in the header section (see **Section 21.2.11.5, FLXAnFRMBS — FlexRay Message Buffer Status Register** and **Section 21.3.13.1, Header Partition**) of the respective message buffer.

An MBC flag is reset to 0 when the header section of the corresponding message buffer is reconfigured or when it has been transferred to the Output Buffer.

Cleared when leaving CONFIG state or when entering STARTUP state.

This bit is cleared by the CHI command CLEAR_RAMS.

21.2.10 Input Buffer

Double buffer structure consisting of Input Buffer Host and Input Buffer Shadow. While the Host can write to Input Buffer Host, the transfer to the Message RAM is done from Input Buffer Shadow. The Input Buffer holds the header and data sections to be transferred to the selected message buffer in the Message RAM. It is used to configure the message buffers in the Message RAM and to update the data sections of transmit buffers.

When updating the header section of a message buffer in the Message RAM from the Input Buffer, the Message Buffer Status as described in **Section 21.2.11.5, FLXAnFRMBS — FlexRay Message Buffer Status Register** is automatically reset to zero.

The header sections of message buffers belonging to the receive FIFO can only be (re)configured when the CC is in DEFAULT_CONFIG or CONFIG state. For those message buffers only the payload length configured and the data pointer need to be configured via FLXAnFRWRHS2.PLC and FLXAnFRWRHS3.DP. All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask.

The data transfer between Input Buffer (IBF) and Message RAM is described in detail in **Section 21.3.12.1, Reconfiguration of Message Buffers**.

These registers cannot be written when the input data transfer function shown in **Section 21.3.16.1, Input Data Transfer** is used and the FLXAnFRITS.ITS bit is 1.

21.2.10.1 FLXAnFRWRDSx — FlexRay Write Data Section Register x (x = 1 to 64)

This register holds the data wards to be transferred to the data section of the specified message buffer. The number of data words written to the Message RAM is defined by the payload length configured in FLXAnFRWRHS2.PLC bit.

Access: FLXAnFRWRDSx can be read or written in 32-bit units.
FLXAnFRWRDSxL and FLXAnFRWRDSxH can be read or written in 16-bit units.
FLXAnFRWRDSxLL, FLXAnFRWRDSxLH, FLXAnFRWRDSxHL, and FLXAnFRWRDSxHH can be read or written in 8-bit units.

Address: FLXAnFRWRDSx: <FLXAn_base> + 0400_H + (x - 1) × 4_H,
FLXAnFRWRDSxL: <FLXAn_base> + 0400_H + (x - 1) × 4_H,
FLXAnFRWRDSxH: <FLXAn_base> + 0400_H + (x - 1) × 4_H + 2_H,
FLXAnFRWRDSxLL: <FLXAn_base> + 0400_H + (x - 1) × 4_H,
FLXAnFRWRDSxLH: <FLXAn_base> + 0400_H + (x - 1) × 4_H + 1_H,
FLXAnFRWRDSxHL: <FLXAn_base> + 0400_H + (x - 1) × 4_H + 2_H,
FLXAnFRWRDSxHH: <FLXAn_base> + 0400_H + (x - 1) × 4_H + 3_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.71 FLXAnFRWRDSx Register Contents

Bit Position	Bit Name	Function
31 to 0	MD[31:0]	Message Data Bit

(1) FLXAnFRWRDSx.MD

Message Data Bit

For information about the byte alignment of the message data in this register see **Section 21.3.17, Byte Alignment**.

CAUTIONS

1. In case FLXAnFRWRHS2.PLC specifies an odd payload length, the remaining message data bytes are unused.
2. When writing to FLXAnFRWRDSx, each 32-bit word has to be filled up by one 32-bit access OR two consecutive 16-bit accesses OR four consecutive 8- bit accesses before the transfer from the Input Buffer to the Message RAM is started. If not all bytes of a 32- bit word have been written by the Host (8/16-bit access only), FLXAnFRWRDSx holds partly undefined data.
Reset by the CHI command CLEAR_RAMS.

21.2.10.2 FLXAnFRWRHS1 — FlexRay Write Header Section Register 1

Access: FLXAnFRWRHS1 can be read or written in 32-bit units.
FLXAnFRWRHS1L and FLXAnFRWRHS1H can be read or written in 16-bit units.
FLXAnFRWRHS1LL, FLXAnFRWRHS1LH, FLXAnFRWRHS1HL, and FLXAnFRWRHS1HH can be read or written in 8-bit units.

Address: FLXAnFRWRHS1: <FLXAn_base> + 0500_H,
FLXAnFRWRHS1L: <FLXAn_base> + 0500_H, FLXAnFRWRHS1H: <FLXAn_base> + 0502_H,
FLXAnFRWRHS1LL: <FLXAn_base> + 0500_H, FLXAnFRWRHS1LH: <FLXAn_base> + 0501_H,
FLXAnFRWRHS1HL: <FLXAn_base> + 0502_H, FLXAnFRWRHS1HH: <FLXAn_base> + 0503_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MBI	TXM	PPIT	CFG	CH[1:0]		—	CYC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	FID[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.72 FLXAnFRWRHS1 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29	MBI	Message Buffer Interrupt Bit 0: The corresponding message buffer interrupt is disabled 1: The corresponding message buffer interrupt is enabled
28	TXM	Transmission Mode Setting Bit 0: Continuous mode 1: Single-shot mode
27	PPIT	Payload Preamble Indicator Transmit Bit 0: Payload Preamble Indicator is set to 0 1: Payload Preamble Indicator is set to 1
26	CFG	Message Buffer Direction Configuration Bit 0: The corresponding buffer is configured as Receive Buffer 1: The corresponding buffer is configured as Transmit Buffer
25,24	CH[1:0]	Channel Filter Control Bit
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	CYC[6:0]	Cycle Code Bit
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	FID[10:0]	Frame ID Bit

(1) FLXAnFRWRHS1.MBI

Message Buffer Interrupt Enable Bit

This bit enables the message buffer interrupt.

After a dedicated receive buffer has been updated by the Message Handler, flag FLXAnFRSIR.RXI and /or FLXAnFRSIR.MBSI are set to 1. After a transmission has completed flag FLXAnFRSIR.TXI is set to 1.

(2) FLXAnFRWRHS1.TXM

Transmission Mode Setting Bit

This bit selects transmit mode of the corresponding message buffer. For transmit mode, see **Section 21.3.9.3, Transmit Buffers**.

(3) FLXAnFRWRHS1.PPIT

Payload Preamble Indicator Transmit Bit

This bit is used to control the state of the Payload Preamble Indicator in transmit frames in transmit frames of the corresponding message buffer.

If the bit is set to 1 in a static message buffer, the respective message buffer holds network management information.

If the bit is set to 1 in a dynamic message buffer the first two bytes of the payload segment may be used for message ID filtering by the receiver. Message ID filtering of received FlexRay frames is not supported by the FlexRay module, but can be done by the Host.

(4) FLXAnFRWRHS1.CFG

Message Buffer Direction Configuration Bit

This bit is used to configure the corresponding buffer as transmit buffer or as receive buffer. For message buffers belonging to the receive FIFO the bit is not evaluated.

When not allocating an unused area of at least 32 bits at the start of the data partition, set this bit to 1 in the data section of the message buffer allocated immediately after the (last buffer of the) header partition so that the message buffer is specified as a transmit buffer.

(5) FLXAnFRWRHS1.CH

Channel Filter Control Bit

The 2-bit channel filtering field associated with each buffer serves as a filter for receive buffers, and as a control field for transmit buffers.

CH[1:0]	Transmit Buffer transmit frame on	Receive Buffer store frame received from
00_B	No transmission	Ignore frame
01_B	Channel A	Channel A
10_B	Channel B	Channel B
11_B	Both channels (static segment only)	Channel A or B (store first semantically valid frame; static segment only)

CAUTION

If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted and received frames are ignored (same function as CH = "00_B")

(6) FLXAnFRWRHS1.CYC

Cycle Code Bit

The 7-bit cycle code determines the cycle set used for cycle counter filtering.

For details about the configuration of the cycle code **Section 21.3.8.2, Cycle Counter Filtering**.

(7) FLXAnFRWRHS1.FID

Frame ID Bit

Configures the frame ID of the selected message buffer. The frame ID defines the slot number for transmission / reception of the respective message.

Message buffers with frame ID = 0 are considered as not valid.

21.2.10.3 FLXAnFRWRHS2 — FlexRay Write Header Section Register 2

Access: FLXAnFRWRHS2 can be read or written in 32-bit units.
FLXAnFRWRHS2L and FLXAnFRWRHS2H can be read or written in 16-bit units.
FLXAnFRWRHS2LL, FLXAnFRWRHS2LH, and FLXAnFRWRHS2HL can be read or written in 8-bit units.

Address: FLXAnFRWRHS2: <FLXAn_base> + 0504_H,
FLXAnFRWRHS2L: <FLXAn_base> + 0504_H, FLXAnFRWRHS2H: <FLXAn_base> + 0506_H,
FLXAnFRWRHS2LL: <FLXAn_base> + 0504_H, FLXAnFRWRHS2LH: <FLXAn_base> + 0505_H,
FLXAnFRWRHS2HL: <FLXAn_base> + 0506_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PLC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.73 FLXAnFRWRHS2 Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	PLC[6:0]	Payload Length Configured Bit
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	CRC[10:0]	Header CRC Bit (vRF!Header!HeaderCRC) Receive Buffer: Configuration not required Transmit Buffer: Header CRC is configured

(1) FLXAnFRWRHS2.PLC

Payload Length Configured Bit

Length of data section (number of 2-byte words) as configured by the Host.

During static segment the static frame payload length as configured by FLXAnFRMHDC.SFDL defines the payload length for all static frames. If the payload length configured by FLXAnFRWRHS2.PLC is shorter than this value padding bytes are inserted to ensure that frames have proper physical length. The padding pattern is "0000_H" (see **Section 21.3.9.3, Transmit Buffers**).

(2) FLXAnFRWRHS2.CRC

Header CRC Bit (vRF!Header!HeaderCRC)

Setting of the receive buffer is not required.

Transmitting of the message buffer needs the header CRC calculation and setting.

For calculation of the header CRC the payload length of the frame send on the bus has to be considered. In static segment the payload length of all frames is configured by FLXAnFRMHDC.SFDL.

21.2.10.4 FLXAnFRWRHS3 — FlexRay Write Header Section Register 3

Access: FLXAnFRWRHS3 can be read or written in 32-bit units.
FLXAnFRWRHS3L can be read or written in 16-bit units.
FLXAnFRWRHS3LL and FLXAnFRWRHS3LH can be read or written in 8-bit units.

Address: FLXAnFRWRHS3: <FLXAn_base> + 0508_H,
FLXAnFRWRHS3L: <FLXAn_base> + 0508_H,
FLXAnFRWRHS3LL: <FLXAn_base> + 0508_H, FLXAnFRWRHS3LH: <FLXAn_base> + 0509_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DP[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.74 FLXAnFRWRHS3 Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	DP[10:0]	Data Pointer Bit

(1) FLXAnFRWRHS3.DP

Data Pointer Bit

Configures the pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

21.2.10.5 FLXAnFRIBCM — FlexRay Input Buffer Command Mask Register

Configures how the message buffer in the Message RAM selected by register FLXAnFRIBCR is updated. When IBF Host and IBF Shadow are swapped, also mask bits FLXAnFRIBCM.LHSH, FLXAnFRIBCM.LDSH, and FLXAnFRIBCM.STXRH are swapped with bits FLXAnFRIBCM.LHSS, FLXAnFRIBCM.LDSS, and FLXAnFRIBCM.STXRS.

Access: FLXAnFRIBCM can be read or written in 32-bit units.
FLXAnFRIBCML can be read or written in 16-bit units.
FLXAnFRIBCMH is a read-only register that can be read in 16-bit units.
FLXAnFRIBCMLL can be read or written in 8-bit units.
FLXAnFRIBCMHL is a read-only register that can be read in 8-bit units.

Address: FLXAnFRIBCM: <FLXAn_base> + 0510_H,
FLXAnFRIBCML: <FLXAn_base> + 0510_H, FLXAnFRIBCMH: <FLXAn_base> + 0512_H,
FLXAnFRIBCMLL: <FLXAn_base> + 0510_H, FLXAnFRIBCMHL: <FLXAn_base> + 0512_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	STXRS	LDSS	LHSS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	STXRH	LDSH	LHSH
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 21.75 FLXAnFRIBCM Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	STXRS	Set Transmission Request Shadow Flag 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission (operation ongoing or finished)
17	LDSS	Load Data Section Shadow Flag 0: Data section is not updated 1: Data section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished)
16	LHSS	Load Header Section Shadow Flag 0: Header section is not updated 1: Header section selected for transfer from Input Buffer to the Message RAM (transfer ongoing or finished)
15 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	STXRH	Set Transmission Request Host Bit 0: Reset TXR flag 1: Set TXR flag, transmit buffer released for transmission
1	LDSH	Load Data Section Host Bit 0: Data section is not updated 1: Data section selected for transfer from Input Buffer to the Message RAM
0	LHSH	Load Header Section Host Bit 0: Header section is not updated 1: Header section selected for transfer from Input Buffer to the Message RAM

(1) FLXAnFRIBCM.STXRS

Set Transmission Request Shadow Flag

(2) FLXAnFRIBCM.LDSS

Load Data Section Shadow Flag

(3) FLXAnFRIBCM.LHSS

Load Header Section Shadow Flag

(4) FLXAnFRIBCM.STXRH

Set Transmission Request Host Bit

If this bit is set to 1, the TXR flag for the selected message buffer is set in the FLXAnFRTXRQ1/2/3/4 registers to release the message buffer for transmission. In single-shot mode the flag is cleared by the CC after transmission has completed.

TXR is evaluated for transmit buffers only.

(5) FLXAnFRIBCM.LDSH

Set Load Data Section Host Bit

(6) FLXAnFRIBCM.LHSH

Set Load Header Section Host Bit

21.2.10.6 FLXAnFRIBCR — FlexRay Input Buffer Command Request Register

When the Host writes the number of the target message buffer in the Message RAM to FLXAnFRIBCR.IBRH, IBF Host and IBF Shadow are swapped. In addition the message buffer numbers stored under FLXAnFRIBCR.IBRH and FLXAnFRIBCR.IBRS are also swapped (see **Section 21.3.12.2 (1), Data Transfer from Input Buffer to Message RAM**).

With this write operation the FLXAnFRIBCR.IBSYS is set to 1. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by FLXAnFRIBCR.IBRS.

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message into the IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, FLXAnFRIBCR.IBSYS is set back to 0 and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to FLXAnFRIBCR.IBRH.

If a write access to FLXAnFRIBCR.IBRH occurs while FLXAnFRIBCR.IBSYS is 1, FLXAnFRIBCR.IBSYH is set to 1. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, FLXAnFRIBCR.IBSYH is reset to 0. FLXAnFRIBCR.IBSYS remains set to 1, and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under FLXAnFRIBCR.IBRH and FLXAnFRIBCR.IBRS are also swapped.

Any write access to an Input Buffer register while both FLXAnFRIBCR.IBSYS and FLXAnFRIBCR.IBSYH are set to 1 will cause the error flag FLXAnFREIR.IIBA to be set to 1.

Access: FLXAnFRIBCR can be read or written in 32-bit units.
FLXAnFRIBCRL can be read or written in 16-bit units.
FLXAnFRIBCRH is a read-only register that can be read in 16-bit units.
FLXAnFRIBCRLL can be read or written in 8-bit units.
FLXAnFRIBCRLH, FLXAnFRIBCRHL, and FLXAnFRIBCRHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRIBCR: <FLXAn_base> + 0514_H,
FLXAnFRIBCRL: <FLXAn_base> + 0514_H, FLXAnFRIBCRH: <FLXAn_base> + 0516_H,
FLXAnFRIBCRLL: <FLXAn_base> + 0514_H, FLXAnFRIBCRLH: <FLXAn_base> + 0515_H,
FLXAnFRIBCRHL: <FLXAn_base> + 0516_H, FLXAnFRIBCRHH: <FLXAn_base> + 0517_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IBSYS	—	—	—	—	—	—	—	—	IBRS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IBSYH	—	—	—	—	—	—	—	—	IBRH[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.76 FLXAnFRIBCR Register Contents (1/2)

Bit Position	Bit Name	Function
31	IBSYS	Input Buffer Busy Shadow Flag 0: Transfer between IBF Shadow and Message RAM completed 1: Transfer between IBF Shadow and Message RAM in progress

Table 21.76 FLXAnFRIBCR Register Contents (2/2)

Bit Position	Bit Name	Function
30 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	IBRS[6:0]	Input Buffer Request Shadow Flag
15	IBSYH	Input Buffer Busy Host Flag 0: No request pending 1: Request while transfer between IBF Shadow and Message RAM in progress
14 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	IBRH[6:0]	Input Buffer Request Host Bit

(1) FLXAnFRIBCR.IBSYS

Input Buffer Busy Shadow Flag

Set to 1 after writing FLXAnFRIBCR.IBRH.

This bit indicates transmitting between the IBF Shadow and the Message RAM is ongoing.

When the transfer between IBF Shadow and the Message RAM has completed, FLXAnFRIBCR.IBSYS is set back to 0.

(2) FLXAnFRIBCR.IBRS

Input Buffer Request Shadow Flag

Number of the target message buffer actually updated / lately updated.

(3) FLXAnFRIBCR.IBSYH

Input Buffer Busy Host Flag

Set to 1 by writing FLXAnFRIBCR.IBRH while FLXAnFRIBCR.IBSYS is still 1.

This bit indicates transmitting between the IBF Shadow and the Message RAM is ongoing.

After the ongoing transfer between IBF Shadow and the Message RAM has completed, the FLXAnFRIBCR.IBSYH is set back to 0.

(4) FLXAnFRIBCR.IBRH

Input Buffer Request Host Bit

Selects the target message buffer in the Message RAM for data transfer from Input Buffer.

21.2.11 Output Buffer

Double buffer structure consisting of Output Buffer Host and Output Buffer Shadow. Used to read out message buffers from the Message RAM. While the Host can read from Output Buffer Host, the Message Handler transfers the selected message buffer from Message RAM to Output Buffer Shadow. The data transfer between Message RAM and Output Buffer (OBF) is described in **Section 21.3.12.2 (2), Data Transfer from Message RAM to Output Buffer**.

These registers cannot be written when the output data transfer function shown in **Section 21.3.16.2, Output Data Transfer**, in Output Data Transfer is used and the FLXAnFROTS.OTS bit is 1.

21.2.11.1 FLXAnFRRDDSx — FlexRay Read Data Section Register x (x = 1 to 64)

Holds the data words read from the data section of the addressed message buffer. This register holds the data wards to be transferred to the data section of the specified message buffer. The number of data words (DWn) read from the Message RAM is defined by the payload length configured in FLXAnFRRDHS2.PLC bit.

Access: FLXAnFRRDDSx is a read-only register that can be read in 32-bit units.
FLXAnFRRDDSxL and FLXAnFRRDDSxH are the read-only registers that can be read in 16-bit units.
FLXAnFRRDDSxLL, FLXAnFRRDDSxLH, FLXAnFRRDDSxHL, and FLXAnFRRDDSxHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRRDDSx: <FLXAn_base> + 0600_H + (x - 1) × 4_H,
FLXAnFRRDDSxL: <FLXAn_base> + 0600_H + (x - 1) × 4_H,
FLXAnFRRDDSxH: <FLXAn_base> + 0600_H + (x - 1) × 4_H + 2_H,
FLXAnFRRDDSxLL: <FLXAn_base> + 0600_H + (x - 1) × 4_H,
FLXAnFRRDDSxLH: <FLXAn_base> + 0600_H + (x - 1) × 4_H + 1_H,
FLXAnFRRDDSxHL: <FLXAn_base> + 0600_H + (x - 1) × 4_H + 2_H,
FLXAnFRRDDSxHH: <FLXAn_base> + 0600_H + (x - 1) × 4_H + 3_H,

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.77 FLXAnFRRDDSx Register Contents

Bit Position	Bit Name	Function
31 to 0	MD[31:0]	Message Data

(1) FLXAnFRRDDSx.MD

Message Data Flag

For information about the byte alignment of the data words in this register see **Section 21.3.17, Byte Alignment**.

CAUTION

In case FLXAnFRWRHS2.PLC specifies an odd payload length, the remaining message data bytes are unused.

Reset by the CHI command CLEAR_RAMS.

21.2.11.2 FLXAnFRRDHS1 — FlexRay Read Header Section Register 1

Access: FLXAnFRRDHS1 is a read-only register that can be read in 32-bit units.
FLXAnFRRDHS1L and FLXAnFRRDHS1H are the read-only registers that can be read in 16-bit units.
FLXAnFRRDHS1LL, FLXAnFRRDHS1LH, FLXAnFRRDHS1HL, and FLXAnFRRDHS1HH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRRDHS1: <FLXAn_base> + 0700_H,
FLXAnFRRDHS1L: <FLXAn_base> + 0700_H, FLXAnFRRDHS1H: <FLXAn_base> + 0702_H,
FLXAnFRRDHS1LL: <FLXAn_base> + 0700_H, FLXAnFRRDHS1LH: <FLXAn_base> + 0701_H,
FLXAnFRRDHS1HL: <FLXAn_base> + 0702_H, FLXAnFRRDHS1HH: <FLXAn_base> + 0703_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	MBI	TXM	PPIT	CFG	CH[1:0]		—	CYC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	FID[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.78 FLXAnFRRDHS1 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29	MBI	Message Buffer Interrupt Flag
28	TXM	Transmission Mode Flag
27	PPIT	Payload Preamble Indicator Transmit Flag
26	CFG	Message Buffer Direction Configuration Flag
25, 24	CH[1:0]	Channel Filter Control Flag
23	Reserved	When read, the value after reset is returned.
22 to 16	CYC[6:0]	Cycle Code
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	FID[10:0]	Frame ID

(1) FLXAnFRRDHS1.MBI

Message Buffer Interrupt Flag

Values as configured by the Host via FLXAnFRWRHS1.MBI.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(2) FLXAnFRRDHS1.TXM

Transmission Mode Flag

Values as configured by the Host via FLXAnFRWRHS1.TXM.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(3) FLXAnFRRDHS1.PPIT

Payload Preamble Indicator Transmit Flag

Values as configured by the Host via FLXAnFRWRHS1.PPIT.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(4) FLXAnFRRDHS1.CFG

Message Buffer Direction Configuration Flag

Values as configured by the Host via FLXAnFRWRHS1.CFG.

In case that the message buffer read from the Message RAM belongs to the receive FIFO this bit is set to 0.

(5) FLXAnFRRDHS1.CH

Channel Filter Control Flag

Values as configured by the Host via FLXAnFRWRHS1.CH.

In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are set to 0.

(6) FLXAnFRRDHS1.CYC

Cycle Code

Values as configured by the Host via FLXAnFRWRHS1.CYC.

In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are set to 0.

(7) FLXAnFRRDHS1.FID

Frame ID

Values as configured by the Host via FLXAnFRWRHS1.FID

In case that the message buffer read from the Message RAM belongs to the receive FIFO these bits are holding the received frame ID.

21.2.11.3 FLXAnFRRDHS2 — FlexRay Read Header Section Register 2

Access: FLXAnFRRDHS2 is a read-only register that can be read in 32-bit units.
FLXAnFRRDHS2L and FLXAnFRRDHS2H are the read-only registers that can be read in 16-bit units.
FLXAnFRRDHS2LL, FLXAnFRRDHS2LH, FLXAnFRRDHS2HL, and FLXAnFRRDHS2HH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRRDHS2: <FLXAn_base> + 0704_H,
FLXAnFRRDHS2L: <FLXAn_base> + 0704_H, FLXAnFRRDHS2H: <FLXAn_base> + 0706_H,
FLXAnFRRDHS2LL: <FLXAn_base> + 0704_H, FLXAnFRRDHS2LH: <FLXAn_base> + 0705_H,
FLXAnFRRDHS2HL: <FLXAn_base> + 0706_H, FLXAnFRRDHS2HH: <FLXAn_base> + 0707_H

Value after reset: $0000\ 0000_H$

CAUTION

For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area FLXAnFRWRHS2 is updated from data frames only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PLR[6:0]							—	PLC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CRC[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.79 FLXAnFRRDHS2 Register Contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
30 to 24	PLR[6:0]	Payload Length Received Flag (vRF!Header!Length)
23	Reserved	When read, the value after reset is returned.
22 to 16	PLC[6:0]	Payload Length Configured Flag
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	CRC[10:0]	Header CRC Flag (vRF!Header!HeaderCRC)

(1) FLXAnFRRDHS2.PLR

Payload Length Received Flag (vRF!Header!Length)

Payload length (vRF!Header!Length) value updated from received data frames (exception: if message buffer belongs to the receive FIFO FLXAnFRRDSH2.PLR is also updated from received null frames).

(2) FLXAnFRRDHS2.PLC

Payload Length Configured Flag

Length of data section (number of 2-byte words) as configured by the Host.

(3) FLXAnFRRDHS2.CRC

Header CRC Flag (vRF!Header!HeaderCRC)

Receive Buffer: Header CRC (vRF!Header!HeaderCRC) updated from received data frames

Transmit Buffer: Header CRC configured by the Host

(4) Data storage

When a message is stored into a message buffer the following behavior with respect to payload length received and payload length configured is implemented:

FLXAnFRRDHS2.PLR > FLXAnFRRDHS2.PLC:

The payload data stored in the message buffer is truncated to the payload length configured if FLXAnFRRDHS2.PLC even or else truncated to FLXAnFRRDHS2.PLC + 1.

FLXAnFRRDHS2.PLR ≤ FLXAnFRRDHS2.PLC:

The received payload data is stored into the message buffers data section. The remaining data bytes of the data section as configured by FLXAnFRRDHS2.PLC are filled with undefined data.

FLXAnFRRDHS2.PLR = zero:

The message buffer’s data section is filled with undefined data

FLXAnFRRDHS2.PLC = zero:

Message buffer has no data section configured. No data is stored into the message buffer's data section.

CAUTIONS

1. The Message RAM is organized in 4-byte words. When received data is stored into a message buffer's data section, the number of 2-byte data words written into the message buffer is FLXAnFRRDHS2.PLC rounded to the next even value.
2. FLXAnFRRDHS2.PLC should be configured identical for all message buffers belonging to the receive FIFO. Header 2 is updated from data frames only.
   For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area FLXAnFRWRHS2 is updated from data frames only.

21.2.11.4 FLXAnFRRDHS3 — FlexRay Read Header Section Register 3

CAUTION

For message buffers belonging to the Static Buffer area or Static and Dynamic Buffer area FLXAnFRWRHS3 is updated from data frames only.

Access: FLXAnFRRDHS3 is a read-only register that can be read in 32-bit units.
FLXAnFRRDHS3L and FLXAnFRRDHS3H are the read-only registers that can be read in 16-bit units.
FLXAnFRRDHS3LL, FLXAnFRRDHS3LH, FLXAnFRRDHS3HL, and FLXAnFRRDHS3HH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRRDHS3: <FLXAn_base> + 0708_H,
FLXAnFRRDHS3L: <FLXAn_base> + 0708_H, FLXAnFRRDHS3H: <FLXAn_base> + $070A_H$,
FLXAnFRRDHS3LL: <FLXAn_base> + 0708_H, FLXAnFRRDHS3LH: <FLXAn_base> + 0709_H,
FLXAnFRRDHS3HL: <FLXAn_base> + $070A_H$, FLXAnFRRDHS3HH: <FLXAn_base> + $070B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	RES	PPI	NFI	SYN	SFI	RCI	—	—	RCC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DP[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.80 FLXAnFRRDHS3 Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29	RES	Reserved Bit Indicator Flag (vRF!Header!Reserved)
28	PPI	Payload Preamble Indicator (vRF!Header!PPIndicator)
27	NFI	Null Frame Indicator Flag (vRF!Header!NFIndicator) 0: Up to now no data frame has been stored into the respective message buffer 1: At least one data frame has been stored into the respective message buffer
26	SYN	Sync Frame Indicator Flag (vRF!Header!SyFIndicator) 0: The received frame is not a sync frame 1: The received frame is a sync frame
25	SFI	Startup Frame Indicator Flag (vRF!Header!SuFIndicator) 0: The received frame is not a startup frame 1: The received frame is a startup frame
24	RCI	Received on Channel Indicator Flag (vSS!Channel) 0: Frame received on channel B 1: Frame received on channel A
23, 22	Reserved	When read, the value after reset is returned.
21 to 16	RCC[5:0]	Receive Cycle Counter (vRF!Header!CycleCount)
15 to 11	Reserved	When read, the value after reset is returned.
10 to 0	DP[10:0]	Data Pointer Flag

(1) FLXAnFRRDHS3.RES

Reserved Bit Flag (vRF!Header!Reserved)

Reflects the state of the received reserved bit. The reserved bit is transmitted as 0.

(2) FLXAnFRRDHS3.PPI

Payload Preamble Indicator Flag (vRF!Header!PPIndicator)

The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.

0 = The payload segment of the received frame does not contain a network management vector nor a message ID

1 = Static segment: Network management vector in the first part of the payload
Dynamic segment: Message ID in the first part of the payload

(3) FLXAnFRRDHS3.NFI

Null Frame Indicator Flag (vRF!Header!NFIndicator)

Is set to 1 after storage of the first received data frame.

(4) FLXAnFRRDHS3.SYN

Sync Frame Indicator Flag (vRF!Header!SyFIndicator)

A sync frame is marked by the sync frame indicator.

(5) FLXAnFRRDHS3.SFI

Startup Frame Indicator Flag (vRF!Header!SuFIndicator)

A startup frame is marked by the startup frame indicator.

(6) FLXAnFRRDHS3.RCI

Received on Channel Indicator Flag (vSS!Channel)

Indicates the channel from which the received data frame was taken to update the respective receive buffer.

(7) FLXAnFRRDHS3.RCC

Receive Cycle Counter (vRF!Header!CycleCount)

Cycle counter value updated from received data frame.

(8) FLXAnFRRDHS3.DP

Data Pointer Flag

Pointer to the first 32-bit word of the data section of the addressed message buffer in the Message RAM.

The bit value is the same as that set in the FLXAnFRWRHS3.DP bit.

21.2.11.5 FLXAnFRMBS — FlexRay Message Buffer Status Register

The message buffer status is updated by the CC with respect to the assigned channel(s) latest at the end of the slot following the slot assigned to the message buffer.

The flags are updated only when the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state.

If only one channel (A or B) is assigned to a message buffer, the channel-specific status flags of the other channel are written to zero. If both channels are assigned to a message buffer, the channel-specific status flags of both channels are updated.

The message buffer status is updated only when the slot counter reached the configured frame ID and when the cycle counter filter matched. When the Host updates a message buffer via Input Buffer, all FLXAnFRMBS flags are reset to zero independent of which FLXAnFRIBCM bits are set or not.

For details about receive / transmit filtering see **Section 21.3.8, Filtering and Masking**, **Section 21.3.9, Transmit Process** and **Section 21.3.10, Receive Process**.

Whenever the Message Handler changes one of the flags FLXAnFRMBS.VFRA, FLXAnFRMBS.VFRB, FLXAnFRMBS.SEOA, FLXAnFRMBS.SEOB, FLXAnFRMBS.CEOA, FLXAnFRMBS.CEOB, FLXAnFRMBS.SVOA, FLXAnFRMBS.SVOB, FLXAnFRMBS.TCIA, FLXAnFRMBS.TCIB, FLXAnFRMBS.ESA, FLXAnFRMBS.ESB, FLXAnFRMBS.MLST, FLXAnFRMBS.FTA, FLXAnFRMBS.FTB the respective message buffer's MBC flag in registers FLXAnFRMBSC1/2/3/4 is set.

Access: FLXAnFRMBS is a read-only register that can be read in 32-bit units.
FLXAnFRMBSL and FLXAnFRMBSH are the read-only registers that can be read in 16-bit units.
FLXAnFRMBSLL, FLXAnFRMBSLH, FLXAnFRMBSHL, and FLXAnFRMBSHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRMBS: <FLXAn_base> + $070C_H$,
FLXAnFRMBSL: <FLXAn_base> + $070C_H$, FLXAnFRMBSH: <FLXAn_base> + $070E_H$,
FLXAnFRMBSLL: <FLXAn_base> + $070C_H$, FLXAnFRMBSLH: <FLXAn_base> + $070D_H$,
FLXAnFRMBSHL: <FLXAn_base> + $070E_H$, FLXAnFRMBSHH: <FLXAn_base> + $070F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	RESS	PPIS	NFIS	SYNS	SFIS	RCIS	—	—	CCS[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTB	FTA	—	MLST	ESB	ESA	TCIB	TCIA	SVOB	SVOA	CEOB	CEOA	SEOB	SEOA	VFRB	VFRA
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.81 FLXAnFRMBS Register Contents (1/3)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned.
29	RESS	Reserved Bit Status Flag (vRF!Header!Reserved)
28	PPIS	Payload Preamble Indicator Status Flag (vRF!Header!PPIndicator) 0: PPI indicator set to 0 1: PPI indicator set to 1
27	NFIS	Null Frame Indicator Status Flag (vRF!Header!NFIndicator) 0: Received frame is a null frame 1: Received frame is not a null frame

Table 21.81 FLXAnFRMBS Register Contents (2/3)

Bit Position	Bit Name	Function
26	SYNS	Sync Frame Indicator Status Flag (vRF!Header!SyFIndicator) 0: No sync frame received 1: The received frame is a sync frame
25	SFIS	Startup Frame Indicator Status Flag (vRF!Header!SuFIndicator) 0: No startup frame received 1: The received frame is a startup frame
24	RCIS	Received on Channel Indicator Status Flag (vSS!Channel) 0: Frame received on channel B 1: Frame received on channel A
23, 22	Reserved	When read, the value after reset is returned.
21 to 16	CCS[5:0]	Cycle Count Status Flag
15	FTB	Frame Transmitted on Channel B Flag 0: No data frame transmitted on channel B 1: Data frame transmitted on channel B
14	FTA	Frame Transmitted on Channel A Flag 0: No data frame transmitted on channel A 1: Data frame transmitted on channel A
13	Reserved	When read, the value after reset is returned.
12	MLST	Message Lost Flag 0: No message lost 1: Unprocessed message was overwritten
11	ESB	Empty Slot Channel B Flag 0: Bus activity detected in the assigned slot on channel B 1: No bus activity detected in the assigned slot on channel B
10	ESA	Empty Slot Channel A Flag 0: Bus activity detected in the assigned slot on channel A 1: No bus activity detected in the assigned slot on channel A
9	TCIB	Transmission Conflict Indication Channel B Flag (vSS!TxConflictB) 0: No transmission conflict occurred on channel B 1: Transmission conflict occurred on channel B
8	TCIA	Transmission Conflict Indication Channel A Flag (vSS!TxConflictA) 0: No transmission conflict occurred on channel A 1: Transmission conflict occurred on channel A
7	SVOB	Slot Boundary Violation Observed on Channel B Flag (vSS!BViolationB) 0: No slot boundary violation observed on channel B 1: Slot boundary violation observed on channel B
6	SVOA	Slot Boundary Violation Observed on Channel A Flag (vSS!BViolationA) 0: No slot boundary violation observed on channel A 1: Slot boundary violation observed on channel A
5	CEOB	Content Error Observed on Channel B Flag (vSS!ContentErrorB) 0: No content error observed on channel B 1: Content error observed on channel B
4	CEOA	Content Error Observed on Channel A Flag (vSS!ContentErrorA) 0: No content error observed on channel A 1: Content error observed on channel A
3	SEOB	Syntax Error Observed on Channel B Flag (vSS!SyntaxErrorB) 0: No syntax error observed on channel B 1: Syntax error observed on channel B
2	SEOA	Syntax Error Observed on Channel A Flag (vSS!SyntaxErrorA) 0: No syntax error observed on channel A 1: Syntax error observed on channel A
1	VFRB	Valid Frame Received on Channel B (vSS!ValidFrameB) 0: No valid frame received on channel B 1: Valid frame received on channel B

Table 21.81 FLXAnFRMBS Register Contents (3/3)

Bit Position	Bit Name	Function
0	VFRA	Valid Frame Received on Channel A Flag (vSS!ValidFrameA) 0: No valid frame received on channel A 1: Valid frame received on channel A

(1) FLXAnFRMBS.RESS

Reserved Bit Status Flag (vRF!Header!Reserved)

Reflects the state of the received reserved bit. The reserved bit is transmitted as 0.

For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(2) FLXAnFRMBS.PPIS

Payload Preamble Indicator Status Flag (vRF!Header!PPIndicator)

The payload preamble indicator defines whether a network management vector or message ID is contained within the payload segment of the received frame.

For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

0 = PPI indicator set to 0
The payload segment of the received frame does not contain a network management vector or a message ID

1 = PPI indicator set to 1
Static segment: Network management vector at the beginning of the payload
Dynamic segment: Message ID at the beginning of the payload

(3) FLXAnFRMBS.NFIS

Null Frame Indicator Status Flag (vRF!Header!NFIndicator)

If set to 0 the payload segment of the received frame contains no usable data.

For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(4) FLXAnFRMBS.SYNS

Sync Frame Indicator Status Flag (vRF!Header!SyFIndicator)

A sync frame is marked by the sync frame indicator.

For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(5) FLXAnFRMBS.SFIS

Startup Frame Indicator Status Flag (vRF!Header!SuFIndicator)

The startup frame indicator specifies a startup frame.

For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(6) FLXAnFRMBS.RCIS

Received on Channel Indicator Status Flag (vSS!Channel)

Indicates the channel on which the frame was received.

For receive buffers (FLXAnFRWRHS1.CFG = 0) this bit is updated from both valid data and null frames. If no valid frame was received, the previous value is maintained. For transmit buffers the flag has no meaning and should be ignored.

(7) FLXAnFRMBS.CCS

Cycle Count Status Flag

Actual cycle count when status was updated.

(8) FLXAnFRMBS.FTB

Frame Transmitted on Channel B Flag

Indicates that this node has transmitted a data frame in the configured slot on channel B.

CAUTION

The FlexRay protocol specification requires that FLXAnFRMBS.FTB can only be reset by the Host. Therefore the Cycle Count Status FLXAnFRMBS.CCS for this bit is only valid for the cycle where the bit is set to 1.

(9) FLXAnFRMBS.FTA

Frame Transmitted on Channel A Flag

Indicates that this node has transmitted a data frame in the configured slot on channel A.

CAUTION

The FlexRay protocol specification requires that FLXAnFRMBS.FTA can only be reset by the Host. Therefore the Cycle Count Status FLXAnFRMBS.CCS for this bit is only valid for the cycle where this bit is set to 1.

(10) FLXAnFRMBS.MLST

Message Lost Flag

The flag is set in case the Host did not read the message before the message buffer was updated from a received data frame.

Not affected by reception of null frames except for message buffers belonging to the receive FIFO. The flag is reset to 0 by a Host write to the message buffer via IBF or when a new message is stored into the message buffer after the message buffers ND flag was reset to 0 by reading out the message buffer via OBF.

(11) FLXAnFRMBS.ESB

Empty Slot Channel B Flag

In an empty slot, there is no activity on the bus. This means that any frame transmission is not detected. This state can be checked in static and dynamic slots.

(12) FLXAnFRMBS.ESA

Empty Slot Channel A Flag

In an empty slot, there is no activity on the bus. This means that any frame transmission is not detected. This state can be checked in static and dynamic slots.

(13) FLXAnFRMBS.TCIB

Transmission Conflict Indication Channel B Flag (vSS!TxConflictB)

A transmission conflict indication is set to 1 if a transmission conflict has occurred on channel B.

(14) FLXAnFRMBS.TCIA

Transmission Conflict Indication Channel A Flag (vSS!TxConflictA)

A transmission conflict indication is set if a transmission conflict has occurred on channel A.

(15) FLXAnFRMBS.SVOB

Slot Boundary Violation Observed on Channel B Flag (vSS!BViolationB)

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on the slot assigned to channel B.

(16) FLXAnFRMBS.SVOA

Slot Boundary Violation Observed on Channel A Flag (vSS!BViolationA)

A slot boundary violation (channel active at the start or at the end of the assigned slot) was observed on the slot assigned to channel A.

(17) FLXAnFRMBS.CEOB

Content Error Observed on Channel B Flag (vSS!ContentErrorB)

A content error was observed in the slot assigned to channel B.

(18) FLXAnFRMBS.CEOA

Content Error Observed on Channel A Flag (vSS!ContentErrorA)

A content error was observed in the slot assigned to channel A.

(19) FLXAnFRMBS.SEOB

Syntax Error Observed on Channel B Flag (vSS!SyntaxErrorB)

A syntax error was observed in the assigned slot on channel B.

(20) FLXAnFRMBS.SEOA

Syntax Error Observed on Channel A Flag (vSS!SyntaxErrorA)

A syntax error was observed in the assigned slot on channel A.

(21) FLXAnFRMBS.VFRB

Valid Frame Received on Channel B Flag (vSS!ValidFrameB)

A valid frame indication is set if a valid frame was received on channel B.

(22) FLXAnFRMBS.VFRA

Valid Frame Received on Channel A Flag (vSS!ValidFrameA)

A valid frame indication is set if a valid frame was received on channel A.

21.2.11.6 FLXAnFROBCM — FlexRay Output Buffer Command Mask Register

Configures how the Output Buffer is updated from the message buffer in the Message RAM selected by FLXAnFROBCR.OBRS.

Mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to the register internal storage when a Message RAM transfer is requested by FLXAnFROBCR.REQ.

When OBF Host and OBF Shadow are swapped, mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the register internal storage to keep them attached to the respective Output Buffer transfer.

The data transfer between Output Buffer and Message RAM is described in detail in **Section 21.3.12.2 (2), Data Transfer from Message RAM to Output Buffer**.

CAUTION

After the transfer of the header section from the Message RAM to OBF Shadow has completed, the message buffer status changed flag MBC of the selected message buffer in the FLXAnFRMBSC1/2/3/4 registers is cleared. After the transfer of the data section from the Message RAM to OBF Shadow has completed, the new data flag ND of the selected message buffer in the FLXAnFRNDAT1/2/3/4 registers is cleared.

Access: FLXAnFROBCM can be read or written in 32-bit units.
FLXAnFROBCML can be read or written in 16-bit units.
FLXAnFROBCMH is a read-only register that can be read in 16-bit units.
FLXAnFROBCMLL can be read or written in 8-bit units.
FLXAnFROBCMHL is a read-only register that can be read in 8-bit units.

Address: FLXAnFROBCM: <FLXAn_base> + 0710_H,
FLXAnFROBCML: <FLXAn_base> + 0710_H, FLXAnFROBCMH: <FLXAn_base> + 0712_H,
FLXAnFROBCMLL: <FLXAn_base> + 0710_H, FLXAnFROBCMHL: <FLXAn_base> + 0712_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDSH	RHSH
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RDSS	RHSS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 21.82 FLXAnFROBCM Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
17	RDSH	Read Data Section Host Flag 0: Data section is not read 1: Data section selected for transfer from Message RAM to Output Buffer
16	RHSH	Read Header Section Host Flag 0: Header section is not read 1: Header section selected for transfer from Message RAM to Output Buffer
15 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 21.82 FLXAnFROBCM Register Contents

Bit Position	Bit Name	Function
1	RDSS	Read Data Section Shadow Bit 0: Data section is not read 1: Data section selected for transfer from Message RAM to Output Buffer
0	RHSS	Read Header Section Shadow Bit 0: Header section is not read 1: Header section selected for transfer from Message RAM to Output Buffer

(1) FLXAnFROBCM.RDSH

Read Data Section Host Flag

(2) FLXAnFROBCM.RHSH

Read Header Section Host Flag

(3) FLXAnFROBCM.RDSS

Read Data Section Shadow Bit

(4) FLXAnFROBCM.RHSS

Read Header Section Shadow Bit

21.2.11.7 FLXAnFROBCR — FlexRay Output Buffer Command Request Register

After setting bit FLXAnFROBCR.REQ to 1 while FLXAnFROBCR.OBSYS is 0, FLXAnFROBCR.OBSYS is automatically set to 1, FLXAnFROBCR.OBRS is copied to the register internal storage, mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to register FLXAnFROBCM internal storage, and the transfer of the message buffer selected by FLXAnFROBCR.OBRS from the Message RAM to OBF Shadow is started. When the transfer between the Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCM.OBSYS back to 0.

By setting bit FLXAnFROBCR.VIEW to 1 while FLXAnFROBCR.OBSYS is 0, OBF Host and OBF Shadow are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the register FLXAnFROBCM internal storage to keep them attached to the respective Output Buffer transfer. FLXAnFROBCR.OBRH signals the number of the message buffer currently accessible by the Host.

If bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW are set to 1 with the same write access while FLXAnFROBCR.OBSYS is 0, FLXAnFROBCR.OBSYS is automatically set to 1 and OBF Shadow and OBF Host are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards FLXAnFROBCR.OBRS is copied to the register internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCR.OBSYS back to 0.

Any write access to FLXAnFROBCR[15:8] while FLXAnFROBCR.OBSYS is set to 1 will cause the error flag FLXAnFREIR.IOBA to be set to 1. n this case, this write access has no effect and the Output Buffer will not be changed.

The data transfer between Output Buffer and Message RAM is described in detail in **Section 21.3.12.2 (2), Data Transfer from Message RAM to Output Buffer**.

Access: FLXAnFROBCR can be read or written in 32-bit units.
FLXAnFROBCRL can be read or written in 16-bit units.
FLXAnFROBCRH is a read-only register that can be read in 16-bit units.
FLXAnFROBCRLL and FLXAnFROBCRLH can be read or written in 8-bit units.
FLXAnFROBCRHL is a read-only register that can be read in 8-bit units.

Address: FLXAnFROBCR: <FLXAn_base> + 0714_H,
FLXAnFROBCRL: <FLXAn_base> + 0714_H, FLXAnFROBCRH: <FLXAn_base> + 0716_H,
FLXAnFROBCRLL: <FLXAn_base> + 0714_H, FLXAnFROBCRLH: <FLXAn_base> + 0715_H,
FLXAnFROBCRHL: <FLXAn_base> + 0716_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	OBRH[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OBSYS	—	—	—	—	—	REQ	VIEW	—	OBRS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.83 FLXAnFROBCR Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	OBRH[6:0]	Output Buffer Request Host Flag
15	OBSYS	Output Buffer Busy Shadow Flag 0: No transfer in progress 1: Transfer between Message RAM and OBF Shadow in progress
14 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	REQ	Request Message RAM Transfer Bit 0: No request 1: Transfer to OBF Shadow requested
8	VIEW	View Shadow Buffer Bit 0: No action 1: Swap OBF Shadow and OBF Host
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	OBRS[6:0]	Output Buffer Request Shadow Bit

(1) FLXAnFROBCR.OBRH

Output Buffer Request Host Flag

Number of message buffer currently accessible by the Host via FLXAnFRRDHS1 to FLXAnFRRDHS3, FLXAnFRMBS, and FLXAnFRRDDS1 to FLXAnFRRDDS64.

By writing FLXAnFROBCR.VIEW to 1 OBF Shadow and OBF Host are swapped and the transferred message buffer is accessible by the Host.

(2) FLXAnFROBCR.OBSYS

Output Buffer Busy Shadow Flag

Set to 1 after setting bit FLXAnFROBCR.REQ. When the transfer between the Message RAM and OBF Shadow has completed, FLXAnFROBCR.OBSYS is set back to 0.

(3) FLXAnFROBCR.REQ

Request Message RAM Transfer Bit

Only writeable while FLXAnFROBCR.OBSYS = 0.

Requests transfer of message buffer addressed by FLXAnFROBCR.OBRS from Message RAM to OBF Shadow.

(4) FLXAnFROBCR.VIEW

View Shadow Buffer Bit

Only writeable while FLXAnFROBCR.OBSYS = 0.

Toggles between OBF Shadow and OBF Host.

(5) FLXAnFROBCR.OBRS

Output Buffer Request Shadow Bit

Only writeable while FLXAnFROBCR.OBSYS = 0.

Number of source message buffer to be transferred from the Message RAM to OBF Shadow.

If the number of the first message buffer of the receive FIFO is written to this register the Message Handler transfers the message buffer addressed by the GET Index (GIDX, see **Section 21.3.11, FIFO Function**) to OBF Shadow.

21.2.12 Data Transfer Control Register

21.2.12.1 FLXAnFRITC — FlexRay Input Transfer Configuration Register

Access: FLXAnFRITC can be read or written in 32-bit units.
FLXAnFRITCL and FLXAnFRITCH can be read or written in 16-bit units.
FLXAnFRITCLL, FLXAnFRITCLH, and FLXAnFRITCHL can be read or written in 8-bit units.

Address: FLXAnFRITC: <FLXAn_base> + 0800_H,
FLXAnFRITCL: <FLXAn_base> + 0800_H, FLXAnFRITCH: <FLXAn_base> + 0802_H,
FLXAnFRITCLL: <FLXAn_base> + 0800_H, FLXAnFRITCLH: <FLXAn_base> + 0801_H,
FLXAnFRITCHL: <FLXAn_base> + 0802_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	ITM[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IQEIE	IQFIE	—	—	—	—	—	—	IQHR	ITE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 21.84 FLXAnFRITC Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	ITM[6:0]	Input queue Table Max Bit These bits configure the number of entries in the input pointer table the input buffer handler is capable to maintain in the input queue.
15 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	IQEIE	Input Queue Empty Interrupt Enable Bit 0: Disabled 1: Enabled
8	IQFIE	Input Queue Full Interrupt Enable Bit 0: Disabled 1: Enabled
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	IQHR	Input Queue Halt Request Bit 0: Input queue run request 1: Input queue halt request
0	ITE	Input Transfer Enable Bit 0: Operation Disable request 1: Operation Enable request

(1) FLXAnFRITC.ITM

Input queue Table Max Bit

The user can only write to this bit when FLXAnFRITS.ITS is 0.

These bits configure the number of entries in the input pointer table the input buffer handler is capable to maintain in the input queue.

Valid values are 00H (1 queue entry) to 7FH (128 queue entries).

Note that each entry requires two long words in the input pointer table.

(2) FLXAnFRITC.IQEIE

Input Queue Empty Interrupt Enable Bit

This bit controls the input queue empty interrupt.

0: Disabled
No interrupt will be requested and the input queue empty interrupt line will be released.

1: Enabled
Input queue empty interrupt will be asserted when FLXAnFRITS.IQEIS is 1.

(3) FLXAnFRITC.IQFIE

Input Queue Full Interrupt Enable Bit

This bit controls the input queue full interrupt.

0: Disabled
No interrupt will be requested and the input queue full interrupt line will be released.

1: Enabled
Input queue full interrupt will be asserted when FLXAnFRITS.IQFIS is 1.

(4) FLXAnFRITC.IQHR

Input Queue Halt Request Bit

The IQHR bit should not be set to 1 when FLXAnFRITS.ITS is 0.

This bit requests a halt of the input queue.

The status of the halt request is shown in the FLXAnFRITS.IQH register.

Refer to **Section 21.3.16.1 (5), Halting the input queue** about usage of this bit.

0: Input queue run request
The input queue resumes their operation.

1: Input queue halt request
The input queue gets halted. An active input transfer will be completed but no further transfer request will start.

(5) FLXAnFRITC.ITE

Input Transfer Enable Bit

The user should only set this bit to 1 when FLXAnFRIBCR.IBSYS is 0.

The user should only set this bit to 0 when FLXAnFRITC.IQHR 0. Otherwise committed input transfers get lost.

This bit controls the operation mode of the input transfer queue.

The operation status of the input transfer queue function is shown in FLXAnFRITS.ITS.

Refer to **Section 21.3.16.1 (1), Activation and deactivation** about usage of this bit.

0: Operation Disable request
The input transfer queue gets disabled when it becomes empty.

1: Operation Enable request
The input transfer queue gets enabled. Input data structures are transferred to the FlexRay internal message RAM.

21.2.12.2 FLXAnFROTC — FlexRay Output Transfer Configuration Register

Access: FLXAnFROTC can be read or written in 32-bit units.
FLXAnFROTCL and FLXAnFROTCH can be read or written in 16-bit units.
FLXAnFROTCLL, FLXAnFROTCLH, and FLXAnFROTCHL can be read or written in 8-bit units.

Address: FLXAnFROTC: <FLXAn_base> + 0804_H,
FLXAnFROTCL: <FLXAn_base> + 0804_H, FLXAnFROTCH: <FLXAn_base> + 0806_H
FLXAnFROTCLL: <FLXAn_base> + 0804_H, FLXAnFROTCLH: <FLXAn_base> + 0805_H,
FLXAnFROTCHL: <FLXAn_base> + 0806_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	FTM[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	FWIE	OWIE	FIE	OIE	—	—	—	—	—	—	OTCS	OTE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 21.85 FLXAnFROTC Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	FTM[4:0]	FIFO Table Max Bit Configures the number of FIFO entries the output transfer handler is capable to maintain in the Local RAM/Global RAM.
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	FWIE	FIFO transfer Warning Interrupt Enable Bit 0: Disabled 1: Enabled
10	OWIE	Output transfer Warning Interrupt Enable Bit 0: Disabled 1: Enabled
9	FIE	FIFO transfer Interrupt Enable Bit 0: Disabled 1: Enabled
8	OIE	Output transfer Interrupt Enable Bit 0: Disabled 1: Enabled
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OTCS	Output Transfer Condition Select Bit 0: New data only mode 1: New data and status changed mode
0	OTE	Output Transfer Enable Bit 0: Operation Disable request 1: Operation Enable request

(1) FLXAnFROTC.FTM

FIFO Table Max Bit

The user can only write to these bits when FLXAnFROTS.OTS is 0.

Configures the number of FIFO entries the output transfer handler is capable to maintain in the Local RAM/Global RAM.

Valid values are 00_H (1 FIFO entry) to $1F_H$ (32 FIFO entries).

(2) FLXAnFROTC.FWIE

FIFO transfer Warning Interrupt Enable Bit

This bit controls the FIFO transfer warning interrupt.

0: Disabled
No interrupt will be requested and the FIFO transfer warning interrupt line will be released.

1: Enabled
FIFO transfer warning interrupt will be asserted when FLXAnFROTS.FWIS is 1.

(3) FLXAnFROTC.OWIE

Output transfer Warning Interrupt Enable Bit

This bit controls the output transfer warning interrupt.

0: Disabled
No interrupt will be requested and the output transfer warning interrupt line will be released.

1: Enabled
Output transfer warning interrupt will be asserted when FLXAnFROTS.OWIS is 1.

(4) FLXAnFROTC.FIE

FIFO transfer Interrupt Enable Bit

This bit controls the FIFO transfer interrupt.

0: Disabled
No interrupt will be requested and the FIFO transfer interrupt line will be released.

1: Enabled
FIFO transfer interrupt will be asserted when FLXAnFROTS.FIS is 1.

(5) FLXAnFROTC.OIE

Output transfer Interrupt Enable Bit

This bit controls the output transfer interrupt.

0: Disabled
No interrupt will be requested and the output transfer interrupt line will be released.

1: Enabled
Output transfer interrupt will be asserted when FLXAnFROTS.OTIS is 1.

(6) FLXAnFROTC.OTCS

Output Transfer Condition Select Bit

The user can only write to this bit when FLXAnFROTS.OTS is 0.

This bit controls the output transfer condition.

0: New data only mode
The ND bits in the FLXAnFRNDATi registers are used to detect a transfer condition for dedicated receive buffer

1: New data and status changed mode
The ND bits in the FLXAnFRNDATi registers and the MBC bits in the FLXAnFRMBSC register are used to detect a transfer condition for dedicated transmit and receive buffer

(7) FLXAnFROTC.OTE

Output Transfer Enable Bit

The user should only set this bit to 1 when FLXAnFROBCR.OBSYS is 0.

This bit controls the operation mode of the output transfer function.

The operation status of the output buffer transfer function is shown in FLXAnFROTS.OTS.

Refer to **Section 21.3.16.2 (2), Output transfer data structure** about usage of this bit.

0: Operation Disable request
The output buffer transfer gets disabled.
An active message buffer transfer will be completed but no further transfer will start.

1: Operation Enable request
The output buffer transfer gets enabled. Message buffers are transferred from the FlexRay internal message RAM to output data structures.

The user cannot change the E-Ray message RAM configuration by writing to the FLXAnFRMRC register.

21.2.12.3 FLXAnFRIBA — FlexRay Input Pointer Table Base Address Register

Access: FLXAnFRIBA can be read or written in 32-bit units.
FLXAnFRIBAL and FLXAnFRIBAH can be read or written in 16-bit units.
FLXAnFRIBALL, FLXAnFRIBALH, FLXAnFRIBAHL, and FLXAnFRIBAHH can be read or written in 8-bit units.

Address: FLXAnFRIBA: <FLXAn_base> + 0808_H,
FLXAnFRIBAL: <FLXAn_base> + 0808_H, FLXAnFRIBAH: <FLXAn_base> + $080A_H$,
FLXAnFRIBALL: <FLXAn_base> + 0808_H, FLXAnFRIBALH: <FLXAn_base> + 0809_H,
FLXAnFRIBAHL: <FLXAn_base> + $080A_H$, FLXAnFRIBAHH: <FLXAn_base> + $080B_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ITA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 21.86 FLXAnFRIBA Register Contents

Bit Position	Bit Name	Function
31 to 0	ITA[31:0]	Input Table Address Bit These bits configure the base address of the input pointer table.

(1) FLXAnFRIBA.ITA

Input Table Address Bit

The user can only write to this bit when FLXAnFRITS.ITS is 0.

The address should be 32 bit aligned, thus the bits FLXAnFRIBA.ITA[1:0] are always 0.

These bits configure the base address of the input pointer table.

The table is used for the input transfer queue transferring message buffers from the Local RAM/Global RAM into the FlexRay internal message RAM.

The size of the input queue is configured in FLXAnFRITC.ITM.

Note that each entry requires two long words in the input pointer table.

21.2.12.4 FLXAnFRFBA — FlexRay FIFO Pointer Table Base Address Register

Access: FLXAnFRFBA can be read or written in 32-bit units.
FLXAnFRFBAL and FLXAnFRFBAH can be read or written in 16-bit units.
FLXAnFRFBALL, FLXAnFRFBALH, FLXAnFRFBAHL, and FLXAnFRFBAHH can be read or written in 8-bit units.

Address: FLXAnFRFBA: <FLXAn_base> + $080C_H$,
FLXAnFRFBAL: <FLXAn_base> + $080C_H$, FLXAnFRFBAH: <FLXAn_base> + $080E_H$,
FLXAnFRFBALL: <FLXAn_base> + $080C_H$, FLXAnFRFBALH: <FLXAn_base> + $080D_H$,
FLXAnFRFBAHL: <FLXAn_base> + $080E_H$, FLXAnFRFBAHH: <FLXAn_base> + $080F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 21.87 FLXAnFRFBA Register Contents

Bit Position	Bit Name	Function
31 to 0	FTA[31:0]	FIFO pointer Table Address Bit These bits configure the base address of the FIFO pointer table.

(1) FLXAnFRFBA.FTA

FIFO pointer Table Address Bit

The user can only write to this bit when FLXAnFROTS.OTS is 0.

The address should be 32 bit aligned, thus the bits FLXAnFRFBA.FTA[1:0] are always 0.

These bits configure the base address of the FIFO pointer table.

The table is used for message buffers transferred from the FlexRay internal FIFO to the Local RAM/ Global RAM.

The size of the FIFO is configured in FLXAnFROTC.FTM.

21.2.12.5 FLXAnFROBA — FlexRay Output Pointer Table Base Address Register

Access: FLXAnFROBA can be read or written in 32-bit units.
FLXAnFROBAL and FLXAnFROBAH can be read or written in 16-bit units.
FLXAnFROBALL, FLXAnFROBALH, FLXAnFROBAHL, and FLXAnFROBAHH can be read or written in 8-bit units.

Address: FLXAnFROBA: <FLXAn_base> + 0810_H,
FLXAnFROBAL: <FLXAn_base> + 0810_H, FLXAnFROBAH: <FLXAn_base> + 0812_H,
FLXAnFROBALL: <FLXAn_base> + 0810_H, FLXAnFROBALH: <FLXAn_base> + 0811_H,
FLXAnFROBAHL: <FLXAn_base> + 0812_H, FLXAnFROBAHH: <FLXAn_base> + 0813_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 21.88 FLXAnFROBA Register Contents

Bit Position	Bit Name	Function
31 to 0	OTA[31:0]	Output pointer Table Address Bit These bits configure the base address of the output pointer table.

(1) FLXAnFROBA.OTA

Output pointer Table Address Bit

The user can only write to this bit when FLXAnFROTS.OTS is 0.

The address should be 32 bit aligned, thus the bits FLXAnFROBA.OTA[1:0] are always 0.

These bits configure the base address of the output pointer table.

The table is used for message buffers transferred from the FlexRay internal message RAM to the Local RAM/Global RAM.

The size of the table depends on the utilization of the FlexRay internal message RAM and can have up to 128 entries.

21.2.12.6 FLXAnFRIQC — FlexRay Input Queue Control Register

Access: FLXAnFRIQC is a write-only register that can be written in 32-bit units.
FLXAnFRIQCL is a write-only register that can be written in 16-bit units.
FLXAnFRIQCLL is a write-only register that can be written in 8-bit units.

Address: FLXAnFRIQC: <FLXAn_base> + 0814_H,
FLXAnFRIQCL: <FLXAn_base> + 0814_H,
FLXAnFRIQCLL: <FLXAn_base> + 0814_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	IMBNR[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W

Table 21.89 FLXAnFRIQC Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When writing, write the value after reset.
6 to 0	IMBNR[6:0]	Input Message Buffer NumbeR Bit Message buffer number added to the input queue

(1) FLXAnFRIQC.IMBNR

Input Message Buffer NumbeR Bit

The user can only write to this bit when FLXAnFRITS.IQFP is 0.

The user cannot write to this register when FLXAnFRITS.ITS is 0 or when FLXAnFRITC.ITE is 0.

These bits are read as 0.

This value specifies the message buffer added to the input queue.

The number has to be identical to FLXAnFRWRHS4.IMBNR (see **Section 21.3.16.1 (3), Input pointer table**) of the input pointer table.

The address to the input data structure has to be provided in the input pointer table at the put index (FLXAnFRITS.IPIDX) before writing to this register.

Writing to this register increments the input put index (FLXAnFRITS.IPIDX).

21.2.12.7 FLXAnFRUIR — FlexRay User Input Transfer Request Register

Access: FLXAnFRUIR can be read or written in 32-bit units.
FLXAnFRUIRL can be read or written in 16-bit units.
FLXAnFRUIRLL can be read or written in 8-bit units.

Address: FLXAnFRUIR: <FLXAn_base> + 0818_H,
FLXAnFRUIRL: <FLXAn_base> + 0818_H,
FLXAnFRUIRLL: <FLXAn_base> + 0818_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	UIDX[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.90 FLXAnFRUIR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	UIDX[7:0]	User requested Input index Bit Input pointer table index requested for input transfer

(1) FLXAnFRUIR.UIDX

User requested Input index Bit

The user can only write to this bit when FLXAnFRITS.UIRP is 0.

The user should not write to this register when FLXAnFRITS.ITS is 0.

The user should not write to this register when FLXAnFRITS.UIRP is 1.

The user should not write to this register when FLXAnFRITS.IQH is 1.

The user should only write FLXAnFRITC.ITM+1 to this register.

This value specifies the input pointer table index for the user requested input transfer.

The address to the input data structure has to be provided in the input pointer table at the index UIDX before writing to this register.

When writing to this register, the requested input data structure will be transferred from input data structure position to the FlexRay internal message RAM.

In opposite to queued input transfers the related DA flag in the FLXAnFRDA register is not influenced by the user input transfer.

21.2.12.8 FLXAnFRUOR — FlexRay User Output Transfer Request Register

Access: FLXAnFRUOR can be read or written in 32-bit units.
FLXAnFRUORL can be read or written in 16-bit units.
FLXAnFRUORLL and FLXAnFRUORLH can be read or written in 8-bit units.

Address: FLXAnFRUOR: <FLXAn_base> + $081C_H$,
FLXAnFRUORL: <FLXAn_base> + $081C_H$,
FLXAnFRUORLL: <FLXAn_base> + $081C_H$, FLXAnFRUORLH: <FLXAn_base> + $081D_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	URDS	—	—	UMBNR[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.91 FLXAnFRUOR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	URDS	User request Read Data Section Bit 0: Data section is not transferred 1: Data section is transferred
8, 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	UMBNR[6:0]	User requested output Message Buffer NumbeR Bit Message buffer number requested for output transfer

(1) FLXAnFRUOR.URDS

User request Read Data Section Bit

The user can only write to this bit when FLXAnFROTS.UORP is 0.

The user should not write to this register when FLXAnFROTS.OTS is 0.

The user should not write to this register when FLXAnFROTS.UORP is 1.

0: Data section is not transferred
The data section of the message buffer selected by the bits UMBNR is not requested

1: Data section is transferred
The data section of the message buffer selected by the bits UMBNR is requested

(2) FLXAnFRUOR.UMBNR

User requested output Message Buffer NumbeR Bit

The user can only write to this bit when FLXAnFROTS.UORP is 0.

The user should not write to this register when FLXAnFROTS.OTS is 0.

The user should not write to this register when FLXAnFROTS.UORP is 1.

The user should restrict UMBNR to dedicated receive and transmit buffers when the FlexRay module is not in the CONFIG state.

When writing to this register, the header sections and optionally the data section (configurable by URDS) of the requested message buffer will be transferred from the FlexRay internal message RAM to the output data structure position defined by the output structure data pointer in the output pointer table.

21.2.12.9 FLXAnFRAHBC — FlexRay H-Bus Configuration Register

Access: FLXAnFRAHBC can be read or written in 32-bit units.
FLXAnFRAHBCL can be read or written in 16-bit units.
FLXAnFRAHBCLL can be read or written in 8-bit units.

Address: FLXAnFRAHBC: <FLXAn_base> + 0840_H,
FLXAnFRAHBCL: <FLXAn_base> + 0840_H,
FLXAnFRAHBCLL: <FLXAn_base> + 0840_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	HPROT[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 21.92 FLXAnFRAHBC Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	HPROT[3:0]	Protection Control Bit HPROT

(1) FLXAnFRAHBC.HPROT

Protection Control Bit

The user can only write to this register when FLXAnFRITS.ITS is 0 and when FLXAnFROTS.OTS is 0.

This register configures the value assigned to the H-Bus protection control signal.

21.2.13 Data Transfer Status Register

21.2.13.1 FLXAnFRITS — FlexRay Input Transfer Status Register

Access: FLXAnFRITS can be read or written in 32-bit units.
FLXAnFRITSL can be read or written in 16-bit units.
FLXAnFRITSH is a read-only register that can be read in 16-bit units.
FLXAnFRITSLL, FLXAnFRITSHL, and FLXAnFRITSHH are the read-only registers that can be read in 8-bit units.
FLXAnFRITSLH can be read or written in 8-bit units.

Address: FLXAnFRITS: <FLXAn_base> + 0820_H,
FLXAnFRITSL: <FLXAn_base> + 0820_H, FLXAnFRITSH: <FLXAn_base> + 0822_H,
FLXAnFRITSLL: <FLXAn_base> + 0820_H, FLXAnFRITSLH: <FLXAn_base> + 0821_H,
FLXAnFRITSHL: <FLXAn_base> + 0822_H, FLXAnFRITSHH: <FLXAn_base> + 0823_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	IGIDX[6:0]							—	IPIDX[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	IQFP	—	—	IQEIS	IQFIS	—	—	—	—	—	UIRP	IQH	ITS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 21.93 FLXAnFRITS Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30 to 24	IGIDX[6:0]	Input queue Get Index Bit Represents the get index of the input pointer table
23	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
22 to 16	IPIDX[6:0]	Input queue Put Index Bit Represents the put index of the input pointer table
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	IQFP	Input Queue Full condition Pending Bit 0: Entries in the input queue are available 1: All entries in the input queue are occupied
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	IQEIS	Input Queue Empty Interrupt Status Bit 0: No input queue empty condition detected 1: Input queue empty condition detected
8	IQFIS	Input Queue Full Interrupt Status Bit 0: No input queue full condition detected 1: Input queue full condition detected
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	UIRP	User Input transfer Request Pending Bit 0: No user input transfer request pending 1: User input transfer request pending

Table 21.93 FLXAnFRITS Register Contents (2/2)

Bit Position	Bit Name	Function
1	IQH	Input Queue Halted Bit 0: Input queue not halted 1: Input queue halted
0	ITS	Input Transfer Status Bit 0: Disabled 1: Enabled

(1) FLXAnFRITS.IGIDX

Input queue Get Index Bit

These bits are only valid when FLXAnFRITS.IQH is 1

These bits indicate the input pointer index the input queue handler will transfer next.

Valid values are 00_H to FLXAnFRITC.ITM.

The get index is incremented when the input data structure has been transferred from the Local RAM/ Global RAM and the related DA flag in the FLXAnFRDA register is cleared.

The index is set to 00_H when FLXAnFRITS.ITS changes from 0 to 1.

(2) FLXAnFRITS.IPIDX

Input queue Put Index Bit

These bits indicate the index where the next input data structure pointer in the input pointer table should be stored.

Valid values are 00_H to FLXAnFRITC.ITM.

After reaching the maximum value the put index continues from 00_H.

The index is incremented when writing to FLXAnFRIQC.IMBNR.

The index is set to 00_H when FLXAnFRITS.ITS changes from 0 to 1.

(3) FLXAnFRITS.IQFP

Input Queue Full condition Pending Bit

This bit indicates that the input queue is full.

There should be no further input transfer requests, by writing to FLXAnFRIQC.IMBNR, as long as FLXAnFRITS.IQFP is 1.

[Clearing condition]

This bit is cleared when there is one free entry in the input queue.

[Setting condition]

This bit is set when all entries in the input queue are occupied.

(4) FLXAnFRITS.IQEIS

Input Queue Empty Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in FLXAnFRITC.IQEIE the input queue empty interrupt is generated when FLXAnFRITS.IQEIS is 1.

[Clearing condition]

This bit is cleared when writing a 1 to FLXAnFRITS.IQEIS.
This bit is cleared when FLXAnFRITS.ITS changes from 0 to 1.

[Setting condition]

This bit is set when all pending input transfers have been processed and consequently the input queue becomes empty.

(5) FLXAnFRITS.IQFIS

Input Queue Full Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in FLXAnFRITC.IQFIE the input queue full interrupt is generated when FLXAnFRITS.IQFIS is 1.

This flag is intended as interrupt status flag. It does not indicate the current input queue status; for this status refer to FLXAnFRITS.IQFP.

[Clearing condition]

This bit is cleared when writing a 1 to FLXAnFRITS.IQFIS.

This bit is cleared when FLXAnFRITS.ITS changes from 0 to 1.

[Setting condition]

This bit is set when all entries in the input queue are occupied.

(6) FLXAnFRITS.UIRP

User Input transfer Request Pending Bit

This bit indicates that a user input transfer is still pending.

There should be no further write access to FLXAnFRUIR.UIDX when this bit is 1.

[Clearing condition]

This bit is cleared when the user input transfer request is processed by the input transfer handler.

[Setting condition]

This bit is set when writing to FLXAnFRUIR.UIDX.

(7) FLXAnFRITS.IQH

Input Queue Halted Bit

This bit indicates the status of the input queue.

There should be no further write access to FLXAnFRUIR.UIDX when this bit is 1.

[Clearing condition]

This bit is cleared when FLXAnFRITC.IQHR is set to 0.

[Setting condition]

This bit is set immediately when the FLXAnFRITC.IQHR is set to 1 and there is no ongoing input transfer.
This bit is set only after an ongoing input transfer has been completed and FLXAnFRITC.IQHR is set to 1.

(8) FLXAnFRITS.ITS

Input Transfer Status Bit

This bit indicates the status of the input queue handler.

While this bit is 1, there can be no read or write access to the address area <FLXAn_base> + 0400_H to <FLXAn_base> + $05FF_H$ and there should be no CLEAR_RAMS command applied to FLXAnFRSUCC1.CMD register.

The input transfer queue indices and related status flags are set to 0 when FLXAnFRITS.ITS changes from 0 to 1.

[Clearing condition]

This bit is cleared immediately when FLXAnFRITC.ITE is set to 0 and there are no pending input transfers

This bit is cleared after all pending requests have been processed and FLXAnFRITC.ITE is 0.

[Setting condition]

This bit is set when FLXAnFRITC.ITE is set to 1.

21.2.13.2 FLXAnFROTS — FlexRay Output Transfer Status Register

Access: FLXAnFROTS can be read or written in 32-bit units.
FLXAnFROTSL can be read or written in 16-bit units.
FLXAnFROTSH is a read-only register that can be read in 16-bit units.
FLXAnFROTSLL, FLXAnFROTSHL, and FLXAnFROTSHH are the read-only registers that can be read in 8-bit units.
FLXAnFROTSLH can be read or written in 8-bit units.

Address: FLXAnFROTS: <FLXAn_base> + 0824_H,
FLXAnFROTSL: <FLXAn_base> + 0824_H, FLXAnFROTSH: <FLXAn_base> + 0826_H,
FLXAnFROTSLL: <FLXAn_base> + 0824_H, FLXAnFROTSLH: <FLXAn_base> + 0825_H,
FLXAnFROTSHL: <FLXAn_base> + 0826_H, FLXAnFROTSHH: <FLXAn_base> + 0827_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	FFL[5:0]						—	—	—	FGIDX[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FWP	OWP	FDA	—	FWS	OWS	FIS	OTIS	—	—	—	—	—	UORP	—	OTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 21.94 FLXAnFROTS Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
29 to 24	FFL[5:0]	FIFO Fill Level Bit Represent the number of unprocessed output FIFO structures
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	FGIDX[4:0]	FIFO Get Index Bit Represent the get index in the FIFO pointer table
15	FWP	FIFO transfer Warning condition Pending Bit 0: No FIFO transfer warning condition pending 1: FIFO transfer warning condition pending
14	OWP	Output transfer Warning condition Pending 0: No output transfer warning condition pending 1: Output transfer warning condition pending
13	FDA	FIFO Data Available Bit 0: No available FIFO structures 1: FIFO structures available at current FLXAnFROTS.FGIDX index
12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	FWIS	FIFO transfer Warning Interrupt Status Bit 0: No FIFO transfer warning condition detected 1: FIFO transfer warning condition detected
10	OWIS	Output transfer Warning Interrupt Status Bit 0: No output transfer warning condition detected 1: Output transfer warning condition detected
9	FIS	FIFO transfer Interrupt Status Bit 0: No FIFO structure updated in Local RAM/Global RAM 1: FIFO structure updated in Local RAM/Global RAM

Table 21.94 FLXAnFROTS Register Contents (2/2)

Bit Position	Bit Name	Function
8	OTIS	Output transfer Interrupt Status Bit 0: No output structure updated in Local RAM/Global RAM 1: Output structure updated in Local RAM/Global RAM
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	UORP	User Output transfer Request Pending Bit 0: No user output transfer request pending 1: User output transfer request pending
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	OTS	Output Transfer Status Bit 0: Disabled 1: Enabled

(1) FLXAnFROTS.FFL

FIFO Fill Level Bit

These bits indicate the number of available output FIFO structures in the Local RAM/Global RAM.

Valid values are 00_H to FLXAnFROTC.FTM+1.

The value 00_H indicates that the FIFO is empty.

The value FLXAnFROTC.FTM+1 indicates that the FIFO is full and no further FIFO transfers will be done.

The FIFO fill level is incremented when a FIFO data structure has been transferred from the FlexRay internal FIFO into the Local RAM/Global RAM.

The FIFO fill level is decremented when the user releases a FIFO data structure in the Local RAM/Global RAM by writing 1 to FLXAnFROTS.FDA.

The FIFO fill level is set to 00_H when the bit FLXAnFROTS.OTS changes from 0 to 1.

(2) FLXAnFROTS.FGIDX

FIFO Get Index Bit

These bits indicate the index where the current output data structure pointer in the FIFO pointer table is available for reading.

Valid values are 00_H to FLXAnFROTC.FTM.

After reaching the maximum value the get index continues from 00_H.

The index is incremented when a FIFO data structure is released by writing 1 to FLXAnFROTS.FDA.

The index is set to 00_H when FLXAnFROTS.OTS changes from 0 to 1.

(3) FLXAnFROTS.FWP

FIFO transfer Warning condition Pending Bit

This bit indicates the FIFO transfer warning condition.

[Clearing condition]

This bit is cleared when there are free output data structures (FLXAnFROTS.FFL ≤ FLXAnFROTC.FTM).

This bit is cleared when the FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

This bit is set when the output transfer handler detects a transfer condition for FIFO message buffers but there are no free output data structures (FLXAnFROTS.FFL = FLXAnFROTC.FTM+1).

(4) FLXAnFROTS.OWP

Output transfer Warning condition Pending Bit

This bit indicates the output transfer warning condition.

[Clearing condition]

This bit is cleared, when all output structure pointers that have a pending output handler transfer condition detected are released (for dedicated transmit and receive message buffers or a user output transfer request).

This bit is cleared when the FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

This bit is set when the output transfer handler detects a transfer condition (for dedicated transmit and receive message buffers or a user output transfer request) but the related output structure pointer was not yet released by the application (data available flag is still set to 1).

This bit is set when the output transfer handler detects a transfer condition for dedicated transmit and receive message buffers but there is a pending input transfer for the same message buffer (data available flag is set to 1 due to the input transfer request).

(5) FLXAnFROTS.FDA

FIFO Data Available Bit

Writing 0 has no effect on the bit value.

When this bit is 1, the next valid output data structure is available.

The related data structure pointer is in the FIFO pointer table at FLXAnFROTS.FGIDX.

Writing 1 to FLXAnFROTS.FDA

- increments FLXAnFROTS.FGIDX and
- decrements the FIFO fill level (FLXAnFROTS.FFL)

If there are still unprocessed data structures FLXAnFROTS.FDA remains 1.

[Clearing condition]

This bit is cleared when writing 1 to FLXAnFROTS.FDA and the FIFO fill level becomes 00_H.

This bit is cleared when the FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

This bit is set when there is at least one FIFO data structure available in the Local RAM/Global RAM.

(6) FLXAnFROTS.FWIS

FIFO transfer Warning Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in FLXAnFROTC.FWIE the FIFO transfer warning interrupt is generated when FLXAnFROTS.FWIS is 1.

[Clearing condition]

This bit is cleared when writing a 1 to FLXAnFROTS.FWIS.

This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

This bit is set when the output transfer handler detects a transfer condition for FIFO message buffers but there are no free output data structures (FLXAnFROTS.FFL = FLXAnFROTC.FTM+1).

(7) FLXAnFROTS.OWIS

Output transfer Warning Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in FLXAnFROTC.OWIE the FIFO transfer warning interrupt is generated when FLXAnFROTS.OWIS is 1.

[Clearing condition]

This bit is cleared when writing a 1 to FLXAnFROTS.OWIS.

This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

This bit is set when the output transfer handler detects a transfer condition (for dedicated transmit and receive message buffers or a user output transfer request) but the related output structure pointer was not yet released by the application (data available flag is still set to 1).

This bit is set when the output transfer handler detects a transfer condition for dedicated transmit and receive message buffers but there is a pending input transfer for the same message buffer (data available flag is set to 1 due to the input transfer request).

(8) FLXAnFROTS.FIS

FIFO transfer Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in FLXAnFROTC.FIE the FIFO transfer interrupt is generated when FLXAnFROTS.FIS is 1.

[Clearing condition]

This bit is cleared when writing a 1 to FLXAnFROTS.FIS.

This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

This bit is set when a FIFO data structure is updated by the transfer handler or the FFL bit changes from 00_H to 01_H.

(9) FLXAnFROTS.OTIS

Output transfer Interrupt Status Bit

Writing 0 has no effect on the bit value.

If enabled in FLXAnFROTC.OIE the output transfer interrupt is generated when FLXAnFROTS.OTIS is 1.

[Clearing condition]

This bit is cleared when writing a 1 to FLXAnFROTS.OTIS.
This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

This bit is set when an output data structure is updated by the transfer handler (from a dedicated transmit or receive message buffer or by a user output transfer request).

(10) FLXAnFROTS.UORP

User Output transfer Request Pending Bit

This bit indicates that a user output transfer is still pending.

There should be no further write access to FLXAnFRUOR.UMBNR when this bit is 1.

[Clearing condition]

This bit is cleared when the user output transfer request is processed by the output transfer handler.

This bit is cleared when the bit FLXAnFROTS.OTS changes from 0 to 1.

[Setting condition]

This bit is set when writing to FLXAnFRUOR.UMBNR.

(11) FLXAnFROTS.OTS

Output Transfer Status Bit

This bit indicates the status of the output transfer handler.

While this bit is 1, there should be no read or write access to the address area <FLXAn_base> + 0600_H to <FLXAn_base> + $07FF_H$ and there should be no CLEAR_RAMS command applied to FLXAnFRSUCC1.CMD register.

While this bit is 1, the user cannot change the E-Ray message RAM configuration by writing to the FLXAnFRMRC register.

The output hander transfer indices and related status flags are set to 0 when FLXAnFROTS.OTS changes from 0 to 1.

[Clearing condition]

This bit is cleared immediately when FLXAnFROTC.OTE is set to 0 and there are no ongoing output transfers.

This bit is cleared after an ongoing transfer has been completed and FLXAnFROTC.OTE is 0.

[Setting condition]

This bit is set when FLXAnFROTC.OTE is set to 1.

21.2.13.3 FLXAnFRAES — FlexRay Access Error Status Register

Access: FLXAnFRAES can be read or written in 32-bit units.
FLXAnFRAESL can be read or written in 16-bit units.
FLXAnFRAESLL is a read-only register that can be read in 8-bit units.
FLXAnFRAESLH can be read or written in 8-bit units.

Address: FLXAnFRAES: <FLXAn_base> + 0828_H,
FLXAnFRAESL: <FLXAn_base> + 0828_H,
FLXAnFRAESLL: <FLXAn_base> + 0828_H, FLXAnFRAESLH: <FLXAn_base> + 0829_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	MAE	FAE	OAE	IAE	EIDX[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 21.95 FLXAnFRAES Register Contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11	MAE	Multiple Access Errors Bit 0: No multiple access errors occurred 1: Multiple access errors occurred
10	FAE	FIFO transfer Access Error Bit 0: No access error occurred during FIFO transfer 1: Access error occurred during FIFO transfer
9	OAE	Output transfer Access Error Bit 0: No access error occurred during output transfer 1: Access error occurred during output transfer
8	IAE	Input transfer Access Error Bit 0: No access error occurred during input transfer 1: Access error occurred during input transfer
7 to 0	EIDX[7:0]	Error Index Bit Data structure pointer index number

(1) FLXAnFRAES.MAE

Multiple Access Errors Bit

Writing 0 has no effect on the bit value.

This bit indicates that there were multiple access errors during a data transfer.

[Clearing condition]

This bit is cleared when writing a 1 to FLXAnFRAES.MAE.

[Setting condition]

This bit is set when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE are set and

- an access to an protected address occurred during a FIFO data transfer or
- an access to an protected address occurred during an output data transfer or
- an access to an protected address occurred during an input data transfer

(2) FLXAnFRAES.FAE

FIFO transfer Access Error Bit

Writing 0 has no effect on the bit value.

This bit indicates that there was an access error during a FIFO data transfer.

[Clearing condition]

This bit is cleared when writing a 1 to FLXAnFRAES.FAE.

[Setting condition]

This bit is set when a Local RAM/Global RAM access error was detected during a FIFO transfer and the bits FLXAnFRAES.OAE, FLXAnFRAES.IAE and FLXAnFRAES.MAE are 0.

(3) FLXAnFRAES.OAE

Output transfer Access Error Bit

Writing 0 has no effect on the bit value.

This bit indicates that there was an access error during a output data transfer.

[Clearing condition]

This bit is cleared when writing a 1 to FLXAnFRAES.OAE.

[Setting condition]

This bit is set when a Local RAM/Global RAM access error was detected during an output transfer and the bits FLXAnFRAES.FAE, FLXAnFRAES.IAE and FLXAnFRAES.MAE are 0.

(4) FLXAnFRAES.IAE

Input transfer Access Error Bit

Writing 0 has no effect on the bit value.

This bit indicates that there was an access error during an input data transfer.

[Clearing condition]

This bit is cleared when writing a 1 to FLXAnFRAES.IAE.

[Setting condition]

This bit is set when a Local RAM/Global RAM access error was detected during an output transfer and the bits FLXAnFRAES.OAE, FLXAnFRAES.FAE and FLXAnFRAES.MAE are 0.

(5) FLXAnFRAES.EIDX

Error Index Bit

This value is only valid when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE is 1.

When the bit FLXAnFRAES.FAE is 1, FLXAnFRAES.EIDX holds the used FIFO put index when the access error has occurred.

When the bit FLXAnFRAES.OAE is 1, FLXAnFRAES.EIDX holds the input pointer table get index used when an access error occurred during an input transfer or when the user request an input transfer.

When the bit FLXAnFRAES.IAE is 1, FLXAnFRAES.EIDX holds the used input pointer table get index when the access error has occurred.

These bits are updated when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE is changed from 0 to 1.

21.2.13.4 FLXAnFRAEA — FlexRay Access Error Address Register

Access: FLXAnFRAEA is a read-only register that can be read in 32-bit units.
FLXAnFRAEAL and FLXAnFRAEAH are the read-only registers that can be read in 16-bit units.
FLXAnFRAEALL, FLXAnFRAEALH, FLXAnFRAEAHL, and FLXAnFRAEAHH are the read-only registers that can be read in 8-bit units.

Address: FLXAnFRAEA: <FLXAn_base> + $082C_H$,
FLXAnFRAEAL: <FLXAn_base> + $082C_H$, FLXAnFRAEAH: <FLXAn_base> + $082E_H$,
FLXAnFRAEALL: <FLXAn_base> + $082C_H$, FLXAnFRAEALH: <FLXAn_base> + $082D_H$,
FLXAnFRAEAHL: <FLXAn_base> + $082E_H$, FLXAnFRAEAHH: <FLXAn_base> + $082F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AEA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AEA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21.96 FLXAnFRAEA Register Contents

Bit Position	Bit Name	Function
31 to 0	AEA[31:0]	Access Error Address Bit Address in the Local RAM/Global RAM when an access error has occurred

(1) FLXAnFRAEA.AEA

Access Error Address Bit

This value is only valid when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE is 1.

These bits indicate the address of the access error indicated in the FLXAnFRAES register.

These bits are updated when one of the bits FLXAnFRAES.FAE, FLXAnFRAES.OAE or FLXAnFRAES.IAE is changed from 0 to 1.

21.2.13.5 FLXAnFRDAi — FlexRay Message Data Available Register i (i = 0 to 3)

Access: FLXAnFRDAi can be read or written in 32-bit units.
FLXAnFRDAiL and FLXAnFRDAiH can be read or written in 16-bit units.
FLXAnFRDAiLL, FLXAnFRDAiLH, FLXAnFRDAiHL, and FLXAnFRDAiHH can be read or written in 8-bit units.

Address: FLXAnFRDAx: <FLXAn_base> + 0830_H + i × 4_H,
FLXAnFRDAxL: <FLXAn_base> + 0830_H + i × 4_H,
FLXAnFRDAxH: <FLXAn_base> + 0830_H + i × 4_H + 2_H,
FLXAnFRDAxLL: <FLXAn_base> + 0830_H + i × 4_H,
FLXAnFRDAxLH: <FLXAn_base> + 0830_H + i × 4_H + 1_H,
FLXAnFRDAxHL: <FLXAn_base> + 0830_H + i × 4_H + 2_H,
FLXAnFRDAxHH: <FLXAn_base> + 0830_H + i × 4_H + 3_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DAb (b = i ×32 + 31 to i × 32 + 16)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DAb (b = i ×32 + 15 to i × 32 + 0)															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.97 FLXAnFRDAi Register Contents

Bit Position	Bit Name	Function
31 to 0	DAb	Data Available Bit 0: No data available for destination 1: Data available for destination

(1) FLXAnFRDAi.DAb (b = i × 32 to (i+1) × 32-1)

Data Available Bit b

The user should not write a 1 to bits that are 0.

To maintain the status of input transfers, the user cannot clear bits related to input transfers.

This register is used for input and output transfers.

Each flag corresponds to a FlexRay message buffer.

[Clearing condition]

Input transfer:
This bit is cleared when the input data structure has been transferred from the Local RAM/Global RAM. The data structure and the data structure pointer can be changed when the related flag is 0.

Output transfer:
This bit is cleared by writing 1 to it.

[Setting condition]

Input transfer:
This bit is set when the corresponding message buffer number has been written to FLXAnFRIQC.IMBNR.

As long as this bit is 1, the input data structure and the data structure pointer corresponding to this input transfer request cannot be changed.

Output transfer:

This bit is set when the output data structure corresponding to this message buffer has been updated. As long as this bit is 1, the data structure is stable; no further update of the data structure will be done by the output handler. While this bit is 1, the application is allowed to change the output data structure pointer in the output pointer table for this message buffer number.

21.3 Functional Description

This chapter describes the FlexRay implementation together with the related FlexRay protocol features. More information about the FlexRay protocol itself can be found in the FlexRay protocol specification.

21.3.1 FlexRay Module Operation Control

21.3.1.1 FlexRay Module Enable

After hardware reset or after the FlexRay module has been disabled (following **Section 21.3.1.2, FlexRay Module Disable**) the FlexRay module is in the reset state (FLXAnFROS.OS is 0) and the clocks of the FlexRay core module are disabled.

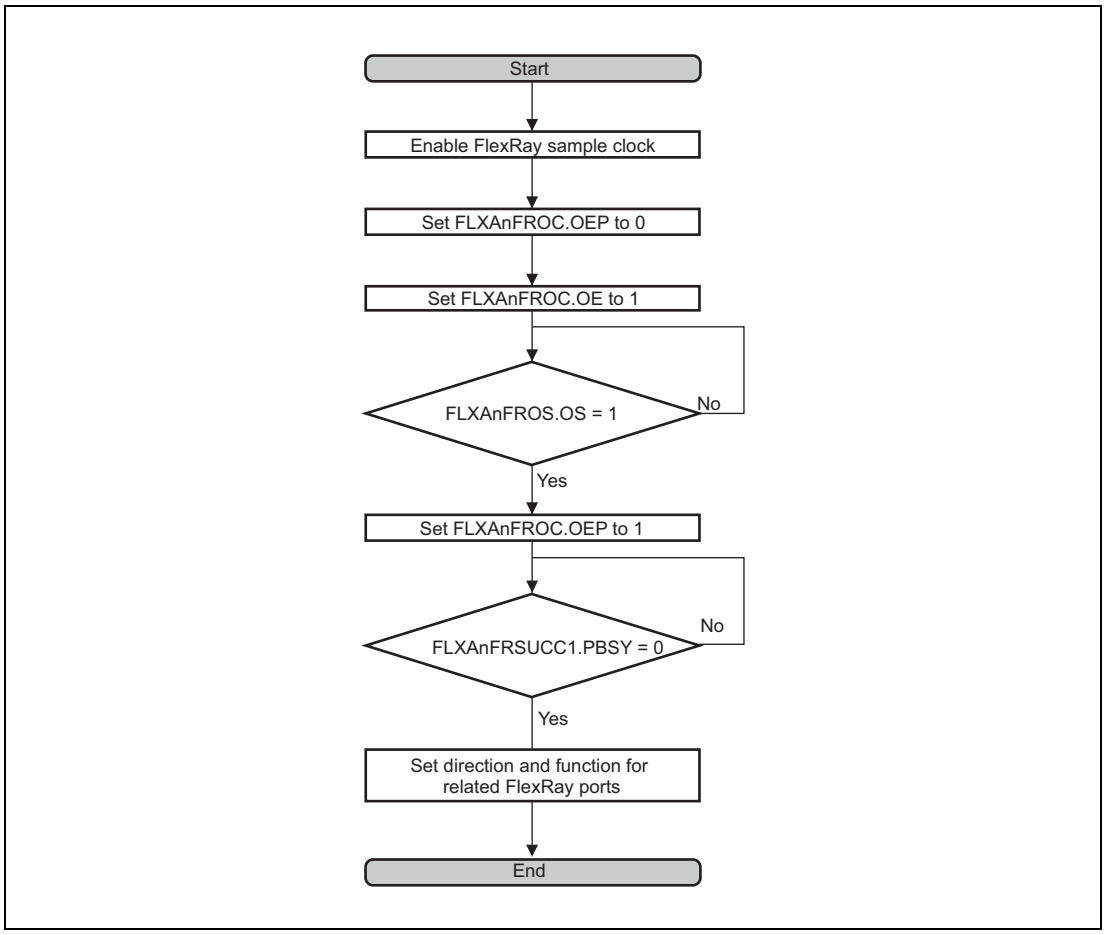


Figure 21.3 FlexRay enable flow

21.3.1.2 FlexRay Module Disable

The FlexRay module can be disabled at any time. However, it is recommended to disable the FlexRay module using the FLXAnFROC.OE register only when the FlexRay module is in HALT, CONFIG or DEFAULT_CONFIG state. Resetting the FlexRay module in any other state will terminate any ongoing FlexRay communication.

If the data transfer function is used, it is also required to disable this function before disabling the FlexRay module (see **Section 21.3.16.1 (1), Activation and deactivation** for suspending input transfer function and **Section 21.3.16.2 (2), Output transfer data structure** for suspending output transfer).

The following flow should be executed to disable the FlexRay module.

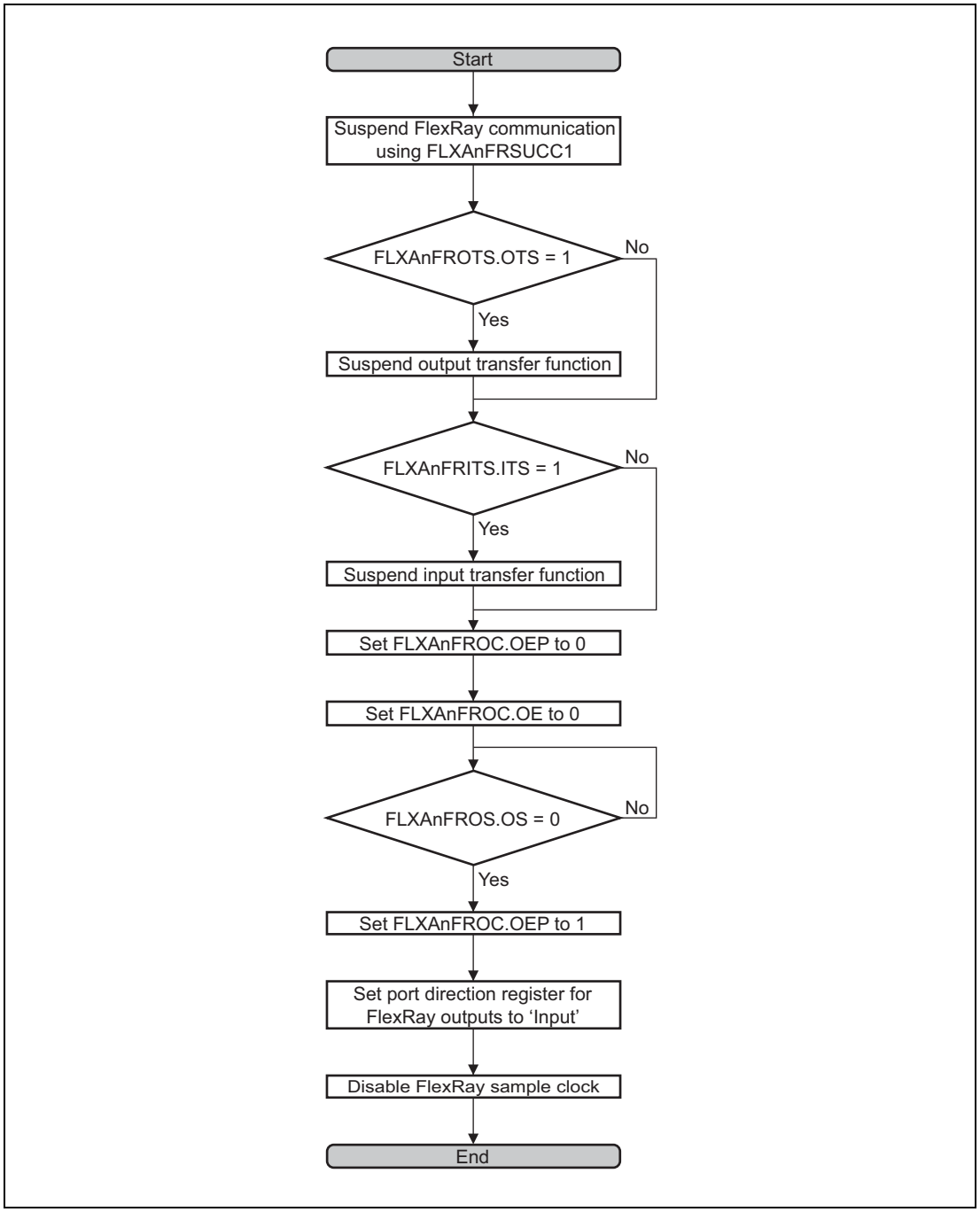


Figure 21.4 FlexRay disable flow

21.3.2 Communication Cycle

Communication on FlexRay networks is based on frames and symbols. The wakeup symbol (WUS) and the collision avoidance symbol (CAS) are transmitted outside the communication cycle to setup the time schedule. Frames and media access test symbols (MTS) are transmitted inside the communication cycle.

A FlexRay communication cycle consists of the following elements:

- Static Segment
- Dynamic Segment (optional)
- Symbol Window (optional)
- Network Idle Time (NIT)

Static segment, dynamic segment, and symbol window form the Network Communication Time (NCT). For each communication channel the slot counter starts at 1 and counts up until the end of the dynamic segment is reached. Both channels share the same arbitration grid which means that they use the same synchronized macrotick.

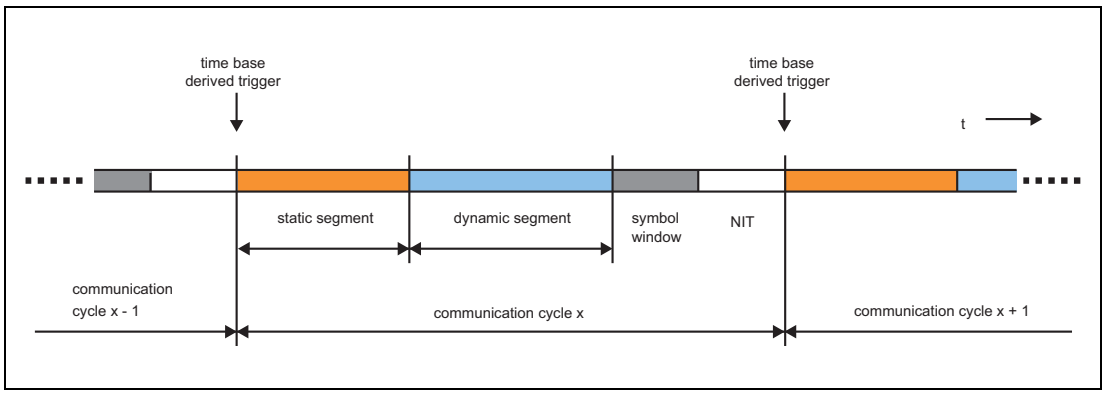


Figure 21.5 Structure of Communication Cycle

21.3.2.1 Static Segment

The Static Segment is characterized by the following features:

- Time slots of fixed length (optionally protected by bus guardian)
- Start of frame transmission at action point of the respective static slot
- Payload length same for all frames on both channels

Parameters:

Number of Static Slots (FLXAnFRGTUC7.NSS)

Static Slot Length (FLXAnFRGTUC7.SSL)

Payload Length Static (FLXAnFRMHDC.SFDL)

Action Point Offset (FLXAnFRGTUC9.APO)

21.3.2.2 Dynamic Segment

The Dynamic Segment is characterized by the following features:

- All controllers have bus access (no bus guardian protection possible)
- Variable payload length and duration of slots, different for both channels
- Start of transmission at minislot action point

Parameters:

Number of Minislots (FLXAnFRGTUC8.NMS)

Minislot Length (FLXAnFRGTUC8.MSL)

Minislot Action Point Offset (FLXAnFRGTUC9.MAPO)

Start of Latest Transmit (last minislot) (FLXAnFRMHDC.SLT)

21.3.2.3 Symbol Window

During the symbol window only one media access test symbol (MTS) may be transmitted per channel. MTS symbols are send in NORMAL_ACTIVE state to test the bus guardian.

The symbol window is characterized by the following features:

- Send single symbol
- Transmission of the MTS symbol starts at the symbol windows action point

Parameters:

Symbol Window Action Point Offset (FLXAnFRGTUC9.APO) (same as for static slots)

Network Idle Time Start Position (FLXAnFRGTUC4.NIT)

21.3.2.4 Network Idle Time (NIT)

During network idle time the CC performs the following tasks:

- Calculate clock correction terms (offset and rate)
- Distribute offset correction over multiple macroticks after offset correction start
- Perform cluster cycle related tasks

Parameters:

Network idle time start position configuration bit (FLXAnFRGTUC4.NIT)

Offset correction start position configuration bit (FLXAnFRGTUC4.OCS)

21.3.2.5 Configuration of NIT Start and Offset Correction Start

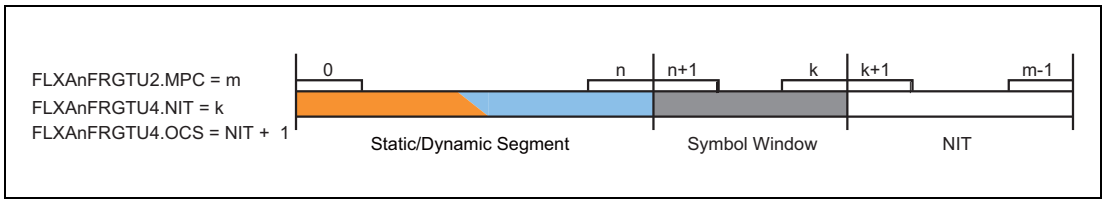


Figure 21.6 Configuration of NIT Start and Offset Correction Start

The number of macroticks per cycle gMacroPerCycle is assumed to be m. It is configured by programming FLXAnFRGTUC2.MPC= m.

The static / dynamic segment starts with macrotick 0 and ends with macrotick n:

n = static segment length + dynamic segment offset + dynamic segment length - 1MT

n = gNumberOfStaticSlots o gdStaticSlot + dynamic segment offset + gNumberOfMinislots o gdMinislot - 1 MT

The static segment length is configured by FLXAnFRGTUC7.SSL and FLXAnFRGTUC7.NSS.

The dynamic segment length is configured by FLXAnFRGTUC8.MSL and FLXAnFRGTUC8.NMS.

The dynamic segment offset is:

If gdActionPointOffset ≤ gdMinislotActionPointOffset:

dynamic segment offset = 0 MT

Else if gdActionPointOffset > gdMinislotActionPointOffset:

dynamic segment offset = gdActionPointOffset -gdMinislotActionPointOffset

The NIT starts with macrotick k+1 and ends with the last macrotick of cycle m-1. It has to be configured by setting FLXAnFRGTUC4.NIT = k.

For the FlexRay module the offset correction start is required to be FLXAnFRGTUC4.OCS ≥ FLXAnFRGTUC4.NIT + 1 = k+1.

The length of symbol window results from the number of macroticks between the end of the static / dynamic segment and the beginning of the NIT. It can be calculated by the number of macroticks (k - n).

21.3.3 Communication Modes

The FlexRay Protocol Specification defines the Time-Triggered Distributed (TT-D) mode.

21.3.3.1 Time-triggered Distributed (TT-D)

In TT-D mode the following configurations are possible:

- Pure static: Minimum 2 static slots + symbol window (optional)
- Mixed static/dynamic: Minimum 2 static slots + dynamic segment + symbol window (optional)

A minimum of two coldstart nodes needs to be configured for distributed time-triggered operation. Two fault-free coldstart nodes are necessary for the cluster startup. Each startup frame must be a sync frame, therefore all coldstart nodes are sync nodes.

21.3.4 Clock Synchronization

In TT-D mode, distributed clock synchronization is used. Each node individually synchronizes itself to the cluster by observing the timing of received sync frames from other nodes.

21.3.4.1 Global Time

Activities in a FlexRay node, including communication, are based on the concept of a global time. It is the clock synchronization mechanism that differentiates the FlexRay cluster from other node collections with independent clock mechanisms. The global time is a vector of two values: the cycle (cycle counter) and the cycle time (macrotick counter).

Cluster definitions:

- Macrotick (MT) = basic unit of time measurement in a FlexRay network, a macrotick consists of an integer number of microticks (µT)
- Cycle length = duration of a communication cycle in units of macroticks (MT)

21.3.4.2 Local Time

Internally, nodes time their behavior with microtick resolution. Microticks are time units derived from the oscillator clock tick of the specific node. Therefore microticks are controller-specific units. They may have different duration in different controllers. The precision of a node's local time difference measurements is a microtick (µT).

Node definitions:

- Oscillator clock → prescaler → microtick (µT)
- µT = basic unit of time measurement in a CC, clock correction is done in units of → µT
- Cycle counter + macrotick counter = nodes local view of the global time

21.3.4.3 Synchronization Process

Clock synchronization is performed by means of sync frames. Only preconfigured nodes (sync nodes) are allowed to send sync frames. In a two-channel cluster a sync node has to send its sync frame on both channels.

For synchronization in FlexRay the following constraints have to be considered:

- Max. one sync frame per node in one communication cycle
- Max. 15 sync frames per cluster in one communication cycle
- Every node has to use a preconfigured number of sync frames (FLXAnFRGTUC2.SNM) for clock synchronization
- Minimum of two sync nodes required for clock synchronization and startup

For clock synchronization the time difference between expected and observed arrival time of sync frames received during the static segment is measured. In a two channel cluster the sync node has to be configured to send sync frames on both channels. The calculation of correction terms is done during NIT (offset: every cycle, rate: every odd cycle) by using an FTM algorithm. For details see FlexRay protocol specification v2.1, chapter 8.

(1) Offset (phase) Correction

- Only deviation values measured and stored in the current cycle used
- For a two channel node the smaller value will be taken
- Calculation during NIT of every communication cycle
- Offset correction value calculated in even cycles used for error checking only
- Checked against limit values
- Correction value is a signed integer number of µTs
- Correction done in odd numbered cycles, distributed over the macroticks beginning at offset correction start up to cycle end (end of NIT) to shift nodes next start of cycle (MTs lengthened / shortened)

(2) Rate (frequency) Correction

- Pairs of deviation values measured and stored in even / odd cycle pair used
- For a two channel node the average of the differences from the two channels is used
- Calculated during NIT of odd numbered cycles
- Cluster drift damping is performed using global damping value
- Checked against limit values
- Correction value is a signed integer number of µTs
- Distributed over macroticks comprising the next even / odd cycle pair (MTs lengthened / shortened)

(3) Sync Frame Transmission

Sync frame transmission is only possible from buffer 0 and 1. Message buffer 1 may be used for sync frame transmission in case that sync frames should have different payloads on the two channels. In this case bit FLXAnFRMRC.SPLM has to be programmed to 1.

Message buffers used for sync frame transmission have to be configured with the key slot ID and can be (re)configured in DEFAULT_CONFIG or CONFIG state only. For nodes transmitting sync frames FLXAnFRSUCC1.TXSY must be set to 1.

(4) External Clock Synchronization

During normal operation, independent clusters can drift significantly. If synchronous operation across independent clusters is desired, external synchronization is necessary; even though the nodes within each cluster are synchronized. This can be accomplished with synchronous application of host-deduced rate and offset correction terms to the clusters.

- External offset / rate correction value is a signed integer
- External offset / rate correction value is added to calculated offset / rate correction value
- Aggregated offset / rate correction term (external + internal) is not checked against configured limits

21.3.5 Error Handling

The implemented error handling concept is intended to ensure that, in case of a lower layer protocol error in one single node, communication between non-affected nodes can be maintained. In some cases, higher layer program activity is required for the CC to resume normal operation. A change of the error handling state will set FLXAnFREIR.PEMC to 1 and may trigger an interrupt to the Host if enabled. The actual error mode is signaled by FLXAnFRCCEV.ERRM.

Table 21.98 Error modes of the POC (degradation model)

Error Mode	Activity
ACTIVE	Full operation, State: NORMAL_ACTIVE The CC is fully synchronized and supports the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR.
PASSIVE	Reduced operation, State: NORMAL_PASSIVE, CC self rescue allowed The CC stops transmitting frames and symbols, but received frames are still processed. Clock synchronization mechanisms are continued based on received frames. No active contribution to the cluster wide clock synchronization. The host is informed of any error condition(s) or status change by interrupt (if enabled) or by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR.
COMM_HALT	Operation halted, State: HALT, CC self rescue not allowed The CC stops frame and symbol processing, clock synchronization processing, and the macrotick generation. The host has still access to error and status information by reading the error and status interrupt flags from registers FLXAnFREIR and FLXAnFRSIR. The bus drivers are disabled.

21.3.5.1 Clock Correction Failed Counter

When the Clock Correction Failed Counter reaches the “maximum without clock correction passive” limit defined by FLXAnFRSUCC3.WCP, the POC transits from NORMAL_ACTIVE to NORMAL_PASSIVE state. When it reaches the “maximum without clock correction fatal” limit defined by FLXAnFRSUCC3.WCF, it transits from NORMAL_ACTIVE or NORMAL_PASSIVE to HALT state.

The Clock Correction Failed Counter FLXAnFRCCEV.CCFC allows the Host to monitor the duration of the inability of a node to compute clock correction terms after the CC passed protocol startup phase. It will be incremented by one at the end of any odd communication cycle during which either the missing offset correction FLXAnFRSFS.MOCS or the missing rate correction FLXAnFRSFS.MRCS flag is set to 1.

The Clock Correction Failed Counter is reset to zero at the end of an odd communication cycle if neither the missing offset correction FLXAnFRSFS.MOCS nor the missing rate correction FLXAnFRSFS.MRCS flag is set to 1.

The Clock Correction Failed Counter stops incrementing when the “maximum without clock correction fatal” value FLXAnFRSUCC3.WCF is reached (i.e. incrementing the counter at its maximum value will not cause it to wrap around back to zero). The Clock Correction Failed Counter is initialized to zero when the CC enters READY state or when NORMAL_ACTIVE state is entered.

CAUTION

The transition to HALT state is prevented if FLXAnFRSUCC1.HCSE is not set to 1.

21.3.5.2 Passive to Active Counter

The passive to active counter controls the transition of the POC from NORMAL_PASSIVE to NORMAL_ACTIVE state. FLXAnFRSUCC1.PTA defines the number of consecutive even / odd cycle pairs that must have valid clock correction terms before the CC is allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state. If FLXAnFRSUCC1.PTA is set to zero the CC is not allowed to transit from NORMAL_PASSIVE to NORMAL_ACTIVE state.

21.3.5.3 HALT Command

If the Host wants to stop FlexRay communication of the local node it can bring the CC into HALT state by asserting the HALT command. This can be done by writing FLXAnFRSUCC1.CMD = "0110_B". In order to shut down communication on an entire FlexRay network, a higher layer protocol is required to confirm that all nodes apply the HALT command at the same time.

The POC state from which the transition to HALT state took place can be read from FLXAnFRCCSV.PSL.

When called in NORMAL_ACTIVE or NORMAL_PASSIVE state the POC transits to HALT state at the end of the current cycle. When called in any other state FLXAnFRSUCC1.CMD will be reset to "0000_B" = command_not_accepted and bit FLXAnFREIR.CNA is set to 1. If enabled an interrupt to the Host is generated.

21.3.5.4 FREEZE Command

In case the Host detects a severe error condition it can bring the CC into HALT state by asserting the FREEZE command. This can be done by writing FLXAnFRSUCC1.CMD = "0111_B". The FREEZE command triggers the entry of the HALT state immediately regardless of the actual POC state.

The POC state from which the transition to HALT state took place can be read from FLXAnFRCCSV.PSL.

CAUTION

When the communication is stopped by the FREEZE or READY command and the communication is restarted as Leading ColdStart, the startup frame is not transmitted on cycle0 depending on the internal condition of the FlexRay module. This case occurs when the Startup frame is set in one of the slot 1 to slot 7.

This does not occur in ColdStart after a hardware reset.

Even if this occurs, the second trial of ColdStart will succeed. ColdStart time becomes longer, but ColdStart will not be affected by the occurrence.

To avoid this, allocate the Startup/Sync frame in the static slot 8 or higher.

21.3.6 Communication Controller States

21.3.6.1 Communication Controller State Diagram

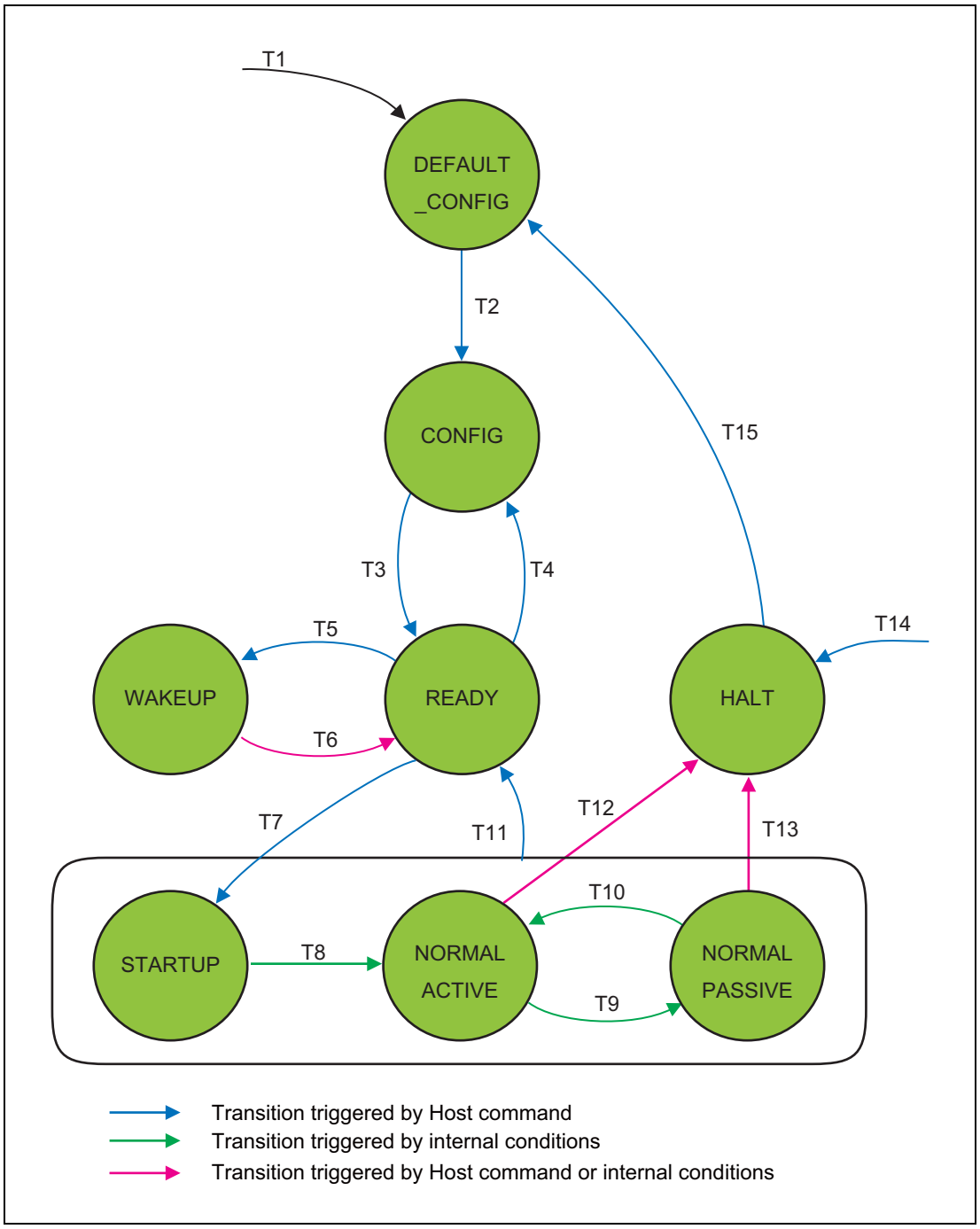


Figure 21.7 Overall State Diagram of FlexRay Communication Controller

State transitions are controlled by reset, FLXAnFR0RXDA, FLXAnFR0RXDB, by the POC state machine, and by the CHI Command Vector FLXAnFRSUCC1.CMD.

The CC transits from all states to HALT state after application of the FREEZE command (FLXAnFRSUCC1.CMD = "0111_B").

Table 21.99 State Transitions of FlexRay Overall State Machine

T#	Condition	From	To
1	Reset	All States	DEFAULT_CONFIG
2	Command CONFIG, FLXAnFRSUCC1.CMD = “0001_B”	DEFAULT_CONFIG	CONFIG
3	Unlock sequence followed by command READY, FLXAnFRSUCC1.CMD = “0010_B”	CONFIG	READY
4	Command CONFIG, FLXAnFRSUCC1.CMD = “0001_B”	READY	CONFIG
5	Command WAKEUP, FLXAnFRSUCC1.CMD = “0011_B”	READY	WAKEUP
6	Complete transmission of wakeup pattern or received WUP or received frame header or wakeup collision detection or command READY, FLXAnFRSUCC1.CMD = “0010_B”	WAKEUP	READY
7	Command RUN, FLXAnFRSUCC1.CMD = “0100_B”	READY	STARTUP
8	Successful STARTUP	STARTUP	NORMAL_ACTIVE
9	Clock Correction Failed counter reached Maximum Without Clock Correction Passive limit configured by FLXAnFRSUCC3.WCP	NORMAL_ACTIVE	NORMAL_PASSIVE
10	Number of valid correction terms reached the Passive to Active limit configured by FLXAnFRSUCC1.PTA	NORMAL_PASSIVE	NORMAL_ACTIVE
11	Command READY, FLXAnFRSUCC1.CMD = “0010_B”	STARTUP, NORMAL_ACTIVE, NORMAL_PASSIVE	READY
12	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by FLXAnFRSUCC3.WCF when bit FLXAnFRSUCC1.HCSE set to 1 or command HALT, FLXAnFRSUCC1.CMD = “0110_B”	NORMAL_ACTIVE	HALT
13	Clock Correction Failed counter reached Maximum Without Clock Correction Fatal limit configured by FLXAnFRSUCC3.WCF when bit FLXAnFRSUCC1.HCSE set to 1 or command HALT, FLXAnFRSUCC1.CMD = “0110_B”	NORMAL_PASSIVE	HALT
14	Command FREEZE, FLXAnFRSUCC1.CMD = “0111_B”	All States	HALT
15	Command CONFIG, FLXAnFRSUCC1.CMD = “0001_B”	HALT	DEFAULT_CONFIG

21.3.6.2 DEFAULT_CONFIG State

In DEFAULT_CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state.

The CC enters this state

- When the reset is applied (HW reset or SW reset)
- When exiting from HALT state

To leave DEFAULT_CONFIG state the Host has to write FLXAnFRSUCC1.CMD = “0001_B”. The CC then transits to CONFIG state.

21.3.6.3 CONFIG State

In CONFIG state, the CC is stopped. All configuration registers are accessible and the pins to the physical layer are in their inactive state. This state is used to initialize the CC configuration.

The CC enters this state

- When exiting from DEFAULT_CONFIG state
- When exiting from READY state

When the state has been entered via HALT and DEFAULT_CONFIG state, the Host can analyze status information and configuration. Before leaving CONFIG state the Host has to confirm that the configuration is fault-free.

To leave CONFIG state, the Host has to perform the unlock sequence as described is **Section 21.2.3.3, FLXAnFRLCK — FlexRay Lock Register**. Directly after unlocking the CONFIG state the Host has to write FLXAnFRSUCC1.CMD to enter the next state.

CAUTION

Status bits FLXAnFRMHDS[14:0], registers FLXAnFRTXRQ1/2/3/4, and status data stored in the Message RAM are not affected by the transition of the POC from CONFIG to READY state.

When the CC is in CONFIG state it is also possible to bring the CC into a power saving mode by halting the module clocks (bus clock and sample clock). To do this the Host has to confirm that all Message RAM transfers have finished before turning off the clocks.

21.3.6.4 READY State

After unlocking CONFIG state and writing FLXAnFRSUCC1.CMD = "0010_B", the CC enters READY state. From this state the CC can transit to WAKEUP state and perform a cluster wakeup or to STARTUP state to perform a coldstart or to integrate into a running cluster.

The CC enters this state

- When exiting from CONFIG, WAKEUP, STARTUP, NORMAL_ACTIVE, or NORMAL_PASSIVE state by writing FLXAnFRSUCC1.CMD = "0010_B" (READY command).

The CC exits from this state

- To CONFIG state by writing FLXAnFRSUCC1.CMD = "0001_B" (CONFIG command)
- To WAKEUP state by writing FLXAnFRSUCC1.CMD = "0011_B" (WAKEUP command)
- To STARTUP state by writing FLXAnFRSUCC1.CMD = "0100_B" (RUN command)

Internal counters and the CC status flags are reset when the CC enters STARTUP state.

CAUTION

Status bits FLXAnFRMHDS[14:0], registers FLXAnFRTXRQ1/2/3/4, and status data stored in the Message RAM are not affected by the transition of the POC from READY to STARTUP state.

21.3.6.5 WAKEUP State

The description below is intended to help configuring wakeup for the FlexRay IP-module. A detailed description of the wakeup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.1.

The CC enters this state

- When exiting from READY state by writing FLXAnFRSUCC1.CMD = "0011_B" (WAKEUP command).

The CC exits from this state to READY state

- After complete non-aborted transmission of wakeup pattern (WUP)
- After WUP reception
- After detecting a WUP collision
- After reception of a frame header
- By writing FLXAnFRSUCC1.CMD = "0010_B" (READY command)

The cluster wakeup must precede the communication startup in order to ensure that all nodes in a cluster are awake. The minimum requirement for a cluster wakeup is that all bus drivers are supplied with power. A bus driver has the ability to wake up the other components of its node when it receives a wakeup pattern on its channel. At least one node in the cluster needs an external wakeup source.

The Host completely controls the wakeup procedure. It is informed about the state of the cluster by the bus driver and the CC and configures bus guardian (if available) and CC to perform the cluster wakeup. The CC provides to the Host the ability to transmit a special wakeup pattern on each of its available channels separately. The CC needs to recognize the wakeup pattern only during WAKEUP state.

Wakeup may be performed on only one channel at a time. The Host has to configure the wakeup channel while the CC is in CONFIG state by writing FLXAnFRSUCC1.WUCS. The CC ensures that ongoing communication on this channel is not disturbed. The CC cannot guarantee that all nodes connected to the configured channel awake upon the transmission of the wakeup pattern, since these nodes cannot give feedback until the startup phase. The wakeup procedure enables single-channel devices in a two-channel system to trigger the wakeup, by only transmitting the wakeup pattern on the single channel to which they are connected. Any coldstart node that deems a system startup necessary will then wake the remaining channel before initiating communication startup.

The wakeup procedure tolerates any number of nodes simultaneously trying to wakeup a single channel and resolves this situation such that only one node transmits the pattern. Additionally the wakeup pattern is collision resilient, so even in the presence of a fault causing two nodes to simultaneously transmit a wakeup pattern, the resulting collided signal can still wake the other nodes.

After wakeup the CC returns to READY state and signals the change of the wakeup status to the Host by setting flag FLXAnFRSIR.WST. The wakeup status vector can be read from FLXAnFRCCSV.WSV. If a valid wakeup pattern was received also either flag FLXAnFRSIR.WUPA or flag FLXAnFRSIR.WUPB is set to 1.

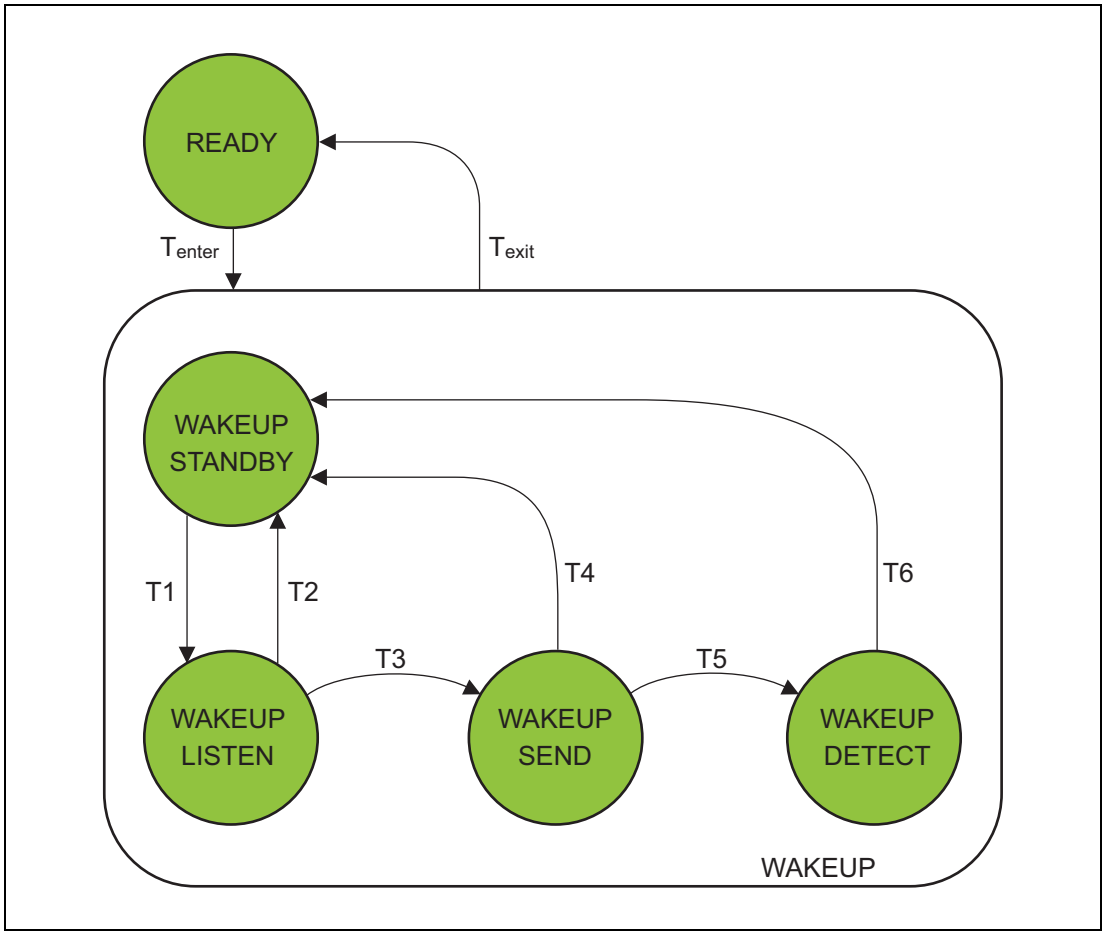


Figure 21.8 Structure of POC State WAKEUP

Table 21.100 State Transitions WAKEUP

T#	Condition	From	To
enter	Host commands change to WAKEUP state by writing FLXAnFRSUCC1.CMD = "0011_B" (WAKEUP command)	READY	WAKEUP
1	CHI command WAKEUP triggers wakeup FSM to transit to WAKEUP_LISTEN state	WAKEUP_STANDBY	WAKEUP_LISTEN
2	Received WUP on wakeup channel selected by bit FLXAnFRSUCC1.WUCS or frame header on either available channel	WAKEUP_LISTEN	WAKEUP_STANDBY
3	Timer event	WAKEUP_LISTEN	WAKEUP_SEND
4	Complete, non-aborted transmission of wakeup pattern	WAKEUP_SEND	WAKEUP_STANDBY
5	Collision detected	WAKEUP_SEND	WAKEUP_DETECT
6	Wakeup timer expired or WUP detected on wakeup channel selected by bit FLXAnFRSUCC1.WUCS or frame header received on either available	WAKEUP_DETECT	WAKEUP_STANDBY
exit	Wakeup completed (after T2 or T4 or T6) or Host commands change to READY state by writing FLXAnFRSUCC1.CMD = "0010_B" (READY command). This command also resets the wakeup FSM to WAKEUP_STANDBY state	WAKEUP	READY

The WAKEUP_LISTEN state is controlled by the wakeup timer and the wakeup noise timer. The two timers are controlled by the ListenTimeout (FLXAnFRSUCC2.LT bit) and ListenTimeoutNoise (FLXAnFRSUCC2.LTN bit) values. ListenTimeout enables a fast cluster wakeup in case of a noise free environment, while ListenTimeoutNoise enables wakeup under more difficult conditions regarding noise interference.

In WAKEUP_SEND state the CC transmits the wakeup pattern on the configured channel and checks for collisions. After return from wakeup the Host has to bring the CC into STARTUP state by CHI command RUN.

In WAKEUP_DETECT state the CC attempts to identify the reason for the wakeup collision detected in WAKEUP_SEND state. The monitoring is bounded by the expiration of listen timeout as configured by FLXAnFRSUCC2.LT. Either the detection of a wakeup pattern indicating a wakeup attempt by another node or the reception of a frame header indicating ongoing communication, causes the direct transition to READY state. Otherwise WAKEUP_DETECT is left after expiration of listen timeout; in this case the reason for wakeup collision is unknown.

The Host has to be aware of possible failures of the wakeup and act accordingly. It is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal time it takes another coldstart node to become awake and to be configured.

The FlexRay Protocol Specification recommends that two different CCs shall awake the two channels.

(1) Host activities

The host must coordinate the wakeup of the two channels and must decide whether, or not, to wake a specific channel. The sending of the wakeup pattern is initiated by the Host. The wakeup pattern is detected by the remote BDs and signaled to their local Host.

Wakeup procedure controlled by Host (single-channel wakeup):

- Configure the CC in CONFIG state
  - Select wakeup channel by programming FLXAnFRSUCC1.WUCS
- Check local BDs whether a WUP was received
- Activate BD of selected wakeup channel
- Command CC to enter READY state
- Command CC to start wakeup on the configured channel by writing FLXAnFRSUCC1.CMD = “0011_B”
  - CC enters WAKEUP
  - CC returns to READY state and signals status of wakeup attempt to the Host
- Wait predefined time to allow the other nodes to wakeup and configure themselves
- Coldstart node:
  - In a dual channel cluster wait for WUP on the other channel
  - Reset coldstart inhibit flag FLXAnFRCCSV.CSI by writing FLXAnFRSUCC1.CMD = “1001_B” (ALLOW_COLDSTART command)
- Command CC to enter startup by writing FLXAnFRSUCC1.CMD = “0100_B” (RUN command)

Wakeup procedure triggered by BD:

- Wakeup recognized by BD
- BD triggers power-up of Host (if required)
- BD signals wakeup event to Host
- Host configures its local CC
- If necessary, Host commands wakeup of second channel and waits predefined time to allow the other nodes to wakeup and configure themselves
- Host commands CC to enter STARTUP state by writing FLXAnFRSUCC1.CMD = "0100_B" (RUN command)

(2) Wakeup pattern (WUP)

The wakeup pattern (WUP) is composed of at least two wakeup symbols (WUS). Wakeup symbol and wakeup pattern are configured by registers FLXAnFRPRTC1 and FLXAnFRPRTC2.

- Single channel wakeup, wakeup symbol may not be sent on both channels at the same time
- Wakeup symbol collision resilient for at least two sending nodes (two overlapping wakeup symbols always recognizable)
- Wakeup symbol must be configured identical in all nodes of a cluster
- Wakeup symbol transmit low time configured by FLXAnFRPRTC2.TXL
- Wakeup symbol idle time used to listen for activity on the bus, configured by FLXAnFRPRTC2.TXI
- A wakeup pattern composed of at least two Tx-wakeup symbols needed for wakeup
- Number of repetitions configurable by FLXAnFRPRTC1.RWP (2 to 63 repetitions)
- Wakeup symbol receive window length configured by FLXAnFRPRTC1.RXW
- Wakeup symbol receive low time configured by FLXAnFRPRTC2.RXL
- Wakeup symbol receive idle time configured by FLXAnFRPRTC2.RXI

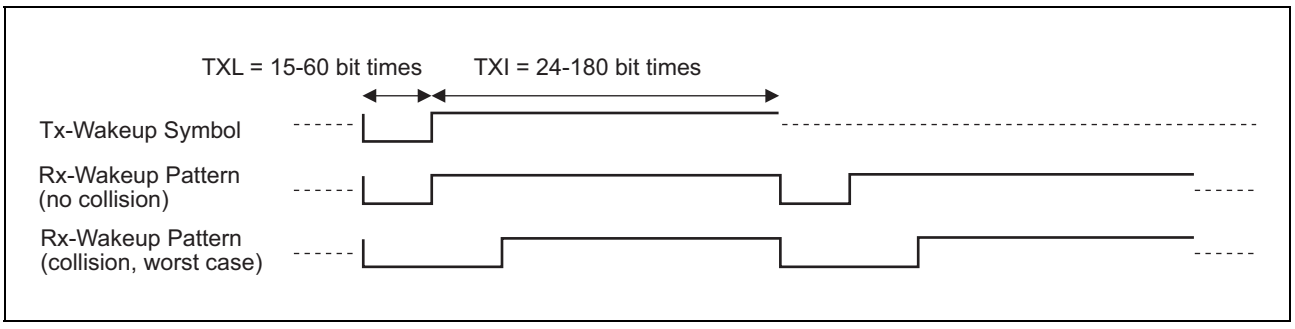


Figure 21.9 Timing of Wakeup Pattern

21.3.6.6 STARTUP State

The description below is intended to help configuring startup for the FlexRay module. A detailed description of the startup procedure together with the respective SDL diagrams can be found in the FlexRay protocol specification v2.1, section 7.2.

Any node entering STARTUP state that has coldstart capability should confirm that both channels attached have been awakened before initiating coldstart.

It cannot be assumed that all nodes and stars need the same amount of time to become completely awake and to be configured. Since at least two nodes are necessary to start up the cluster communication, it is advisable to delay any potential startup attempt of the node having instigated the wakeup by the minimal amount of time it takes another coldstart node to become awake, to be configured and to enter startup. It may require several hundred milliseconds (depending on the hardware used) before all nodes and stars are completely awakened and configured.

Startup is performed on all channels synchronously. During startup, a node only transmits startup frames. Startup frames are both sync frames and null frames during startup.

A fault-tolerant, distributed startup strategy is used to ensure that all nodes are synchronized initially. In general, a node may enter NORMAL_ACTIVE state via (see **Figure 21.10**):

- Coldstart path initiating the schedule synchronization (LeadingColdstart node)
- Coldstart path joining other coldstart nodes (FollowingColdstart node)
- Integration path integrating into an existing communication schedule (all other nodes)

A coldstart attempt begins with the transmission of a collision avoidance symbol (CAS). Only a coldstart node that had transmitted the CAS transmits frames in the first four cycles after the CAS, it is then joined firstly by the other coldstart nodes and afterwards by all other nodes.

A coldstart node has bits FLXAnFRSUCC1.TXST and FLXAnFRSUCC1.TXSY set to 1. Message buffer 0 holds the key slot ID which defines the slot number where the startup frame is send. In the frame header of the startup frame the startup frame indicator bit is set to 1.

In clusters consisting of three or more nodes, at least three nodes shall be configured to be coldstart nodes. In clusters consisting of two nodes, both nodes must be coldstart nodes. At least two fault-free coldstart nodes are necessary for the cluster to startup.

Each startup frame must also be a sync frame; therefore each coldstart node will also be a sync node. The number of coldstart attempts is configured by FLXAnFRSUCC1.CSA.

A non-coldstart node requires at least two startup frames from distinct nodes for integration. It may start integration before the coldstart nodes have finished their startup. It will not finish its startup until at least two coldstart nodes have finished their startup.

Both non-coldstart nodes and coldstart nodes start passive integration via the integration path as soon as they receive sync frames from which to derive the TDMA schedule information. During integration, the node has to adapt its own clock to the global clock (rate and offset) and has to make its cycle time consistent with the global schedule observable at the network. Afterwards, these settings are checked for consistency with all available network nodes. The node can only leave the integration phase and actively participate in communication when these checks are passed.

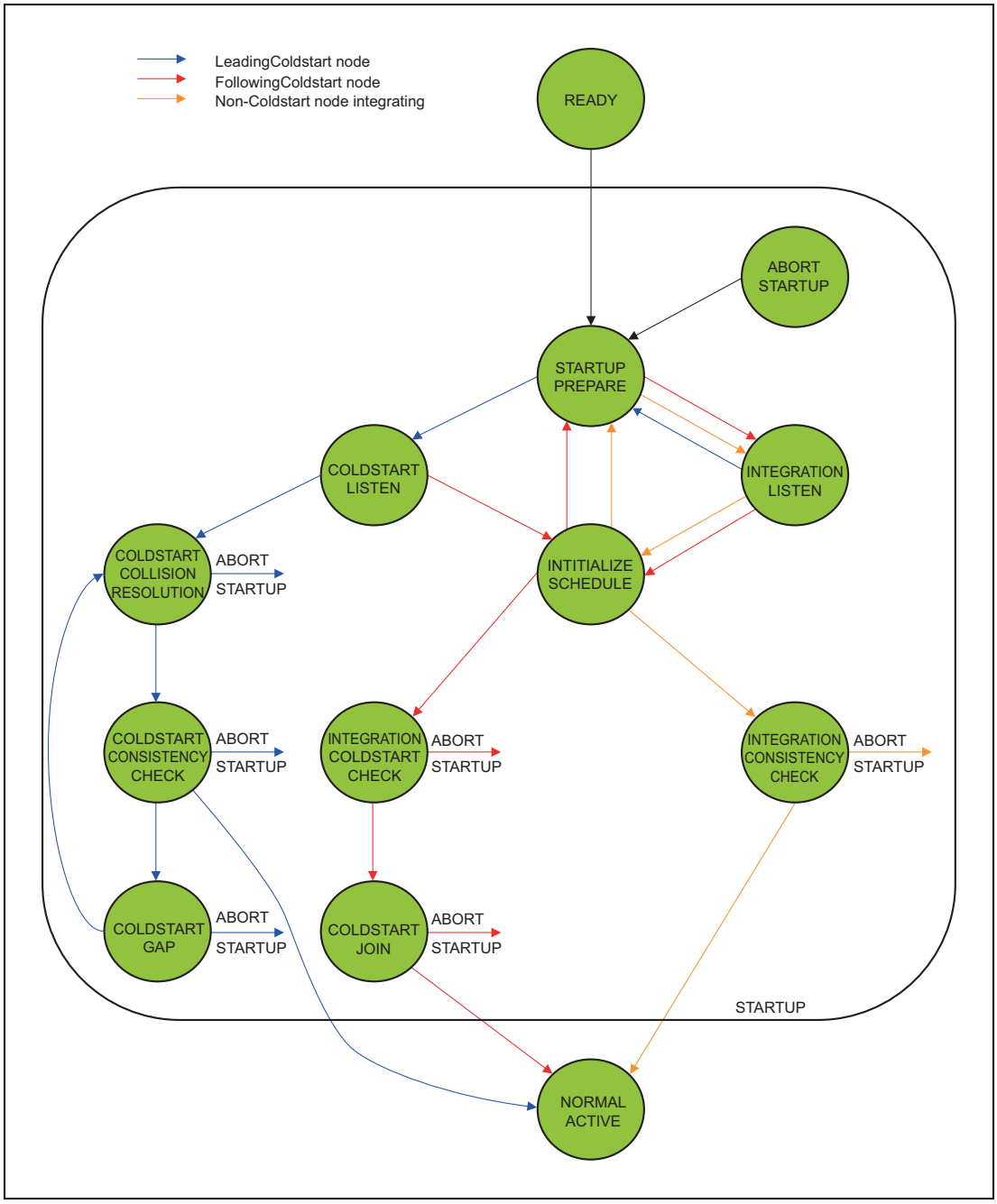


Figure 21.10 State Diagram Time-Triggered Startup

(1) Coldstart Inhibit Mode

In coldstart inhibit mode the node is prevented from initializing the TDMA communication schedule. If bit FLXAnFRCCSV.CSI is set to 1, the node is not allowed to initialize the cluster communication, i.e. entering the coldstart path is prohibited. The node is allowed to integrate to a running cluster or to transmit startup frames after another coldstart node started the initialization of the cluster communication.

The coldstart inhibit bit FLXAnFRCCSV.CSI is set to 1 whenever the POC enters READY state. The bit has to be cleared under control of the Host by CHI command ALLOW_COLDSTART (FLXAnFRSUCC1.CMD = "1001_B")

(2) Startup Timeouts

The CC supplies two different µT timers supporting two timeout values, startup timeout and startup noise timeout. The two timers are started when the CC enters the COLDSTART_LISTEN state. The expiration of either of these timers causes the node to leave the initial sensing phase (COLDSTART_LISTEN state) with the intention of starting up communication.

CAUTION

The startup and startup noise timers are identical with the wakeup and wakeup noise timers and use the same configuration values FLXAnFRSUCC2.LT and FLXAnFRSUCC2.LTN.

(a) Startup Timeout

The startup timeout limits the listen time used by a node to determine if there is already communication between other nodes or at least one coldstart node actively requesting the integration of others. The startup timer is configured by programming FLXAnFRSUCC2.LT (see **Section 21.2.6.2, FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2**).

The startup timeout is:

pdListenTimeout = FLXAnFRSUCC2.LT

The startup timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Both channels reaching idle state while in COLDSTART_LISTEN state

The startup timer is stopped:

- If communication channel activity is detected on one of the configured channels while the node is in the COLDSTART_LISTEN state
- When the COLDSTART_LISTEN state is left

Once the startup timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The timer status is kept for further processing by the startup state machine.

(b) Startup Noise Timeout

At the same time the startup timer is started for the first time (transition from STARTUP_PREPARE state to COLDSTART_LISTEN state), the startup noise timer is started. This additional timeout is used to improve reliability of the startup procedure in the presence of noise. The startup noise timeout is configured by programming FLXAnFRSUCC2.LTN (see **Section 21.2.6.2, FLXAnFRSUCC2 — FlexRay SUC Configuration Register 2**).

The startup noise timeout is:

pdListenTimeout × gListenNoise = FLXAnFRSUCC2.LT × (FLXAnFRSUCC2.LTN + 1)

The startup noise timer is restarted upon:

- Entering the COLDSTART_LISTEN state
- Reception of correctly decoded headers or CAS symbols while the node is in COLDSTART_LISTEN state

The startup noise timer is stopped when the COLDSTART_LISTEN state is left.

Once the startup noise timeout expires, neither an overflow nor a cyclic restart of the timer is performed. The status is kept for further processing by the startup state machine. Since the startup noise timer won't be restarted when random channel activity is sensed, this timeout defines the fall-back solution that guarantees that a node will try to start up the communication cluster even in the presence of noise.

(3) Path of Leading Coldstart Node (initiating coldstart)

When a coldstart node enters COLDSTART_LISTEN, it listens to its attached channels.

If no communication is detected, the node enters the COLDSTART_COLLISION_RESOLUTION state and commences a coldstart attempt. The initial transmission of a CAS symbol is succeeded by the first regular cycle. This cycle has the number zero.

From cycle zero on, the node transmits its startup frame. Since each coldstart node may perform a coldstart attempt, it may occur that several nodes simultaneously transmit the CAS symbol and enter the coldstart path. This situation is resolved during the first four cycles after CAS transmission.

As soon as a node that initiates a coldstart attempt receives a CAS symbol or a frame header during these four cycles, it re-enters the COLDSTART_LISTEN state. Thereby, only one node remains in this path. In cycle four, other coldstart nodes begin to transmit their startup frames.

After four cycles in COLDSTART_COLLISION_RESOLUTION state, the node that initiated the coldstart enters the COLDSTART_CONSISTENCY_CHECK state. It collects all startup frames from cycle four and five and performs the clock correction. If the clock correction does not deliver any errors and it has received at least one valid startup frame pair, the node leaves COLDSTART_CONSISTENCY_CHECK and enters NORMAL_ACTIVE state.

The number of coldstart attempts that a node is allowed to perform is configured by FLXAnFRSUCC1.CSA. The number of remaining coldstarts attempts can be read from FLXAnFRCCSV.RCA. The number of remaining coldstart attempts is reduced by one for each attempted coldstart. A node may enter the COLDSTART_LISTEN state only if this value is larger than one and it may enter the COLDSTART_COLLISION_RESOLUTION state only if this value is larger than zero. If the number of coldstart attempts is one, coldstart is inhibited but integration is still possible.

(4) Path of Following Coldstart Node (responding to Leading Coldstart Node)

When a coldstart node enters the COLDSTART_LISTEN state, it tries to receive a valid pair of startup frames to derive its schedule and clock correction from the leading coldstart node.

As soon as a valid startup frame has been received the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_COLDSTART_CHECK state is entered.

In INTEGRATION_COLDSTART_CHECK state it is confirmed that the clock correction can be performed correctly and that the coldstart node from which this node has initialized its schedule is still valid. The node collects all sync frames and performs clock correction in the following double-cycle. If clock correction does not signal any errors and if the node continues to receive sufficient frames from the same node it has integrated on, the COLDSTART_JOIN state is entered.

In COLDSTART_JOIN state following coldstart nodes begin to transmit their own startup frames and continue to do so in subsequent cycles. Thereby, the leading coldstart node and the nodes joining it can check if their schedules agree with each other. If the clock correction signals any error, the node aborts the integration attempt. If a node in this state sees at least one valid startup frame during all even cycles in this state and at least one valid startup frame pair during all double cycles in this state, the node leaves COLDSTART_JOIN state and enters NORMAL_ACTIVE state. Thereby it leaves STARTUP at least one cycle after the node that initiated the coldstart.

(5) Path of Non-coldstart Node

When a non-coldstart node enters the INTEGRATION_LISTEN state, it listens to its attached channels.

As soon as a valid startup frame has been received, the INITIALIZE_SCHEDULE state is entered. If the clock synchronization can successfully receive a matching second valid startup frame and derive a schedule from this, the INTEGRATION_CONSISTENCY_CHECK state is entered.

In INTEGRATION_CONSISTENCY_CHECK state the node verifies that the clock correction can be performed correctly and that enough coldstart nodes (at least 2) are sending startup frames that agree with the node's own schedule. Clock correction is activated, and if any errors are signaled, the integration attempt is aborted.

During the first even cycle in this state, either two valid startup frames or the startup frame of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

During the first double-cycle in this state, either two valid startup frame pairs or the startup frame pair of the node that this node has integrated on must be received; otherwise the node aborts the integration attempt.

If after the first double-cycle less than two valid startup frames are received within an even cycle, or less than two valid startup frame pairs are received within a double-cycle, the startup attempt is aborted.

Nodes in this state need to see two valid startup frame pairs for two consecutive double-cycles each to be allowed to leave STARTUP and enter NORMAL_OPERATION. Consequently, they leave startup at least one double-cycle after the node that initiated the coldstart and only at the end of a cycle with an odd cycle number.

21.3.6.7 NORMAL_ACTIVE State

As soon as the node that transmitted the first CAS symbol (resolving the potential access conflict and entering STARTUP via coldstart path) and one additional node have entered the NORMAL_ACTIVE state, the startup phase for the cluster has finished. In the NORMAL_ACTIVE state, all configured messages are scheduled for transmission. This includes all data frames as well as the sync frames. Rate and offset measurement is started in all even cycles (even / odd cycle pairs required).

In NORMAL_ACTIVE state the CC supports regular communication functions

- The CC performs transmissions and reception on the FlexRay bus as configured
- Clock synchronization is running
- The Host interface is operational

The CC exits from that state to

- HALT state by writing FLXAnFRSUCC1.CMD = "0110_B" (HALT command, at the end of the current cycle)
- HALT state by writing FLXAnFRSUCC1.CMD = "0111_B" (FREEZE command, immediately)
- HALT state due to change of the error state from ACTIVE to COMM_HALT
- NORMAL_PASSIVE state due to change of the error state from ACTIVE to PASSIVE
- READY state by writing FLXAnFRSUCC1.CMD = "0010_B" (READY command)

21.3.6.8 NORMAL_PASSIVE State

NORMAL_PASSIVE state is entered from NORMAL_ACTIVE state when the error state changes from ACTIVE to PASSIVE.

In NORMAL_PASSIVE state, the node is able to receive all frames (node is fully synchronized and performs clock synchronization). Contrary to the NORMAL_ACTIVE state, the node does not actively participate in communication, i.e. neither symbols nor frames are transmitted.

In NORMAL_PASSIVE state

- The CC performs reception on the FlexRay bus
- The CC does not transmit any frames or symbols on the FlexRay bus
- Clock synchronization is running
- The Host interface is operational

The CC exits from this state to

- HALT state by writing FLXAnFRSUCC1.CMD = "0110_B" (HALT command, at the end of the current cycle)
- HALT state by writing FLXAnFRSUCC1.CMD = "0111_B" (FREEZE command, immediately)
- HALT state due to change of the error state from PASSIVE to COMM_HALT
- NORMAL_ACTIVE state due to change of the error state from PASSIVE to ACTIVE. The transition takes place when FLXAnFRCCEV.PTAC equals FLXAnFRSUCC1.PTA - 1
- To READY state by writing FLXAnFRSUCC1.CMD = "0010_B" (READY command)

21.3.6.9 HALT State

In this state all communication (reception and transmission) is stopped.

The CC enters this state

- By writing FLXAnFRSUCC1.CMD = "0110_B" (HALT command) while the CC is in NORMAL_ACTIVE or NORMAL_PASSIVE state
- By writing FLXAnFRSUCC1.CMD = "0111_B" (FREEZE command) from all states
- When exiting from NORMAL_ACTIVE state because the clock correction failed counter reached the "maximum without clock correction fatal" limit and FLXAnFRSUCC1.HCSE is set to 1
- When exiting from NORMAL_PASSIVE state because the clock correction failed counter reached the "maximum without clock correction fatal" limit and FLXAnFRSUCC1.HCSE is set to 1

The CC exits from this state to DEFAULT_CONFIG state

- By writing FLXAnFRSUCC1.CMD = "0001_B" (CONFIG command)

When the CC enters HALT state, all configuration and status data is maintained for analyzing purposes.

When the Host writes FLXAnFRSUCC1.CMD = "0110_B" (HALT command), the CC sets bit FLXAnFRCCSV.HRQ to 1 and enters HALT state at the next end of cycle.

When the Host writes FLXAnFRSUCC1.CMD = "0111_B" (FREEZE command), the CC enters HALT state immediately and sets bit FLXAnFRCCSV.FSI to 1.

The POC state from which the transition to HALT state took place can be read from FLXAnFRCCSV.PSL.

21.3.7 Network Management

The accrued Network Management (NM) vector can be read from registers FLXAnFRNMV1 to FLXAnFRNMV3. The CC performs a bit-wise OR operation over all NM vectors out of all received valid NM frames with the Payload Preamble Indicator (PPI) bit set. Only static frames may be configured to hold NM information. The CC updates the NM vector at the end of each cycle.

The length of the NM vector can be configured from 0 to 12 bytes by FLXAnFRNEMC.NML. The NM vector length must be configured identically in all nodes of a cluster.

To configure a transmit buffer to send FlexRay frames with the PPI bit set as 1, bit PPIT in the header section of the respective transmit buffer has to be set to 1 via FLXAnFRWRHS1.PPIT. In addition the Host has to write the NM information to the data section of the respective transmit buffer.

The evaluation of the NM vector has to be done by the application running on the Host.

Section 21.3.17, Byte Alignment, for byte alignment of the received NM vector in registers FLXAnFRNMV1 to FLXAnFRNMV3.

CAUTIONS

1. **In case a message buffer is configured for transmission / reception of network management frames, the payload length configured in header 2 of that message buffer should be equal or greater than the length of the NM vector configured by FLXAnFRNEMC.NML.**
2. **When the CC transits to HALT state, the cycle count is not incremented and therefore the NM vector is not updated. In this case FLXAnFRNMV1 to FLXAnFRNMV3 holds the value from the cycle before.**

21.3.8 Filtering and Masking

Filtering is done by comparison of the configuration of assigned message buffers against actual slot and cycle counter values and channel ID (channel A, B). A message buffer is only updated / transmitted if the required matches occur.

Filtering is done on:

- Slot Counter
- Cycle Counter
- Channel ID

The following filter combinations for acceptance / transmit filtering are allowed:

- Slot Counter + Channel ID
- Slot Counter + Cycle Counter + Channel ID

All configured filters must match in order to store a received message in a message buffer.

CAUTION

For the FIFO the acceptance filter is configured by the FIFO Rejection Filter (FLXAnFRFRF) and the FIFO Rejection Filter Mask (FLXAnFRFRFM).

A message will be transmitted in the time slot corresponding to the configured frame ID on the configured channel(s). If cycle counter filtering is enabled the configured cycle filter value must also match.

21.3.8.1 Slot Counter Filtering

Every transmit and receive buffer contains a frame ID stored in the header section. This frame ID is compared against the actual slot counter value in order to assign receive and transmit buffers to the corresponding slot.

If two or more message buffers are configured with the same frame ID and channel ID, and if they have a matching cycle counter filter value for the same slot, then the message buffer with the lowest message buffer number is used.

21.3.8.2 Cycle Counter Filtering

Cycle counter filtering is based on the notion of a cycle set. For filtering purposes, a match is detected if any one of the elements of the cycle set is matched. The cycle set is defined by the cycle code field in header section 1 of each message buffer.

If message buffer 0 or 1 is configured to hold the startup / sync frame or the single slot frame by bits FLXAnFRSUCC1.TXST, FLXAnFRSUCC1.TXSY, and FLXAnFRSUCC1.TSM, cycle counter filtering for message buffer 0 or 1 shall be disabled.

CAUTION

Sharing of a static time slot via cycle counter filtering between different nodes of a FlexRay network is not allowed.

The set of cycle numbers belonging to a cycle set is determined as described in **Table 21.101**.

Table 21.101 Definition of cycle set

Cycle Code	Matching Cycle Counter Values
0b000000x	all Cycles
0b000001c	every second Cycle at (Cycle Count) mod2 = c
0b00001cc	every fourth Cycle at (Cycle Count) mod4 = cc
0b0001ccc	every eighth Cycle at (Cycle Count) mod8 = ccc
0b001cccc	every sixteenth Cycle at (Cycle Count) mod16 = cccc
0b01ccccc	every thirty-second Cycle at (Cycle Count) mod32 = ccccc
0b1cccccc	every sixty-fourth Cycle at (Cycle Count) mod64 = cccccc

Table 21.102 below gives some examples for valid cycle sets to be used for cycle counter filtering.

Table 21.102 Examples for valid cycle sets

Cycle Code	Matching Cycle Counter Values
0b0000011	1-3-5-7-…. -63
0b0000100	0-4-8-12-…. -60
0b0001110	6-14-22-30-…. -62
0b0011000	8-24-40-56
0b0100011	3-35
0b1001001	9

The received message is stored only if the cycle counter value of the cycle during which the message is received matches an element of the receive buffer's cycle set. Channel ID and frame ID must also be met.

The content of a transmit buffer is transmitted on the configured channel(s) when an element of the cycle set matches the current cycle counter value. Channel ID and frame ID must also match.

21.3.8.3 Channel ID Filtering

There is a 2-bit channel filtering field (CH) located in the header section of each message buffer in the Message RAM. It serves as a filter for receive buffers, and as a control field for transmit buffers (see **Table 21.103**).

Table 21.103 Channel filtering configuration

CH[1:0]	**Transmit Buffer** transmit frame	**Receive Buffer store valid receive frame**
00_B	no transmission	ignore frame
01_B	on channel A	received on channel A
10_B	on channel B	received on channel B
11_B	on both channels (static segment only)	received on channel A or B (store first semantically valid frame, static segment only)

The contents of a transmit buffer is transmitted on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a transmit buffer may be set up for transmission on both channels (CH = "11_B").

Valid received frames are stored if they are received on the channels specified in the channel filtering field when the slot counter filtering and cycle counter filtering criteria are also met. Only in static segment a receive buffer may be setup for reception on both channels (CH = "11_B").

CAUTION

If a message buffer is configured for the dynamic segment and both bits of the channel filtering field are set to 1, no frames are transmitted and received frames are ignored (same function as CH = "00_B").

21.3.8.4 FIFO Filtering

For FIFO filtering registers FLXAnFRFRF and FLXAnFRFRFM are used. The FIFO filter consists of channel filter FLXAnFRFRF.CH, frame ID filter FLXAnFRFRF.FID, and cycle counter filter FLXAnFRFRF.CYF. Registers FLXAnFRFRF and FLXAnFRFRFM can be configured in DEFAULT_CONFIG or CONFIG state only. The filter configuration in the header section of message buffers belonging to the FIFO is ignored.

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles not belonging to the cycle set specified by FLXAnFRFRF.CYF, all frames are rejected.

A valid received frame is stored in the FIFO if channel ID, frame ID, and cycle counter are not rejected by the configured rejection filter and rejection filter mask, and if there is no matching dedicated receive buffer.

21.3.9 Transmit Process

21.3.9.1 Static Segment

For the static segment, if there are several messages pending for transmission, the message with the frame ID corresponding to the next sending slot is selected for transmission.

The data section of transmit buffers assigned to the static segment can be updated until the end of the preceding time slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

21.3.9.2 Dynamic Segment

In the dynamic segment, if several messages are pending, the message with the highest priority (lowest frame ID) is selected next. In the dynamic segment different slot counter sequences on channel A and channel B are possible (concurrent sending of different frame IDs on both channels).

The data section of transmit buffers assigned to the dynamic segment can be updated until the end of the preceding slot. This means that a transfer from the Input Buffer has to be started by writing to the Input Buffer Command Request register latest at this time.

The start of latest transmit configured by FLXAnFRMHDC.SLT defines the maximum minislot value allowed before inhibiting new frame transmission in the dynamic segment of the current cycle.

21.3.9.3 Transmit Buffers

FlexRay message buffers can be configured as transmit buffers by programming bit CFG in the header section of the respective message buffer to 1 via FLXAnFRWRHS1.

There exist the following possibilities to assign a transmit buffer to the CC channels:

- Static segment: channel A or channel B,
  channel A and channel B
- Dynamic segment: channel A or channel B

Message buffers 0 and 1 are dedicated to holding the startup frame, the sync frame, or the designated single slot frame as configured by FLXAnFRSUCC1.TXST, FLXAnFRSUCC1.TXSY, and FLXAnFRSUCC1.TSM. In this case, it can be reconfigured in DEFAULT_CONFIG or CONFIG state only. This ensures that any node transmits at most one startup / sync frame per communication cycle. Transmission of startup / sync frames from other message buffers is not possible.

All other message buffers configured for transmission in static or dynamic segment are reconfigurable during runtime depending on the configuration of FLXAnFRMRC.SEC (see **Section 21.3.12.1, Reconfiguration of Message Buffers**). Due to the organization of the data partition in the Message RAM (reference by data pointer), reconfiguration of the configured payload length and the data pointer in the header section of a message buffer may lead to erroneous configurations.

If a message buffer is reconfigured (header section updated) during runtime, it may happen that this message buffer is not send out in the respective communication cycle.

The CC does not have the capability to calculate the header CRC. The Host is supposed to provide the header CRCs for all transmit buffers. If network management is required, the Host has to set the PPIT bit in the header section of the respective message buffer to 1 and write the network management information to the data section of the message buffer (see **Section 21.3.7, Network Management**).

The payload length field configures the payload length in 2-byte words. If the configured payload length of a static transmit buffer is shorter than the payload length configured for the static segment by

FLXAnFRMHDC.SFDL, the CC generates padding bytes to ensure that frames have proper physical length. The padding pattern is “0000H”.

CAUTION

In case of an odd payload length (PLC = 1,3,5,...) the application has to write zero to the last 16 bit of the message buffers data section to ensure that the padding pattern is “0000_H”.

Each transmit buffer provides a transmission mode flag TXM that allows the Host to configure the transmission mode for the transmit buffer. If this bit is set, the transmitter operates in the single-shot mode. If this bit is cleared, the transmitter operates in the continuous mode.

In single-shot mode the CC resets the respective TXR flag to 0 after transmission has completed. Now the Host may update the transmit buffer.

In continuous mode, the CC does not reset the respective transmission request flag TXR to 0 after successful transmission. In this case a frame is sent out each time the filter criteria match. The TXR flag can be reset to 0 by the Host by writing the respective message buffer number to the FLXAnFRIBCR register while bit FLXAnFRIBCM.STXRH is set to 0.

If two or more transmit buffers meet the filter criteria simultaneously, the transmit buffer with the lowest message buffer number will be transmitted in the respective slot.

21.3.9.4 Frame Transmission

The following steps are required to prepare a message buffer for transmission:

- Configure the transmit buffer in the Message RAM via FLXAnFRWRHS1, FLXAnFRWRHS2, and FLXAnFRWRHS3
- Write the data section of the transmit buffer via FLXAnFRWRDSx
- Transfer the configuration and message data from Input Buffer to the Message RAM by writing the number of the target message buffer to register FLXAnFRIBCR
- If configured in register FLXAnFRIBCM, the transmission request flag TXR for the respective message buffer will be set as soon as the transfer has completed, and the message buffer is ready for transmission.
- Check whether the message buffer has been transmitted by checking the respective TXR bit (TXR = 0) in the FLXAnFRTXRQ1/2/3/4 registers (single-shot mode only).

After transmission has completed, the respective TXR flag in the FLXAnFRTXRQ1/2/3/4 register is reset to 0 (single-shot mode), and, if bit MBI in the header section of the message buffer is set to 1, flag FLXAnFRSIR.TXI is set to 1. If enabled, an interrupt is generated.

21.3.9.5 Null Frame Transmission

If in static segment the Host does not set the transmission request flag to 1 before transmit time, the CC transmits a null frame with the null frame indication bit set to 0 and the payload data set to zero.

In the following cases the CC transmits a null frame:

- If the message buffer with the lowest message buffer number matching the filter criteria does not have its transmission request flag set (TXR = 0) to 1.
- No transmit buffer configured for the slot has a cycle counter filter that matches the current cycle. In this case, no message buffer status FLXAnFRMBS is updated.

Null frames are not transmitted in the dynamic segment.

21.3.10 Receive Process

21.3.10.1 Dedicated Receive Buffers

A portion of the FlexRay message buffers can be configured as dedicated receive buffers by programming bit CFG in the header section of the respective message buffer to 0 via FLXAnFRWRHS1.

The following possibilities exist to assign a receive buffer to the CC channels:

- Static segment: channel A or channel B,
  channel A and channel B (the CC stores the first semantically valid frame)
- Dynamic segment: channel A or channel B

The CC transfers the payload data of valid received messages from the shift register of the FlexRay channel protocol controller (channel A or B) to the receive buffer with the matching filter configuration. A receive buffer stores all frame elements except the frame CRC.

All message buffers configured for reception in static or dynamic segment are reconfigurable during runtime depending on the configuration of FLXAnFRMRC.SEC (see **Section 21.3.12.1, Reconfiguration of Message Buffers**). If a message buffer is reconfigured (header section updated) during runtime it may happen that in the respective communication cycle a received message is lost.

If two or more receive buffers meet the filter criteria simultaneously, the receive buffer with the lowest message buffer number is updated with the received message.

21.3.10.2 Frame Reception

The following steps are required to prepare a dedicated message buffer for reception:

- Configure the receive buffer in the Message RAM via FLXAnFRWRHS1, FLXAnFRWRHS2, and FLXAnFRWRHS3
- Transfer the configuration from Input Buffer to the Message RAM by writing the number of the target message buffer to register FLXAnFRIBCR

Once these steps are performed, the message buffer functions as an active receive buffer and participates in the internal acceptance filtering process which takes place every time the CC receives a message. The first matching receive buffer is updated from the received message.

If a valid payload segment was stored in the data section of a message buffer, the respective ND flag in the FLXAnFRNDAT1/2/3/4 registers is set to 1, and, if bit MBI in the header section of that message buffer is set to 1, flag FLXAnFRSIR.RXI is set to 1. If enabled, an interrupt is generated.

In case that bit ND was already set to 1 when the Message Handler updates the message buffer, bit FLXAnFRMBS.MLST of the respective message buffer is set to 1 and the unprocessed message data is lost.

If no frame, a null frame, or a corrupted frame was received in a slot, the data section of the message buffer configured for this slot is not updated. In this case only the respective message buffer status FLXAnFRMBS is updated.

When the Message Handler changed the message buffer status FLXAnFRMBS in the header section of a message buffer, the respective MBC flag in the FLXAnFRMBSC1/2/3/4 registers is set to 1, and if bit MBI in the header section of that message buffer is set, flag FLXAnFRSIR.MBSI is set to 1. If enabled an interrupt is generated.

If the payload length of a received frame PLR is longer than the value programmed by PLC in the header section of the respective message buffer, the data field stored in the message buffer is truncated to that length.

To read a receive buffer from the Message RAM via the Output Buffer, proceed as described in **Section 21.3.12.2 (2), Data Transfer from Message RAM to Output Buffer**.

CAUTION

The ND and MBC flags are automatically cleared by the Message Handler when the payload data and the header of a received message have been transferred to the Output Buffer, respectively.

21.3.10.3 Null Frame Reception

The payload segment of a received null frame is not copied into the matching dedicated receive buffer. If a null frame has been received, only the message buffer status FLXAnFRMBS of the matching message buffer is updated from the received null frame. All bits in header 2 and 3 of the matching message buffer remain unchanged. They are updated from received data frames only.

When the Message Handler changed the message buffer status FLXAnFRMBS in the header section of a message buffer, the respective MBC flag in the FLXAnFRMBSC1/2/3/4 register is set to 1, and if bit MBI in the header section of that message buffer is set to 1, flag FLXAnFRSIR.MBSI is set to 1. If enabled, an interrupt is generated.

21.3.11 FIFO Function

21.3.11.1 Description

A portion of the message buffers can be configured as a cyclic First-In-First-Out (FIFO) buffer. The group of message buffers belonging to the FIFO is contiguous in the register map starting with the message buffer referenced by FLXAnFRMRC.FFB and ending with the message buffer referenced by FLXAnFRMRC.LCB. Up to 127 message buffers can be assigned to the FIFO.

Every valid incoming message not matching with any dedicated receive buffer but passing the programmable FIFO filter is stored into the FIFO. In this case frame ID, payload length, receive cycle count, and the message buffer status FLXAnFRMBS of the addressed FIFO message buffer are overwritten with frame ID, payload length, receive cycle count, and the status from the received frame. Bit FLXAnFRSIR.RFNE shows that the FIFO is not empty, bit FLXAnFRSIR.RFCL is set to 1 when the receive FIFO fill level FLXAnFRFSR.RFFL is equal or greater than the critical level as configured by FLXAnFRSIR.RFCL, bit FLXAnFREIR.RFO shows that a FIFO overrun has been detected. If enabled, interrupts are generated.

If null frames are not rejected by the FIFO rejection filter, the null frames will be treated like data frames when they are stored into the FIFO.

There are two index registers associated with the FIFO. The PUT Index Register (PIDX) is an index to the next available location in the FIFO. When a new message has been received it is written into the message buffer addressed by the PIDX register. The PIDX register is then incremented and addresses the next available message buffer. If the PIDX register is incremented past the highest numbered message buffer of the FIFO, the PIDX register is loaded with the number of the first (lowest numbered) message buffer in the FIFO chain. The GET Index Register (GIDX) is used to address the next message buffer of the FIFO to be read. The GIDX register is incremented after transfer of the contents of a message buffer belonging to the FIFO to the Output Buffer. The PUT Index Register and the GET Index Register are not accessible by the Host.

The FIFO is completely filled when the PUT index (PIDX) reaches the value of the GET index (GIDX). When the next message is written to the FIFO before the oldest message has been read, both PUT index and GET index are incremented and the new message overwrites the oldest message in the FIFO. This will set FIFO overrun flag FLXAnFREIR.RFO to 1.

A FIFO non empty status is detected when the PUT index (PIDX) differs from the GET index (GIDX). In this case flag FLXAnFRSIR.RFNE is set to 1. This indicates that there is at least one received message in the FIFO. The FIFO empty, FIFO not empty, and the FIFO overrun states are explained in **Figure 21.11** for a three message buffer FIFO.

The programmable FlexRay FIFO Rejection Filter (FLXAnFRFRF) defines a filter pattern for messages to be rejected. The FIFO filter consists of channel filter, frame ID filter, and cycle counter filter. If bit FLXAnFRFRF.RSS is set to 1, all messages received in the static segment are rejected by the FIFO. If bit FLXAnFRFRF.RNF is set to 1, received null frames are not stored in the FIFO.

The FlexRay FIFO Rejection Filter Mask (FLXAnFRFRFM) specifies which bits of the frame ID filter in the FIFO Rejection Filter register are marked 'don't care' for rejection filtering.

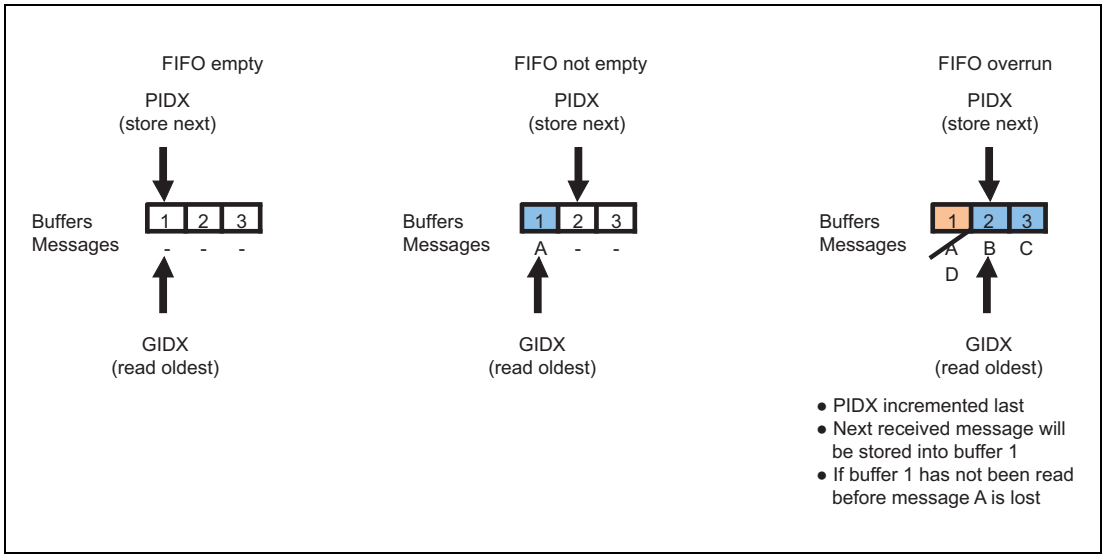


Figure 21.11 FIFO Status: Empty, Not Empty, Overrun

21.3.11.2 Configuration of the FIFO

(Re)configuration of message buffers belonging to the FIFO is only possible when the CC is in DEFAULT_CONFIG or CONFIG state. While the CC is in DEFAULT_CONFIG or CONFIG state, the FIFO function is not available.

For all message buffers belonging to the FIFO the payload length configured should be programmed to the same value via FLXAnFRWRHS2.PLC. The data pointer to the first 32-bit word of the data section of the respective message buffer in the Message RAM has to be configured via FLXAnFRWRHS3.DP.

All information required for acceptance filtering is taken from the FIFO rejection filter and the FIFO rejection filter mask. The values configured in the header sections of the message buffers belonging to the FIFO are, with exception of DP and PLC, irrelevant.

CAUTIONS

1. **It is recommended to program the MBI bits of the message buffers belonging to the FIFO to 0 via FLXAnFRWRHS1.MBI to avoid generation of interrupts.**
2. **If the payload length of a received frame is longer than the value programmed by FLXAnFRWRHS2.PLC in the header section of the respective message buffer, the data field stored in a message buffer of the FIFO is truncated to that length.**

21.3.11.3 Access to the FIFO

(1) When the output buffer is used:

For FIFO access outside DEFAULT_CONFIG and CONFIG state, the Host has to trigger a transfer from the Message RAM to the Output Buffer by writing the number of the first message buffer of the FIFO (referenced by FLXAnFRMRC.FFB) to the register FLXAnFROBCR. The Message Handler then transfers the message buffer addressed by the GET Index Register (GIDX) to the Output Buffer. After this transfer the GET Index Register (GIDX) is incremented.

(2) When the data transfer function is used:

The message received in FIFO can be transferred to the Local RAM/Global RAM by using the output data transfer function. For the output data transfer function, see **Section 21.3.16.2, Output Data Transfer**, Output Data Transfer.

21.3.12 Message Handling

The Message Handler controls data transfers between the Input / Output Buffer and the Message RAM and between the Message RAM and the two Temporary buffers.

Access to the message buffers stored in the Message RAM is done under control of the Message Handler state machine. This avoids conflicts between accesses of the two FlexRay channel protocol controllers and the Host to the Message RAM.

Frame IDs of message buffers assigned to the static segment have to be in the range from 1 to FLXAnFRGTUC7.NSS. Frame IDs of message buffers assigned to the dynamic segment have to be in the range from FLXAnFRGTUC7.NSS + 1 to 2047.

Received messages with no matching dedicated receive buffer (static or dynamic segment) are stored in the receive FIFO (if configured) if they pass the FIFO rejection filter.

Access of the Host to the message buffer contents using the input or output buffer function is described in this subsection. Access to the message buffer contents using the data transfer function is mentioned in **Section 21.3.16, Usage of Data Transfer**.

21.3.12.1 Reconfiguration of Message Buffers

In case that an application needs to operate with more than 128 different messages, static and dynamic message buffers may be reconfigured during FlexRay operation. This is done by updating the header section of the respective message buffer via Input Buffer registers FLXAnFRWRHS1 to FLXAnFRWRHS3.

Reconfiguration has to be enabled via control bits FLXAnFRMRC.SEC in the Message RAM Configuration register.

If a message buffer has not been transmitted / updated from a received frame before reconfiguration starts, the respective message is lost.

The point in time when a reconfigured message buffer is ready for transmission / reception according to the reconfigured frame ID depends on the actual state of the slot counter when the update of the header section has completed. Therefore it may happen that a reconfigured message buffer is not transmitted / updated from a received frame in the cycle where it was reconfigured.

The Message RAM is scanned according to **Table 21.104** below.

Table 21.104 Scan of Message RAM

Start of Scan in Slot	Scan for Slots
1	2…15, 1 (next cycle)
8	16…23, 1 (next cycle)
16	24…31, 1 (next cycle)
24	32…39, 1 (next cycle)
…	…

A Message RAM scan is terminated with the start of NIT regardless whether it has completed or not. The scan of the Message RAM for slots 2 to 15 starts at the beginning of slot 1 of the actual cycle. The scan of the Message RAM for slot 1 is done in the cycle before by checking in parallel to each scan of the Message RAM whether there is a message buffer configured for slot 1 of the next cycle.

The number of the first dynamic message buffer is configured by FLXAnFRMRC.FDB. In case a Message RAM scan starts while the CC is in dynamic segment, the scan starts with the message buffer number configured by FLXAnFRMRC.FDB.

In case a message buffer should be reconfigured to be used in slot 1 of the next cycle, the following has to be considered:

- If the message buffer to be reconfigured for slot 1 is part of the “Static Buffers”, it will only be found if it is reconfigured before the last Message RAM scan in the static segment of the actual cycle evaluates this message buffer.
- If the message buffer to be reconfigured for slot 1 is part of the “Static + Dynamic Buffers”, it will be found if it is reconfigured before the last Message RAM scan in the actual cycle evaluates this message buffer.
- The start of NIT terminates the Message RAM scan. In case the Message RAM scan has not evaluated the reconfigured message buffer until this point in time, the message buffer will not be considered for the next cycle.

CAUTION

Reconfiguration of message buffers may lead to the loss of messages and therefore has to be used very carefully. In worst case (reconfiguration in consecutive cycles) it may happen that a message buffer is never transmitted / updated from a received frame.

21.3.12.2 Host access to Message RAM

The message transfer between Input Buffer and Message RAM as well as between Message RAM and Output Buffer is triggered by the Host by writing the number of the target / source message buffer to be accessed to FLXAnFRIBCR or FLXAnFROBCR register.

The FLXAnFRIBCM and FLXAnFROBCM registers can be used to write / read header and data section of the selected message buffer separately.

If bit FLXAnFRIBCM.STXR is set to 1, the transmission request flag TXR of the selected message buffer is automatically set to 1 after the message buffer has been updated. If bit FLXAnFRIBCM.STXR is reset to 0, the transmission request flag TXR of the selected message buffer is reset. This can be used to stop transmission from message buffers operated in continuous mode.

Input Buffer (IBF) and Output Buffer (OBF) are configured as a double buffer structure. One half of this double buffer structure is accessible by the Host (IBF Host / OBF Host), while the other half (IBF Shadow / OBF Shadow) is accessed by the Message Handler for data transfers between IBF / OBF and Message RAM.

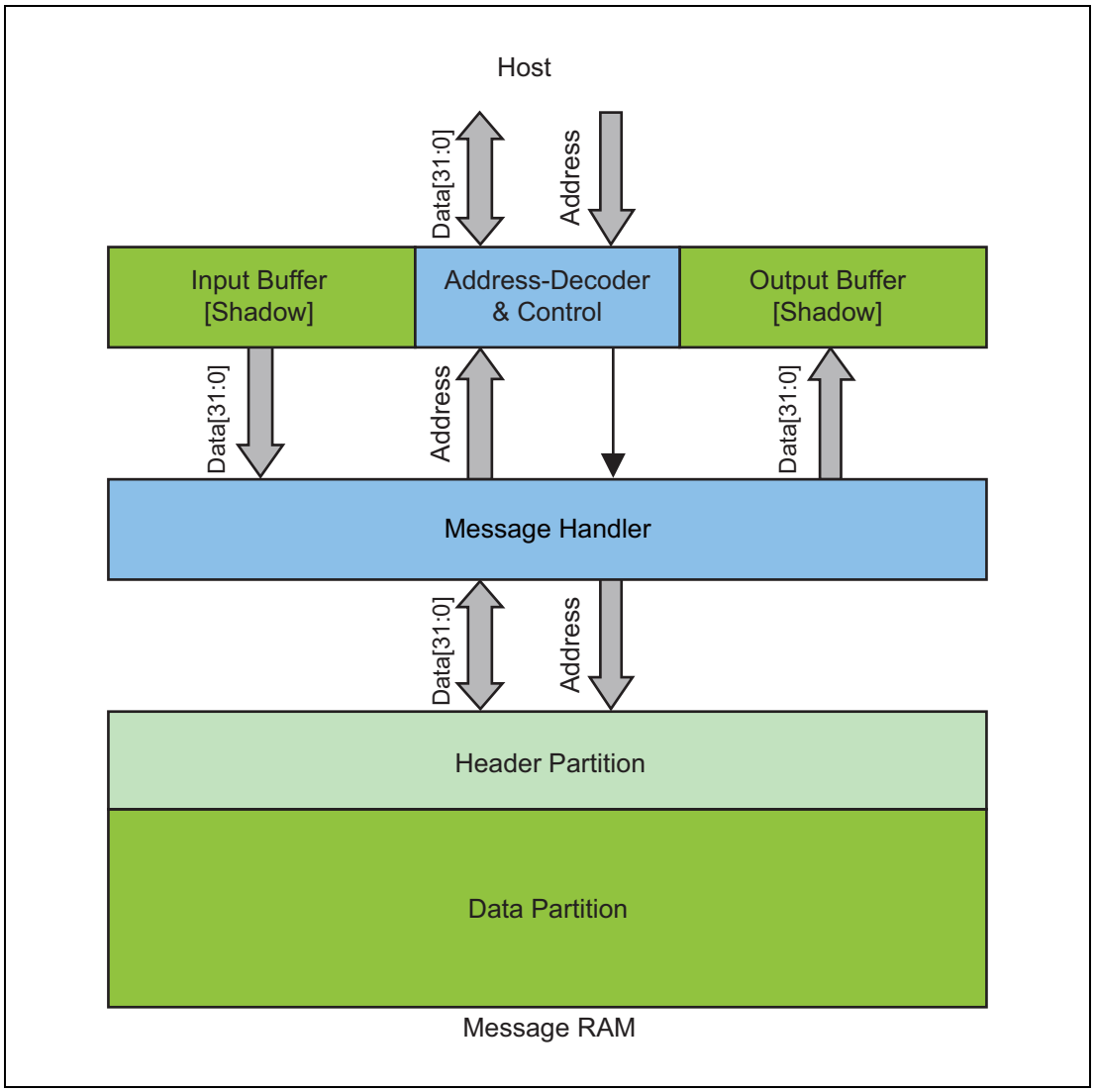


Figure 21.12 Host access to Message RAM

(1) Data Transfer from Input Buffer to Message RAM

To configure / update a message buffer in the Message RAM, the Host has to write the data to FLXAnFRWRDSx and the header to FLXAnFRWRHS1 to FLXAnFRWRHS3. The specific action is selected by configuring the FlexRay Input Buffer Command Mask FLXAnFRIBCM.

When the Host writes the number of the target message buffer in the Message RAM to FLXAnFRIBCR.IBRH, IBF Host and IBF Shadow are swapped (see **Figure 21.13**).

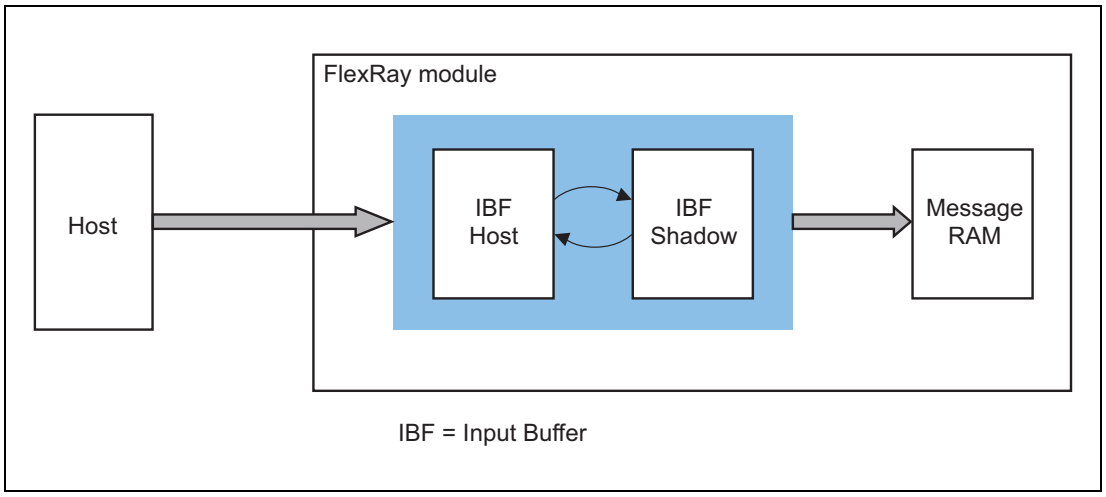


Figure 21.13 Double Buffer Structure Input Buffer

In addition the bits in the FLXAnFRIBCM and FLXAnFRIBCR registers are also swapped to keep them attached to the respective IBF section (see **Figure 21.14**).

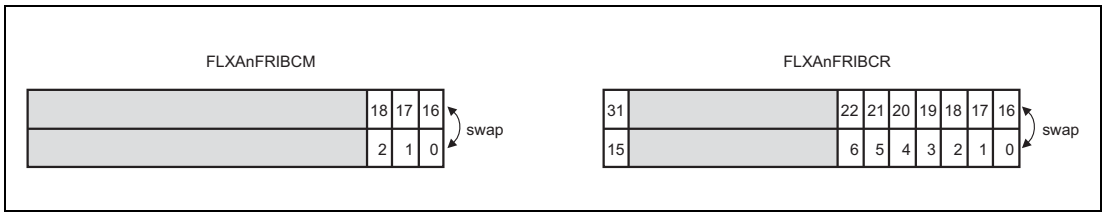


Figure 21.14 Swapping of FLXAnFRIBCM and FLXAnFRIBCR bits

With this write operation bit FLXAnFRIBCR.IBSYS is set to 1. The Message Handler then starts to transfer the contents of IBF Shadow to the message buffer in the Message RAM selected by FLXAnFRIBCR.IBRS.

While the Message Handler transfers the data from IBF Shadow to the target message buffer in the Message RAM, the Host may write the next message to IBF Host. After the transfer between IBF Shadow and the Message RAM has completed, bit FLXAnFRIBCR.IBSYS is set back to 0 and the next transfer to the Message RAM may be started by the Host by writing the respective target message buffer number to FLXAnFRIBCR.IBRH.

If a write access to FLXAnFRIBCR.IBRH occurs while FLXAnFRIBCR.IBSYS is 1, FLXAnFRIBCR.IBSYH is set to 1. After completion of the ongoing data transfer from IBF Shadow to the Message RAM, IBF Host and IBF Shadow are swapped, FLXAnFRIBCR.IBSYH is reset to 0, FLXAnFRIBCR.IBSYS remains set to 1, and the next transfer to the Message RAM is started. In addition the message buffer numbers stored under FLXAnFRIBCR.IBRH and FLXAnFRIBCR.IBRS and the command mask flags are also swapped.

Example of 8/16/32-bit Host access sequence:

Configure / update n-th message buffer via IBF

- Wait until FLXAnFRIBCR.IBSYH is reset
- Write data section to FLXAnFRWRDSx
- Write header section to FLXAnFRWRHS1 to FLXAnFRWRHS3
- Write Command Mask: write FLXAnFRIBCM.STXRH, FLXAnFRIBCM.LDSH, FLXAnFRIBCM.LHSH
- Demand data transfer to target message buffer: write FLXAnFRIBCR.IBRH

Configure / update (n+1)th message buffer via IBF

- Wait until FLXAnFRIBCR.IBSYH is reset
- Write data section to FLXAnFRWRDSx
- Write header section to FLXAnFRWRHS1 to FLXAnFRWRHS3
- Write Command Mask: write FLXAnFRIBCM.STXRH, FLXAnFRIBCM.LDSH, FLXAnFRIBCM.LHSH
- Demand data transfer to target message buffer: write FLXAnFRIBCR.IBRH

CAUTION

Any write access to IBF while FLXAnFRIBCR.IBSYH is 1 will set error flag FLXAnFREIR.IIBA to 1. In this case the write access has no effect.

Table 21.105 Assignment of FLXAnFRIBCM Bits

Pos.	Access	Bit	Function
18	r	STXRS	Set Transmission Request Shadow ongoing or finished
17	r	LDSS	Load Data Section Shadow ongoing or finished
16	r	LHSS	Load Header Section Shadow ongoing or finished
2	r/w	STXRH	Set Transmission Request Host
1	r/w	LDSH	Load Data Section Host
0	r/w	LHSH	Load Header Section Host

Table 21.106 Assignment of FLXAnFRIBCR Bits

Pos.	Access	Bit	Function
31	r	IBSYS	IBF Busy Shadow, signals ongoing transfer from IBF Shadow to Message RAM
22 to 16	r	IBRS	IBF Request Shadow, number of message buffer from which a message is currently being transferred or was last transferred
15	r	IBSYH	IBF Busy Host, transfer request pending for message buffer referenced by IBRH
6 to 0	r/w	IBRH	IBF Request Host, number of message buffer from which a message is to be transferred next

(2) Data Transfer from Message RAM to Output Buffer

To read a message buffer from the Message RAM, the Host has to write to register FLXAnFROBCR to trigger the data transfer as configured in FLXAnFROBCM. After the transfer has completed, the Host can read the transferred data from FLXAnFRRDDSx, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS.

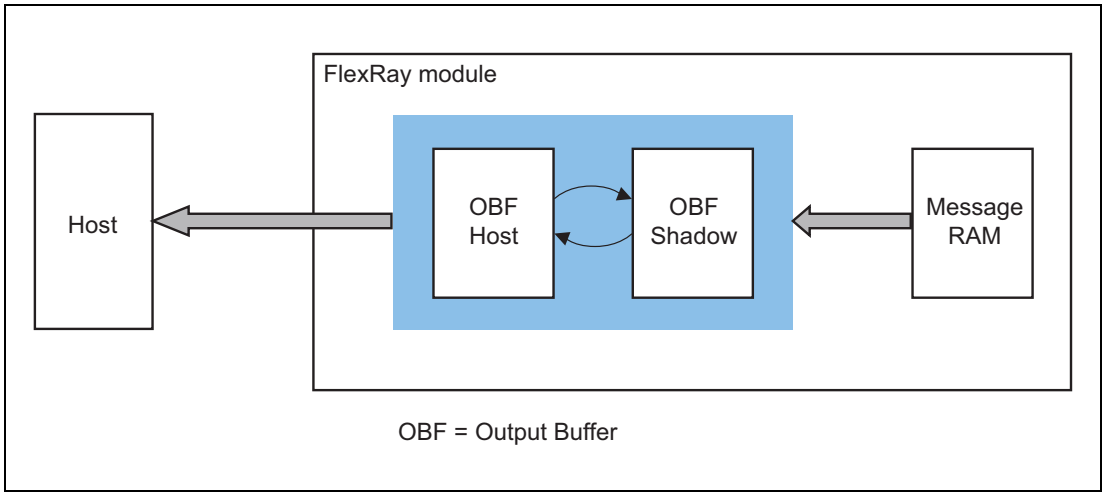


Figure 21.15 Double buffer structure Output Buffer

OBF Host and OBF Shadow as well as bits FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS, FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH and bits FLXAnFROBCR.OBRS, FLXAnFROBCR.OBRH are swapped under control of bits FLXAnFROBCR.VIEW and FLXAnFROBCR.REQ.

Writing bit FLXAnFROBCR.REQ to 1 copies bits FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS and bits FLXAnFROBCR.OBRS to an internal storage (see **Figure 21.16**).

After setting FLXAnFROBCR.REQ to 1, FLXAnFROBCR.OBSYS is set to 1, and the transfer of the message buffer selected by FLXAnFROBCR.OBRS from the Message RAM to OBF Shadow is started. After the transfer between the Message RAM and OBF Shadow has completed, the FLXAnFROBCR.OBSYS bit is set back to 0. Bits FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW can only be set to 1 while FLXAnFROBCR.OBSYS is 0.

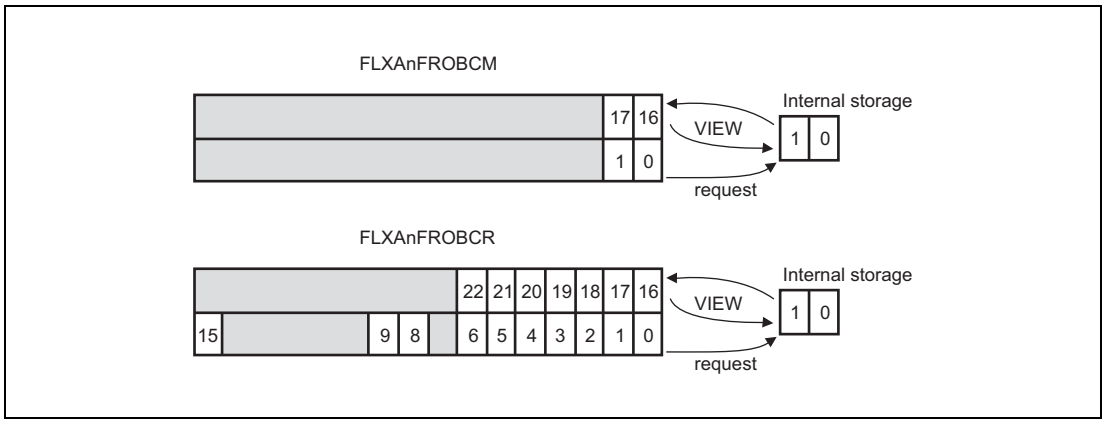


Figure 21.16 Swapping of FLXAnFROBCM and FLXAnFROBCR bits

OBF Host and OBF Shadow are swapped by setting bit FLXAnFROBCR.VIEW to 1 while bit FLXAnFROBCR.OBSYS is 0 (see **Figure 21.15**).

In addition bits FLXAnFROBCR.OBRH and bits FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH are swapped with the registers internal storage thus assuring that the message buffer number stored in FLXAnFROBCR.OBRH and the mask configuration stored in FLXAnFROBCM.RHSH, FLXAnFROBCM.RDSH matches the transferred data stored in OBF Host (see **Figure 21.16**).

Now the Host can read the transferred message buffer from OBF Host while the Message Handler may transfer the next message from the Message RAM to OBF Shadow.

If bits REQ and VIEW are set to 1 with the same write access while FLXAnFROBSYS is 0, FLXAnFROBSYS is automatically set to 1 and OBF Shadow and OBF Host are swapped. Additionally mask bits FLXAnFROBCM.RDSH and FLXAnFROBCM.RHSH are swapped with the registers internal storage to keep them attached to the respective Output Buffer transfer. Afterwards FLXAnFROBCR.OBRS is copied to the register internal storage, mask bits FLXAnFROBCM.RDSS and FLXAnFROBCM.RHSS are copied to register FLXAnFROBCM internal storage, and the transfer of the selected message buffer from the Message RAM to OBF Shadow is started. While the transfer is ongoing the Host can read the message buffer transferred by the previous transfer from OBF Host. When the current transfer between Message RAM and OBF Shadow has completed, this is signaled by setting FLXAnFROBCR.OBSYS back to 0.

Example of an 8/16/32-bit Host access to a single message buffer:

If a single message buffer has to be read out, two separate write accesses to FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW are necessary:

- Wait until FLXAnFROBCR.OBSYS is reset
- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS
- Request transfer of message buffer to OBF Shadow by writing FLXAnFROBCR.OBRS and FLXAnFROBCR.REQ (in case of and 8-bit Host interface, FLXAnFROBCR.OBRS has to be written before FLXAnFROBCR.REQ).
- Wait until FLXAnFROBCR.OBSYS is reset
- Toggle OBF Shadow and OBF Host by writing FLXAnFROBCR.VIEW = 1
- Read out transferred message buffer by reading FLXAnFRRDDSx, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

Example of an 8/16/32-bit Host access sequence:

Request transfer of 1st message buffer to OBF Shadow

- Wait until FLXAnFROBCR.OBSYS is reset
- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS for 1st message buffer
- Request transfer of 1st message buffer to OBF Shadow by writing FLXAnFROBCR.OBRS and FLXAnFROBCR.REQ (in case of an 8-bit Host interface, FLXAnFROBCR.OBRS has to be written before FLXAnFROBCR.REQ).

Toggle OBF Shadow and OBF Host to read out 1st transferred message buffer and request transfer of 2nd message buffer:

- Wait until FLXAnFROBCR.OBSYS is reset to 0

- Write Output Buffer Command Mask FLXAnFROBCM.RHSS, FLXAnFROBCM.RDSS for 2nd message buffer
- Toggle OBF Shadow and OBF Host and start transfer of 2nd message buffer to OBF Shadow simultaneously by writing FLXAnFROBCR.OBRS of 2nd message buffer, FLXAnFROBCR.REQ, and FLXAnFROBCR.VIEW (in case of and 8-bit Host interface, FLXAnFROBCR.OBRS has to be written before FLXAnFROBCR.REQ and FLXAnFROBCR.VIEW).
- Read out 1st transferred message buffer by reading FLXAnFRRDDSx, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

…

Demand access to last requested message buffer without request of another message buffer:

- Wait until FLXAnFROBCR.OBSYS is reset to 0
- Demand access to last transferred message buffer by writing FLXAnFROBCR.VIEW
- Read out last transferred message buffer by reading FLXAnFRRDDSx, FLXAnFRRDHS1 to FLXAnFRRDHS3, and FLXAnFRMBS

Table 21.107 Assignment of FLXAnFROBCM bits

Pos.	Access	Bit	Function
17	r	RDSH	Data Section available for Host access
16	r	RHSH	Header Section available for Host access
1	r/w	RDSS	Read Data Section Shadow
0	r/w	RHSS	Read Header Section Shadow

Table 21.108 Assignment of FLXAnFROBCR bits

Pos.	Access	Bit	Function
22 to 16	r	OBRH	OBF Request Host, number of message buffer available for Host access
15	r	OBSYS	OBF Busy Shadow, signals ongoing transfer from Message RAM to OBF Shadow
9	r/w	REQ	Request Transfer from Message RAM to OBF Shadow
8	r/w	VIEW	View OBF Shadow, swap OBF Shadow and OBF Host
6 to 0	r/w	OBRS	OBF Request Shadow, number of message buffer from which a message is to be transferred next

21.3.12.3 FlexRay Protocol Controller Access to Message RAM

The two Temporary buffers (TBF A,B) are used to buffer the data for transfer between the two FlexRay Protocol Controllers and the Message RAM.

Each Temporary buffer is build up as a double buffer, able to store two complete FlexRay messages. There is always one buffer assigned to the corresponding Protocol Controller while the other one is accessible by the Message Handler.

If, for example, the Message Handler writes the next message to be sent to Temporary buffer Tx, the FlexRay Channel Protocol Controller can access Temporary buffer Rx to store the message it is actually receiving. During transmission of the message stored in Temporary buffer Tx, the Message Handler transfers the last received message stored in Temporary buffer Rx to the Message RAM (if it passes acceptance filtering) and updates the respective message buffer.

Data transfers between the Temporary buffers and the shift registers of the FlexRay Channel Protocol Controllers are done in words of 32 bit. This enables the use of a 32 bit shift register independent of the length of the FlexRay messages.

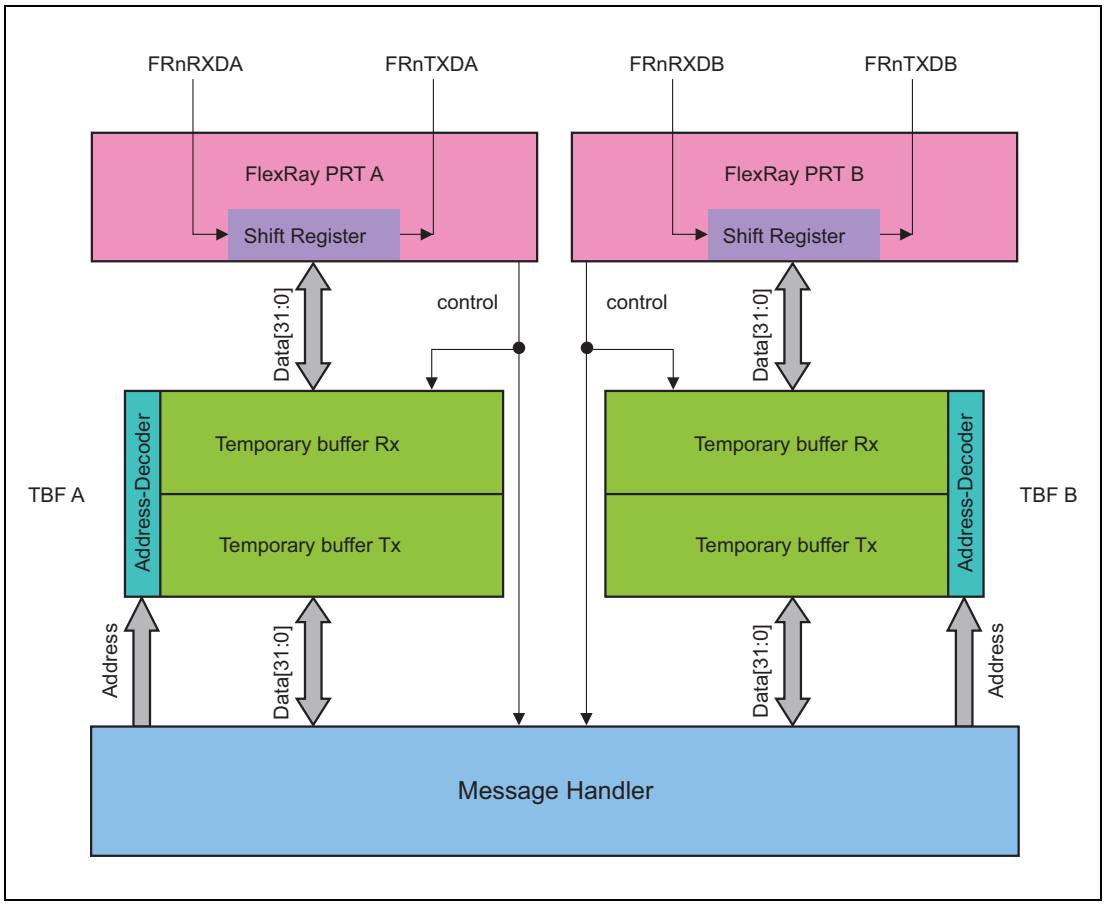


Figure 21.17 Access to Temporary Buffers

21.3.13 Message RAM

To avoid conflicts between Host access to the Message RAM and FlexRay message reception / transmission, the Host cannot directly access the message buffers in the Message RAM. These accesses are handled via the Input and Output Buffers. The Message RAM is able to store up to 128 message buffers depending on the configured payload length.

The Message RAM is able to store up to 2048 32-bit words. To achieve the required flexibility with respect to different numbers of data bytes per FlexRay frame (0 to 254), the Message RAM has a structure as shown in **Figure 21.18**.

When specifying in the data section of the message buffer allocated immediately after the header partition that the message buffer is a receive buffer (FLXAnFRWRHS1.CFG bit = 0) or a receive FIFO buffer, configure an unused area of at least 32 bits at the start of the data partition. This means that the data partition can start from the message RAM word No. calculated by ((Value set by FLXAnFRMRC.LCB[7:0] bits +1) × 4) + 1.

When specifying in the data section of the message buffer allocated immediately after the header partition that the message buffer is a transmit buffer (FLXAnFRWRHS1.CFG bit = 1), the data partition can start from the message RAM word No. calculated by ((Value set by FLXAnFRMRC.LCB[7:0] bits + 1) × 4).

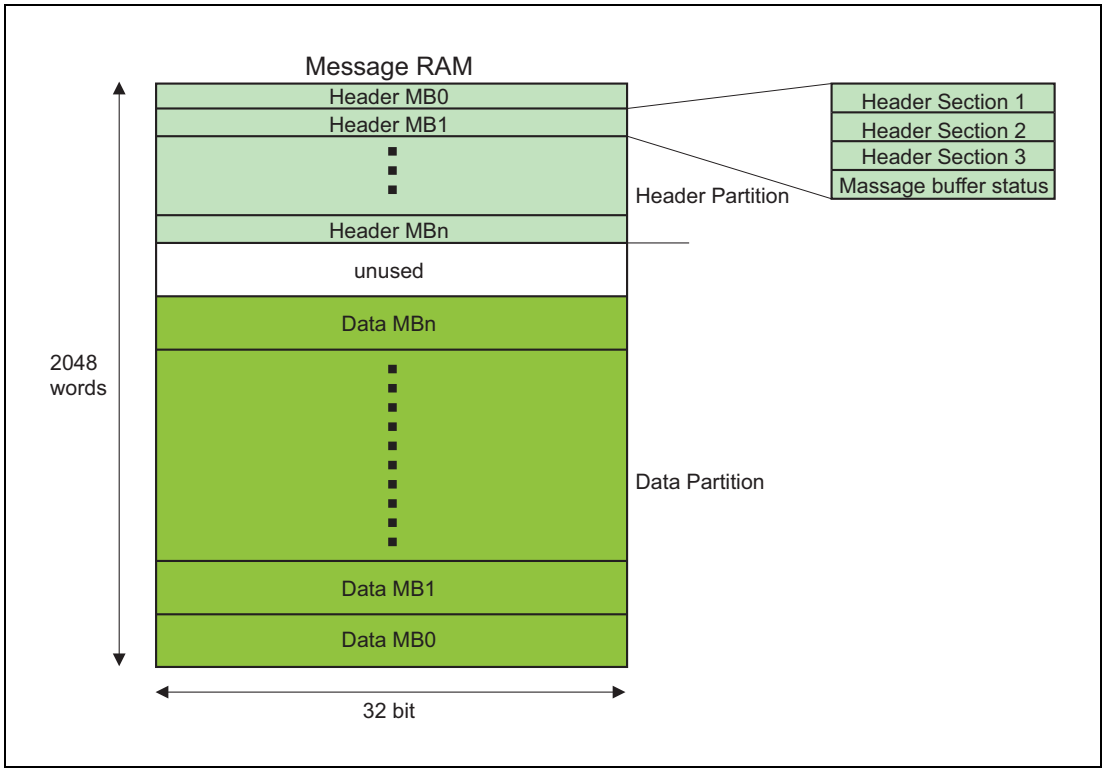


Figure 21.18 Configuration Example of Message Buffers in the Message RAM

Header Partition

Stores header sections of the configured message buffers:

- Supports a maximum of 128 message buffers
- Each message buffer has a header section of four 32 bit words
- Header 3 of each message buffer holds the 11-bit data pointer to the respective data section in the data partition

Data Partition

Flexible storage of data sections with different length. Some maximum values are:

- 30 message buffers with 254 byte data section each
- Or 56 message buffers with 128 byte data section each
- Or 128 message buffers with 48 byte data section each

CAUTION

header partition + data partition may not occupy more than 2048 32-bit words.

21.3.13.1 Header Partition

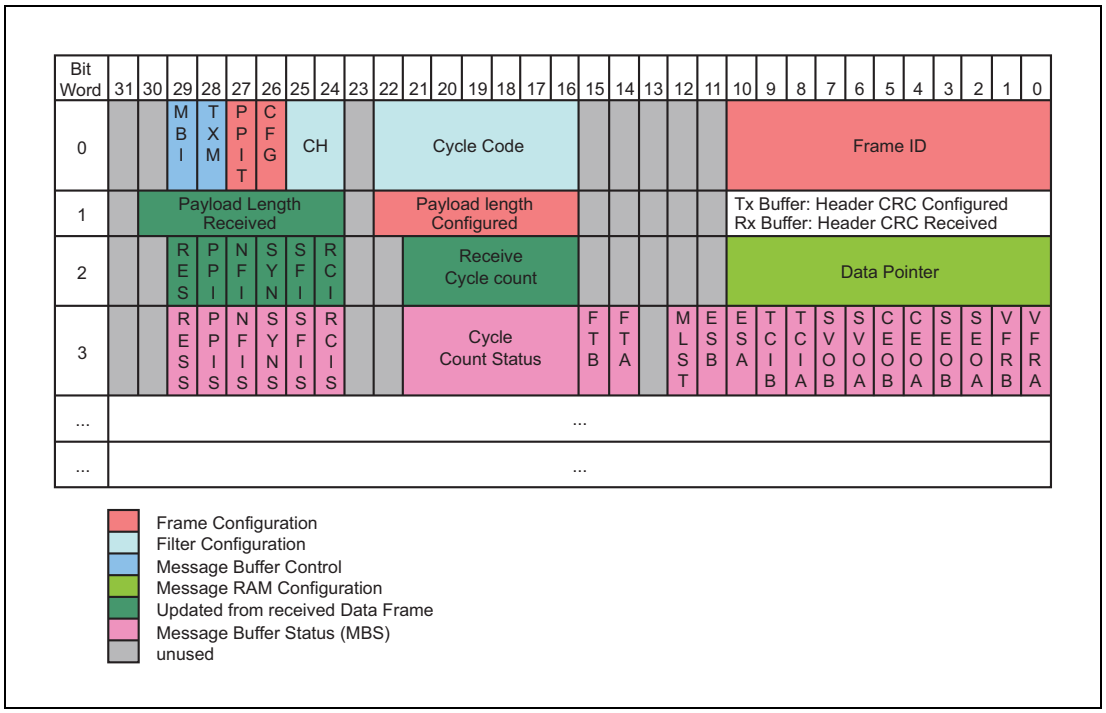
The elements used for configuration of a message buffer as well as the actual message buffer status are stored in the header partition of the Message RAM as listed in **Table 21.109** below. Configuration of the header sections of the message buffers is done via IBF (FLXAnFRWRHS1 to FLXAnFRWRHS3). Read access to the header sections is done via OBF (FLXAnFRRDHS1 to FLXAnFRRDHS3 + FLXAnFRMBS). The data pointer has to be calculated by the user to define the starting point of the data section for the respective message buffer in the data partition of the Message RAM. The data pointer should not be modified during runtime. For message buffers belonging to the receive FIFO (re)configuration is possible in DEFAULT_CONFIG or CONFIG state only.

The header section of each message buffer occupies four 32-bit words in the header partition of the Message RAM. The header of message buffer 0 starts with the first word in the Message RAM.

For transmit buffers the Header CRC has to be calculated by the Host.

Payload Length Received PLR, Receive Cycle Count RCC, Received on Channel Indicator RCI, Startup Frame Indicator SFI, Sync Frame Indicator SYN, Null Frame Indicator NFI, Payload Preamble Indicator PPI, and Reserved Bit RES are updated from received valid data frames only.

Table 21.109 Header Section of a Message Buffer in the Message RAM



(1) Header section 1 (word 0)

Write access via FLXAnFRWRHS1, read access via FLXAnFRRDHS1:

- Frame ID
  - Slot counter filtering configuration
- Cycle Code
  - Cycle counter filtering configuration
- CH
  - Channel filtering configuration
- CFG
  - Message buffer direction configuration: receive / transmit
- PPIT
  - Payload Preamble Indicator Transmit
- TXM
  - Transmit mode configuration: single-shot / continuous
- MBI
  - Message buffer receive / transmit interrupt enable

(2) Header section 2 (word 1)

Write access via FLXAnFRWRHS2, read access via FLXAnFRRDHS2:

- Header CRC
  - Transmit Buffer: Configured by the Host (calculated from frame header)
  - Receive Buffer: Updated from received frame
- Payload Length Configured
  - Length of data section (2-byte words) as configured by the Host
- Payload Length Received
  - Length of payload segment (2-byte words) stored from received frame

(3) Header section 3 (word 2)

Write access via FLXAnFRWRHS3, read access via FLXAnFRRDHS3:

- Data Pointer
  - Pointer to the beginning of the corresponding data section in the data partition

Read access via FLXAnFRRDHS3, valid for receive buffers only, updated from received frames:

- Receive Cycle Count
  - Cycle count from received frame
- RCI
  - Received on Channel Indicator

- SFI
  - Startup Frame Indicator
- SYN
  - Sync Frame Indicator
- NFI
  - Null Frame Indicator
- PPI
  - Payload Preamble Indicator
- RES
  - Reserved bit

(4) Message Buffer Status FLXAnFRMBS (word 3)

Read access via FLXAnFRMBS, updated by the CC at the end of the configured slot.

- VFRA
  - Valid Frame Received on channel A
- VFRB
  - Valid Frame Received on channel B
- SEOA
  - Syntax Error Observed on channel A
- SEOB
  - Syntax Error Observed on channel B
- CEOA
  - Content Error Observed on channel A
- CEOB
  - Content Error Observed on channel B
- SVOA
  - Slot boundary Violation Observed on channel A
- SVOB
  - Slot boundary Violation Observed on channel B
- TCIA
  - Transmission Conflict Indication channel A
- TCIB
  - Transmission Conflict Indication channel B
- ESA
  - Empty Slot Channel A

- ESB
  - Empty Slot Channel B
- MLST
  - Message Lost
- FTA
  - Frame Transmitted on Channel A
- FTB
  - Frame Transmitted on Channel B
- Cycle Count Status
  - Actual cycle count when status was updated
- RCIS
  - Received on Channel Indicator Status
- SFIS
  - Startup Frame Indicator Status
- SYNS
  - Sync Frame Indicator Status
- NFIS
  - Null Frame Indicator Status
- PPIS
  - Payload Preamble Indicator Status
- RESS
  - Reserved bit Status

21.3.13.2 Data Partition

The data partition of the Message RAM stores the data sections of the message buffers configured for reception / transmission as defined in the header partition. The number of data bytes for each message buffer can vary from 0 to 254. To optimize the data transfer between the shift registers of the two FlexRay Protocol Controllers and the Message RAM as well as between the Host interface and the Message RAM, the physical width of the Message RAM is set to 4 bytes.

The data partition starts after the last word of the header partition. When configuring the message buffers in the Message RAM the user has to confirm that the data pointers point to addresses within the data partition. **Table 21.110** below shows an example how the data sections of the configured message buffers can be stored in the data partition of the Message RAM.

The beginning and the end of a message buffer's data section is determined by the data pointer and the payload length configured in the message buffer's header section, respectively. This enables a flexible usage of the available RAM space for storage of message buffers with different data length.

If the size of the data section is an odd number of 2-byte words, the remaining 16 bits in the last 32-bit word are unused (see **Table 21.110** below).

Table 21.110 Example for Structure of the Data Partition in the Message RAM

Bit Word	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
...	unused								unused								unused								unused							
...	unused								unused								unused								unused							
...	MBn Data3								MBn Data2								MBn Data1								MBn Data0							
...	...								...								...								...							
...	...								...								...								...							
...	MBn Data(m)								MBn Data(m-1)								MBn Data(m-2)								MBn Data(m-3)							
...	...								...								...								...							
...	...								...								...								...							
...	...								...								...								...							
...	MB1 Data3								MB1 Data2								MB1 Data1								MB1 Data0							
...	...								...								...								...							
2046	MB0 Data3								MB0 Data2								MB0 Data1								MB0 Data0							
2047	unused								unused								MB0 Data5								MB0 Data4							

21.3.13.3 Message Data Integrity Check

There is a data integrity checking mechanism implemented in the FlexRay core to ensure the integrity of the data stored in the related RAM. Each RAM has a checksum generator / checker attached as shown in **Figure 21.19**.

When data is written to a RAM, the local checksum generator generates the checksum. The checksum is stored together with the respective data word. The checksum is checked each time a data word is read from a RAM.

If a checksum error is detected, the respective access error flag is set. The access error flags FLXAnFRMHDS.AMR, FLXAnFRMHDS.ATBF1, FLXAnFRMHDS.ATBF2 and the faulty message buffer indicators FLXAnFRMHDS.FMBD, FLXAnFRMHDS.MFMB, FLXAnFRMHDS.FMB are located in the FlexRay Message Handler Status register. These single access error flags control the error interrupt flag FLXAnFREIR.AERR.

Figure 21.19 shows the data paths between the Input Buffer, Temporary Buffer and Message RAM.

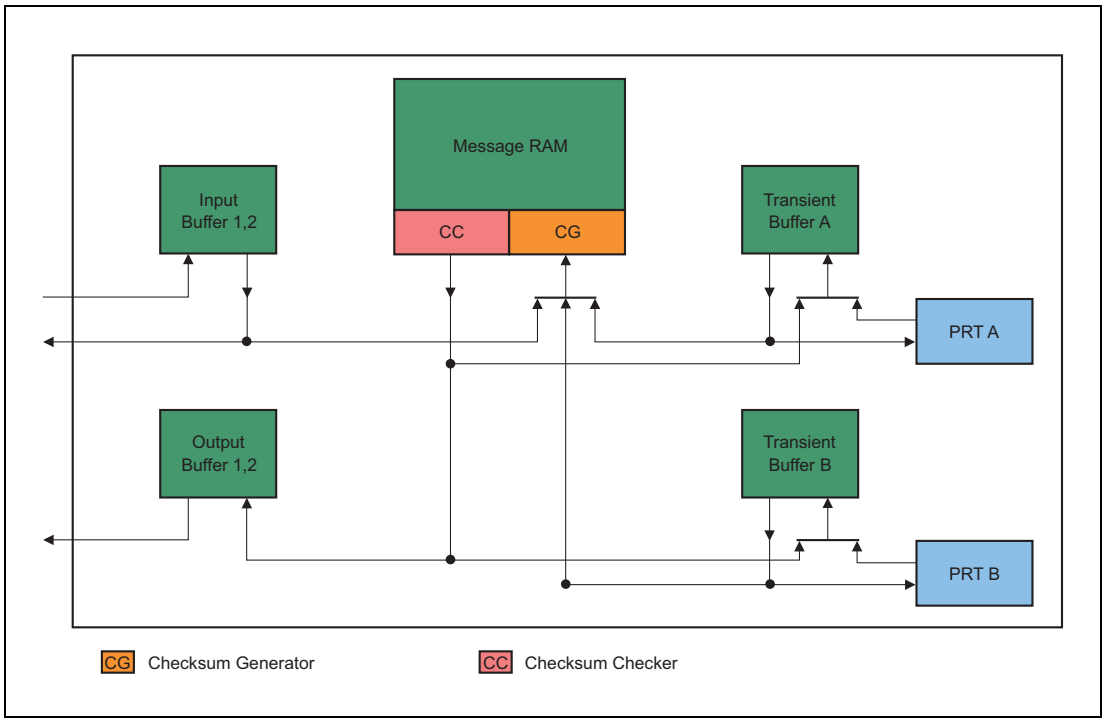


Figure 21.19 Checksum Generation and Check

When an access error has been detected the following actions will be performed:

In all cases:

- The respective access error flag in FLXAnFRMHDS register is set
- The access error flag FLXAnFREIR.AERR is set and, if enabled, a module interrupt to the Host will be generated.

Additionally in specific cases:

(1) Access error during data transfer from Input Buffer 1, 2 to Message RAM when reading header section of respective message buffer from Message RAM:

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.
- The data section of the respective message buffer is not updated.
- Transmit buffer: Transmission request for the respective message buffer is not set.

(2) Access error during scan of header sections in Message RAM:

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.
- Ignore message buffer (message buffer is skipped).

(3) Access error during data transfer from Message RAM to Temporary Buffer 1, 2:

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.
- Frame not transmitted, frames already in transmission are invalidated by setting the frame CRC to zero.

(4) Access error during data transfer from Temporary Buffer 1, 2 to Message RAM when reading header section of respective message buffer from Message RAM:

- FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.
- The data section of the respective message buffer is not updated.

(5) Access error during data transfer from Message RAM to Output Buffer:

- The access error flag FLXAnFRMHDS.AMR is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.

(6) Access error during a data transfer from Temporary Buffer 1, 2 to Protocol Controller 1, 2:

- FLXAnFRMHDS.ATBF1, 2 bit is set.
- Frames already in transmission are invalidated by setting the frame CRC to zero.

(7) Access error during data transfer from Temporary Buffer 1, 2 to Message RAM when reading Temporary Buffer RAM 1, 2:

- FLXAnFRMHDS.ATBF1, 2 bit is set.
- FLXAnFRMHDS.FMBD bit is set to indicate that FLXAnFRMHDS.FMB points to a faulty message buffer.
- FLXAnFRMHDS.FMB indicates the number of the faulty message buffer.

(8) Access error during data read of Temporary Buffer RAM 1, 2:

- When an access error occurs while the Message Handler read a frame with network management information (PPI = 1) from the Temporary Buffer RAM 1, 2, the corresponding network management vector registers FLXAnFRNMV1 to 3 are not updated from this frame.

21.3.13.4 Host Handling of Access Errors

Access error caused by temporary bit flips can be fixed by:

(1) Self-healing

Access errors located in the Data Section of Message RAM, Temporary Buffer RAM A or Temporary Buffer RAM B are overwritten with the next write access to the disturbed bit(s) caused by Host access or by FlexRay communication.

(2) CLEAR_RAMS Command

The POC command CLEAR_RAMS initializes the message RAM to zero, when called in the DEFAULT_CONFIG or CONFIG state.

(3) Temporary Unlocking of Header Section

An access error in the header section of a locked message buffer can be fixed by a transfer from the Input Buffer to the locked buffer Header Section. For this transfer, the write access to the FLXAnFRIBCR (specifying the message buffer number) must be immediately preceded by the unlock sequence normally used to leave CONFIG state (see **Section 21.2.3.3, FLXAnFRLCK — FlexRay Lock Register**).

For that single transfer the respective message buffer header is unlocked, regardless whether it belongs to the FIFO or whether its locking is controlled by FLXAnFRMRC.SEC, and will be updated with new data.

21.3.14 Module Interrupt

In general, interrupts provide a close link to the protocol timing as they are triggered almost immediately when an error or status change is detected by the CC, a frame is received or transmitted, a configured timer interrupt is activated, or a stop watch event occurred. This enables the Host to react very quickly on specific error conditions, status changes, or timer events. On the other hand too many interrupts can cause the Host to miss deadlines required for the application. Therefore the CC supports enable / disable controls for each individual interrupt source separately.

An interrupt request may be triggered when

- An error was detected
- A status flag is set to 1
- A timer reaches a preconfigured value
- A message transfer from Input Buffer to Message RAM or from Message RAM to Output Buffer has completed
- A message transfer from the Local RAM/Global RAM to Message RAM or from Message RAM to Local RAM/Global RAM has completed
- A stop watch event occurred

Tracking status and generating interrupts when a status change or an error occurs are two independent tasks. Regardless of whether an interrupt is enabled or not, the corresponding status is tracked and indicated by the CC. The Host has access to the actual status and error information by reading registers FLXAnFREIR, FLXAnFRSIR, FLXAnFROS, FLXAnFROTS and FLXAnFRITS.

The general purpose interrupt lines to the Host, FlexRay Interrupt 0, FlexRay Interrupt 1, are controlled by the enabled interrupts in FLXAnFREIES and FLXAnFRSIES. In addition each of the two interrupt lines can be enabled / disabled separately by programming bit FLXAnFRILE.EINT0 and FLXAnFRILE.EINT1.

The input data transfer interrupt lines to the Host, FlexRay input queue empty interrupt, FlexRay input queue full interrupt, are controlled by the enabled interrupts in FLXAnFRITS. In addition each of the input data transfer interrupts can be enabled / disabled separately by programming the related bits in FLXAnFRITC.

The output data transfer interrupt lines to the Host, FlexRay FIFO transfer warning interrupt, FlexRay output transfer warning interrupt, FlexRay FIFO transfer interrupt, FlexRay output transfer interrupt, are controlled by the enabled interrupts in FLXAnFROTS. In addition each of the output data transfer interrupts can be enabled / disabled separately by programming the related bits in FLXAnFROTC.

The three timer interrupts lines to the Host are controlled by the enabled interrupts in FLXAnFROS. In addition each of the interrupt lines can be enabled / disabled separately by programming bit FLXAnFROC.T0IE, FLXAnFROC.T1IE and FLXAnFROC.T2IE.

When a transfer between IBF / OBF and the Message RAM has completed bit FLXAnFRSIR.TIBC or FLXAnFRSIR.TOBC is set to 1.

A stop watch event may be triggered via input pin FLXAnSTPWT.

21.3.15 Assignment of FlexRay Configuration Parameters

Table 21.111 FlexRay configuration parameters (1/2)

Parameter	Bit (field)
pKeySlotusedForStartup	FLXAnFRSUCC1.TXST
pKeySlotUsedForSync	FLXAnFRSUCC1.TXSY
gColdStartAttempts	FLXAnFRSUCC1.CSA
pAllowPassiveToActive	FLXAnFRSUCC1.PTA
pWakeupChannel	FLXAnFRSUCC1.WUCS
pSingleSlotEnabled	FLXAnFRSUCC1.TSM
pAllowHaltDueToClock	FLXAnFRSUCC1.HCSE
pChannels	FLXAnFRSUCC1.CCH
pdListenTimeOut	FLXAnFRSUCC2.LT
gListenNoise	FLXAnFRSUCC2.LTN
gMaxWithoutClockCorrectionPassive	FLXAnFRSUCC3.WCP
gMaxWithoutClockCorrectionFatal	FLXAnFRSUCC3.WCF
gNetworkManagementVectorLength	FLXAnFRNEMC.NML
gdTSSTransmitter	FLXAnFRPRTC1.TSST
gdCASRxLowMax	FLXAnFRPRTC1.CASM
gdSampleClockPeriod	FLXAnFRPRTC1.BRP
pSamplesPerMicrotick	FLXAnFRPRTC1.BRP
gdWakeupSymbolRxWindow	FLXAnFRPRTC1.RXW
pWakeupPattern	FLXAnFRPRTC1.RWP
gdWakeupSymbolRxIdle	FLXAnFRPRTC2.RXI
gdWakeupSymbolRxLow	FLXAnFRPRTC2.RXL
gdWakeupSymbolTxIdle	FLXAnFRPRTC2.TXI
gdWakeupSymbolTxLow	FLXAnFRPRTC2.TXL
gPayloadLengthStatic	FLXAnFRMHDC.SFDL
pLatestTx	FLXAnFRMHDC.SLT
pMicroPerCycle	FLXAnFRGTUC1.UT
gMacroPerCycle	FLXAnFRGTUC2.MPC
gSyncNodeMax	FLXAnFRGTUC2.SNM
pMicroInitialOffset[A]	FLXAnFRGTUC3.UIOA
pMicroInitialOffset[B]	FLXAnFRGTUC3.UIOB
pMacroInitialOffset[A]	FLXAnFRGTUC3.MIOA
pMacroInitialOffset[B]	FLXAnFRGTUC3.MIOB
gdNIT	FLXAnFRGTUC4.NIT
gOffsetCorrectionStart	FLXAnFRGTUC4.OCS
pDelayCompensation[A]	FLXAnFRGTUC5.DCA
pDelayCompensation[B]	FLXAnFRGTUC5.DCB
pClusterDriftDamping	FLXAnFRGTUC5.CDD
pDecodingCorrection	FLXAnFRGTUC5.DEC
pdAcceptedStartupRange	FLXAnFRGTUC6.ASR
pdMaxDrift	FLXAnFRGTUC6.MOD
gdStaticSlot	FLXAnFRGTUC7.SSL
gNumberOfStaticSlots	FLXAnFRGTUC7.NSS

Table 21.111 FlexRay configuration parameters (2/2)

Parameter	Bit (field)
gdMinislot	FLXAnFRGTUC8.MSL
gNumberOfMinislots	FLXAnFRGTUC8.NMS
gdActionPointOffset	FLXAnFRGTUC9.APO
gdMinislotActionPointOffset	FLXAnFRGTUC9.MAPO
gdDynamicSlotIdlePhase	FLXAnFRGTUC9.DSI
pOffsetCorrectionOut	FLXAnFRGTUC10.MOC
pRateCorrectionOut	FLXAnFRGTUC10.MRC
pExternOffsetCorrection	FLXAnFRGTUC11.EOC
pExternRateCorrection	FLXAnFRGTUC11.ERC

21.3.16 Usage of Data Transfer

A mechanism is implemented to allow storage of FlexRay messages directly into the Local RAM/ Global RAM (user RAM) and have transfers between the FlexRay internal message RAM and the Local RAM/Global RAM and vice versa with minimum CPU support. The data in the Local RAM/ Global RAM should be indexed by data structure pointers located in data pointer tables stored in the Local RAM/Global RAM.

Data transfer from the Local RAM/Global RAM to the FlexRay internal message RAM (input transfer) needs to be initiated by the application. These transfers can be used to configure message buffers or to update transmit data.

A data transfer from the FlexRay internal message RAM to the Local RAM/Global RAM (output transfer) is initiated automatically by a reception into a receive message buffer or FlexRay internal FIFO or by a change in the slot status. It can be initiated also by a specific user transfer request.

The input and output data transfer can be activated independently. When the input data transfer is activated the application should not directly access message buffers using the FlexRay input buffer. When the output data transfer is activated the application should not directly access message buffers using the FlexRay output buffer.

21.3.16.1 Input Data Transfer

When the automatic input data transfer function is enabled, committed input data structures are transferred from the Local RAM/Global RAM to the FlexRay internal message RAM with minimum CPU support.

(1) Activation and deactivation

The input data transfer function should be activated before usage. The activation of the input transfer handler initializes the input queue put index (FLXAnFRITS.IPIDX) and get index (FLXAnFRITS.IGIDX) to 0. Also the interrupt status flags in the FLXAnFRITS register (IQEIS and IQFIS) are set to 0.

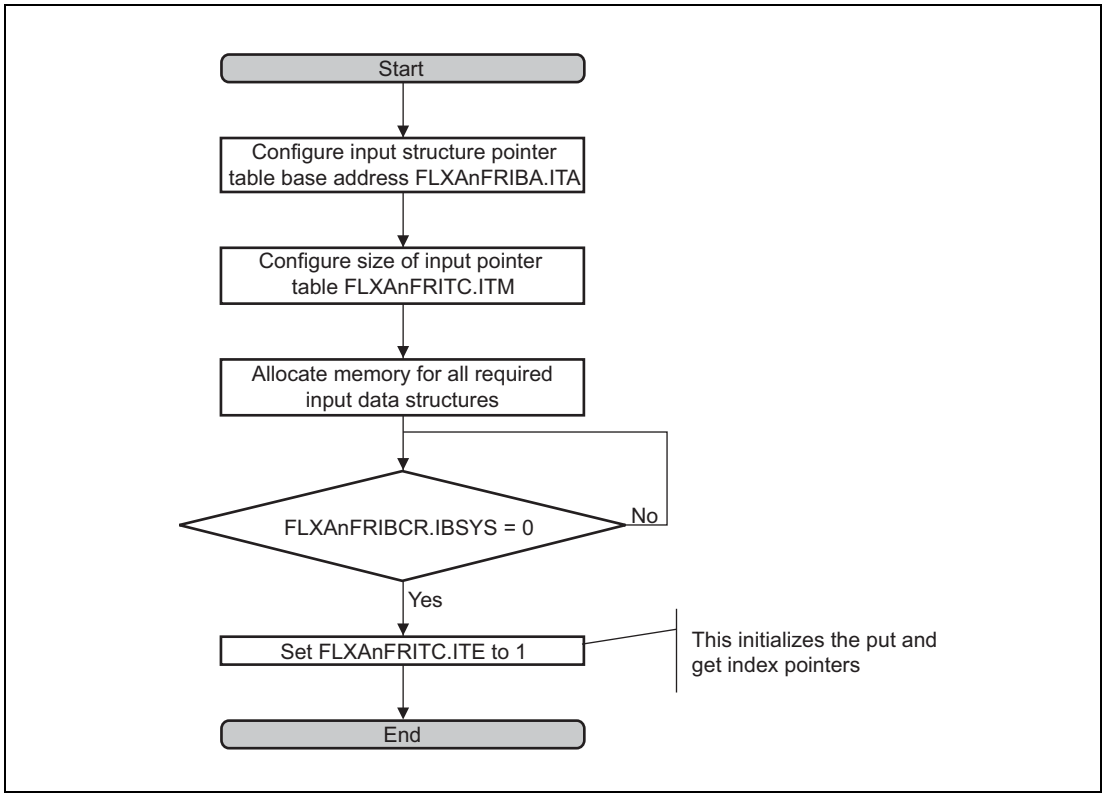


Figure 21.20 Input Transfer Enable Flow

A deactivation request of the input transfer function can be made at any time. The input queue put index and the input queue status are maintained independently from the input transfer function state.

Before the transfer function gets disabled (status indicated by FLXAnFRITS.ITS), user requested input transfers and all committed input transfers will be completed.

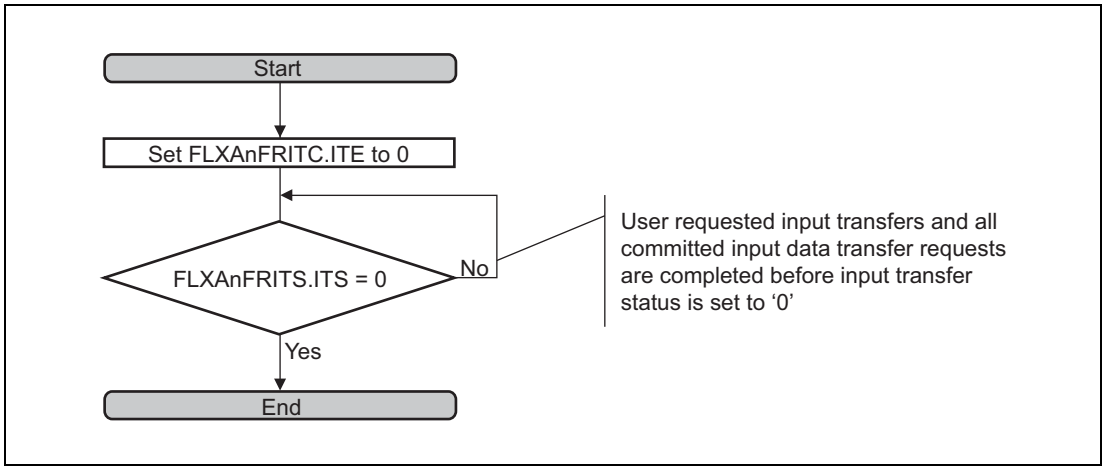


Figure 21.21 Input Transfer Disable Flow

(2) Input data structure

The application has to reserve a location in the Local RAM/Global RAM to provide the content for message buffer configuration (input data structure).

The location of this input data structure needs to be defined by an input data structure pointer also located in the Local RAM/Global RAM.

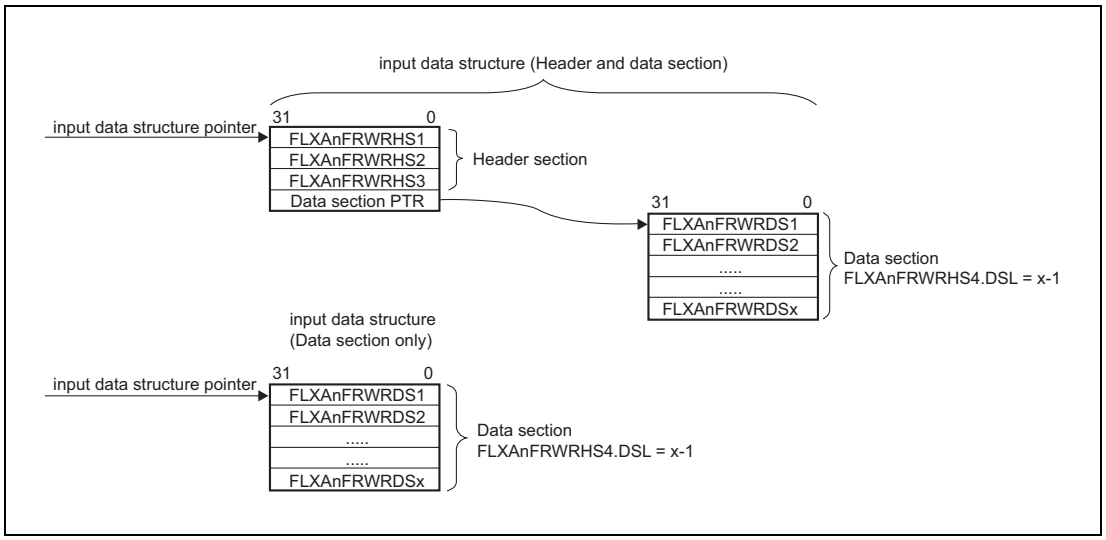


Figure 21.22 Input Data Structure

In general the input data structure consists of two sections, the header and the data section.

The header section consists of FLXAnFRWRHS1, FLXAnFRWRHS2, FLXAnFRWRHS3 and the data section pointer.

For bit alignment and bit function in the header section, see **Section 21.3.13.1, Header Partition**.

Depending on the settings in the control field (FLXAnFRWRHS4) located in the input pointer table, the data structure pointer is a reference to the address of FLXAnFRWRHS1 or FLXAnFRWRDS1. The data structure pointer has to be aligned to a 32 bit address.

If the bit LHS in the address related to FLXAnFRWRHS4 is set to 1 it is required to provide a valid header section. In this case FLXAnFRWRDS1 is the first element of the data structure.

If the bit LHS in the address related to FLXAnFRWRHS4 is set to 0 a header section is not required. In this case FLXAnFRWRDS1 is the first element of the data structure.

If the bit LDS in the address related to FLXAnFRWRHS4 is set to 1 it is required to provide a valid data section. The pointer to the data section is a reference to the address of the first payload long word (FLXAnFRWRDS1) and has to be aligned to a 32 bit address.

If the bit LDS in the address related to FLXAnFRWRHS4 is set to 0 a data section is not required. The data section pointer is not evaluated by the input handler.

The byte order for the FlexRay payload data in the input data structure is determined by FLXAnFROC.BEC. For information about the payload data alignment within the data section refer to **Section 21.3.13.2, Data Partition** and **Section 21.3.17, Byte Alignment**.

The length of the data section and the size to be allocated in the Local RAM/Global RAM depends on the configuration of the bits DSL in the address related to FLXAnFRWRHS4.

For the transfer into the FlexRay core internal message RAM the number of 16 bit words configured by FLXAnFRWRHS2.PLC is used. The application has to ensure, that a proper number of data words is provided in the Local RAM/Global RAM. In case the buffer is configured by FLXAnFRWRHS2.PLC to hold an odd payload length, the application has to write zero to the last 16 bit of the payload section to ensure that the padding data is all zero.

(3) Input pointer table

To transfer data from the input data structures located in the Local RAM/Global RAM to the FlexRay internal message RAM the related input data structure pointer and control field needs to be added to the input pointer table which is located in the Local RAM/Global RAM.

The location of the first element of this table is identified by the input pointer table base address (FLXAnFRIBA.ITA). This base address has to be aligned to a 32 bit address.

The maximum number of input requests that can be queued is defined by the Input queue Table Max register (FLXAnFRITC.ITM).

Each Input pointer table entry requires two long words. The required address range of the input pointer table for the queued transfer requests can be calculated by

Input pointer table size (byte) = (((FLXAnFRITC.ITM + 1) × 2) × 4)

Equation 1

The input pointer entry for the user requested input transfer should be added after the end of the input pointer table.

The pointer table index related to this entry and hence the number to be written to FLXAnFRUIR.UIDX, is FLXAnFRITC.ITM+1. The address in the input pointer table related to the user requested input transfer (user input address) can be calculated by

User input address = FLXAnFRIBA.ITA + Input pointer table size

Equation 2

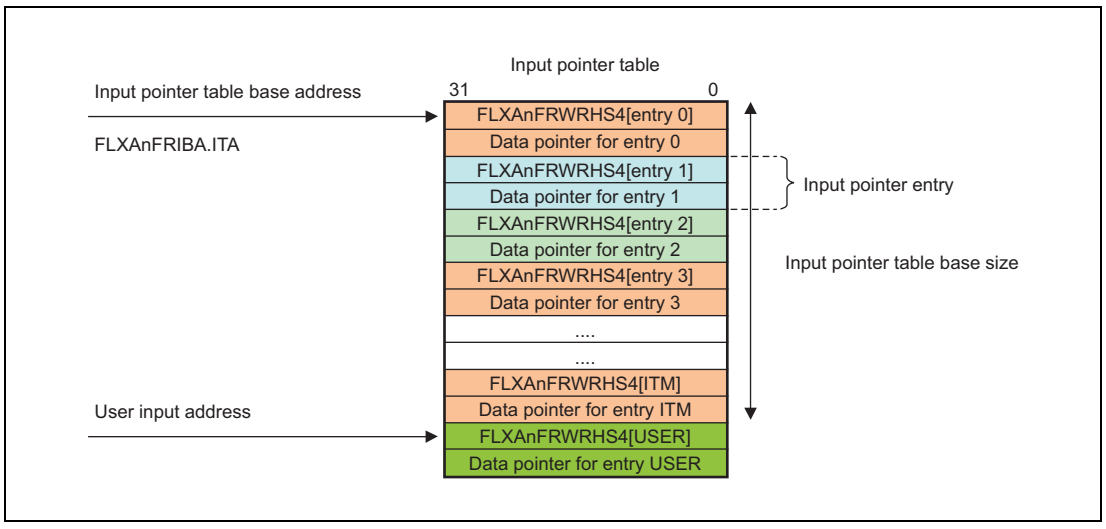


Figure 21.23 Input Pointer Table

The input pointer table holds the control field FLXAnFRWRHS4 and the pointers to the Local RAM/ Global RAM location where the message buffer content (header section and/or data section) is stored.

The application has to write FLXAnFRWRHS4 and the input data structure pointer at the addresses in the input pointer table related to the put index position before a transfer request is initiated.

FLXAnFRWRHS4:

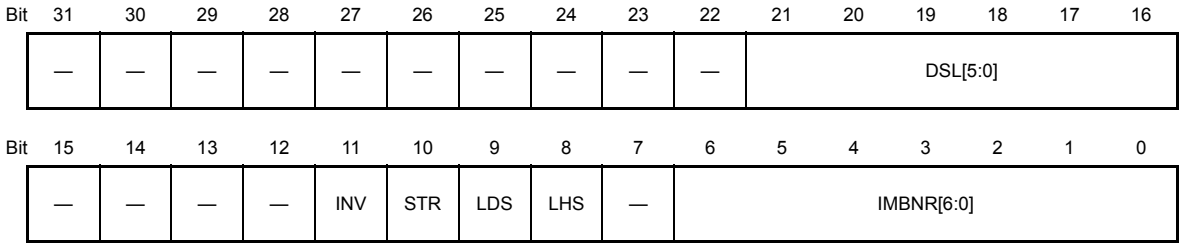


Table 21.112 FLXAnFRWRHS4 Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, an undefined value is read. When writing, always write 0.
21 to 16	DSL[5:0]	Data Section Length Bit Specifies the length of the data section in terms of 32 bit values.
15 to 12	Reserved	When read, an undefined value is read. When writing, always write 0.
11	INV	Invalidate entry Bit 0: The data structure is valid and will be transferred to the FlexRay internal message RAM. 1: The data structure is invalid. FlexRay internal message RAM is not updated using this input pointer entry.
10	STR	Set transmission request Bit 0: The bit FLXAnFRTXRQx.TXR for the message buffer selected by the bits IMBNR is set to 0. No data from this message buffer is transmitted. 1: The bit FLXAnFRTXRQx.TXR for the message buffer selected by the bits IMBNR is set to 1 to release the message buffer for transmission The application should not set the bit STR to 1 for receive buffers.
9	LDS	Load data section Bit 0: No update of data section. 1: Data section for the message buffer selected by the bits IMBNR is updated.
8	LHS	Load header section Bit 0: no update of header section. 1: Header section for the message buffer selected by the bits IMBNR is updated.
7	Reserved	When read, an undefined value is read. When writing, always write 0.
6 to 0	IMBNR[6:0]	Message buffer number to be updated Bit Selects the target message buffer number in the FlexRay internal message RAM for transfer

Note that the LHS bit should not be set for protected message buffers.

The bit LDS defines if the data section of the message buffer selected by the bits IMBNR should be updated.

If LDS is set to 1 (DSL + 1) 32 bit words of payload data are transferred from the Local RAM/Global RAM to the message buffer selected by the bits IMBNR.

If LDS is set to 0 no payload data is transferred from the Local RAM/Global RAM.

Note that the payload transferred is independent from the configured payload length (bits PLC in the address related to FLXAnFRWRHS2).

The bit INV can be used to invalidate a committed data structure. This bit should be only used to cancel the transfer of committed data structures when the input queue is halted (see **Section 21.3.16.1 (5), Halting the input queue**).

When this bit is set to 1 the message buffer number IMBNR is not updated. When the bit is set to 0 the message buffer number IMBNR is updated.

(4) Transfer function of input data structure

To use the input data structure transfer function the input transfer has to be activated (see **Section 21.3.16.1 (1), Activation and deactivation**). The activation process requires the setup of the input pointer table (see **Section 21.3.16.1 (3), Input pointer table**) in order to specify the source location (input data structures) for the data structures to be transferred. When the input transfer gets enabled the get index pointer is initialized to zero.

All FlexRay internal message buffers can be updated using the input transfer queue which is built in the input pointer table. The application has to write the pointer and control field (table entry) to the data structure to be transferred into the input pointer table. For that purpose the application has to maintain a put index for the input pointer table that indicates where the pointer has to be written to.

To commit this table entry to the input handler, the application has to write the target message buffer number to the input queue control register (FLXAnFRIQC.IMBNR). Afterwards the application has to increment the application internal put index.

By writing to the input queue control register the data available flag (FLXAnFRDAi.DA[IMBNR]) is automatically set to 1. The input transfer handler also maintains the put index pointer in the status register (FLXAnFRITS.IPIDX).

In case the input queue gets full (number of queued input transfer requests is equal to the input queue table size) FLXAnFRITS.IQFP and FLXAnFRITS.IQFIS are set to 1. The input queue full condition pending flag (FLXAnFRITS.IQFP) changes from 1 to 0 when there are entries in the input queue available, whereby the input queue full interrupt status flag (FLXAnFRITS.IQFIS) needs to be cleared by the application.

The application cannot make any further write access to the bit IMBNR in the FLXAnFRIQC register as long as the bit IQFP in the FLXAnFRITS register is 1.

In case the input queue gets empty (number of queued input transfer requests changes to zero) FLXAnFRITS.IQEIS is set to 1. The input queue empty interrupt status flag (FLXAnFRITS.IQEIS) needs to be cleared by the application.

The transfer of the input data structures to the FlexRay message RAM is controlled by a get index pointer which is handled inside the FlexRay module and flagged in FLXAnFRITS.IGIDX. Note that the index is referring to the input entry and not the address offset in the input pointer table.

If the input queue is not empty, the transfer handler reads out the input pointer table entry of the transfer queue and starts the transfer of the input data structure from the address the input pointer is referring to. When all required data words are transferred to the FlexRay module, the data available flag for the transferred message buffer number is set to 0 and the get index in the transfer handler is incremented by one.

In case of an invalidated data structure (see **Section 21.3.16.1 (5), Halting the input queue**) no FlexRay internal message buffer is updated and the related data available flag is automatically set to 0. The change of the data available flag can be used to confirm the cancellation a transmit request.

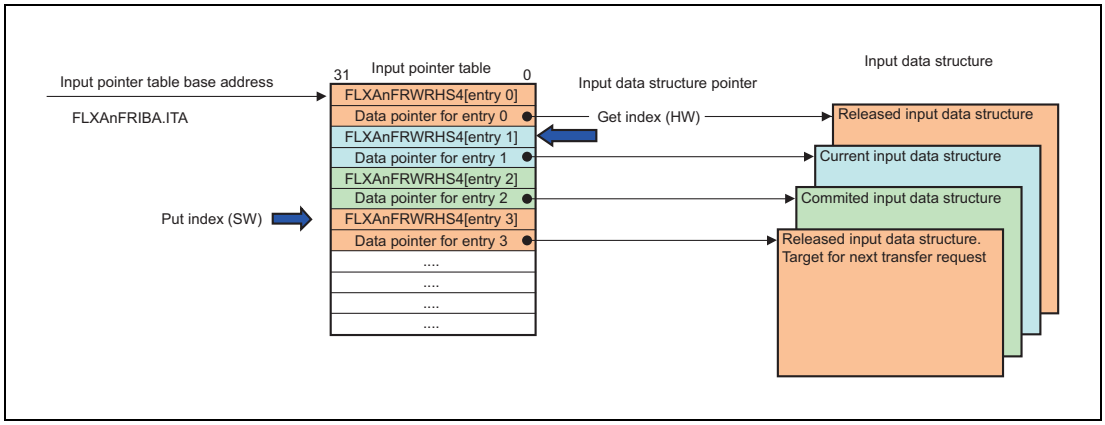


Figure 21.24 Input Pointer Table

Receive message buffers can be also configured using the input data transfer by setting up the required header sections and mark only the header section (FLXAnFRWRHS4.LDS = 0, FLXAnFRWRHS4.LHS = 1) to be updated in the FlexRay module.

(5) Halting the input queue

Committed data structures cannot be removed, but can be invalidated or updated when the input queue is halted.

To cancel data structures already committed to the input queue, the queue can be halted by writing a 1 to FLXAnFRITC.IQHR.

After the ongoing input transfer has been completed the queue is halted and FLXAnFRITS.IQH changes from 0 to 1.

To invalidate an entry of the input queue FLXAnFRWRHS4.INV has to be set to 1. All other bits in FWRHS4 should be unchanged.

Following flow shall be used to analyze whether a committed message has been already transferred to the FlexRay internal message RAM or not.

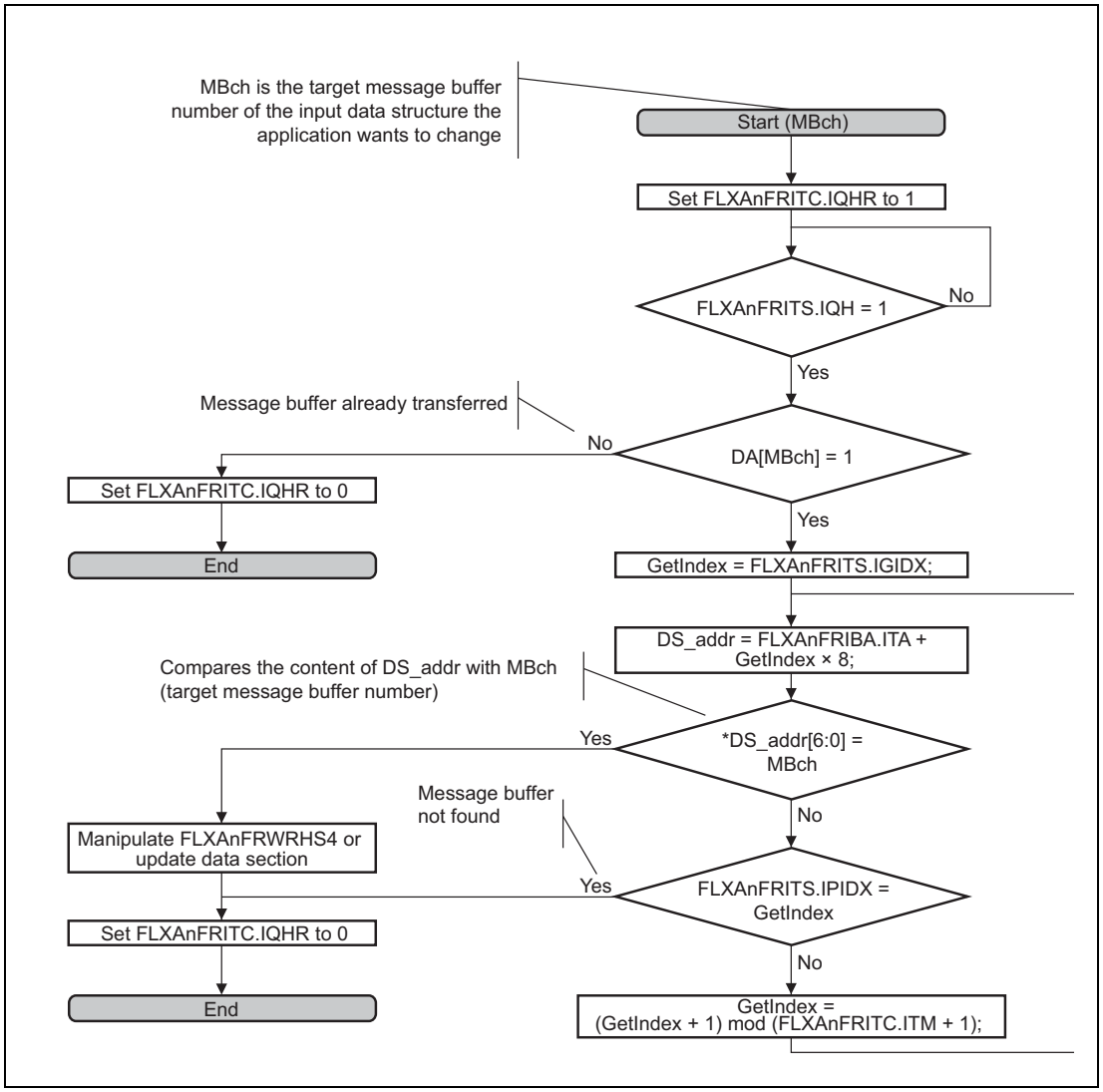


Figure 21.25 Input Table Analysis

In case the message buffer was already transferred to the FlexRay internal message RAM, the user input transfer request can be used to bypass the actual queue and update the required message buffer (see **Section 21.3.16.1 (6), Transfer function of user requested input transfers**).

(6) Transfer function of user requested input transfers

To use this function the input transfer has to be activated (see **Section 21.3.16.1 (1), Activation and deactivation**).

The application is capable, by using FLXAnFRUIR.UIDX, to request a transfer of an input data structure. The user input transfer request is serviced first.

The application has to write the pointer and control field (table entry) to the data structure to be transferred into the input pointer table. The table entry for the user input transfer request should be added after the end of the input pointer table (see **Section 21.3.16.1 (3), Input pointer table**).

To commit this table entry to the input handler, the application has to write the index (FLXAnFRITC.ITM+1) to the user input transfer request register (FLXAnFRUIR.UIDX).

By writing to the user input transfer request register, the user input transfer request pending flag (FLXAnFRITS.UIRP) is automatically set to 1.

As long this flag is 1 the application should not make any further user input transfer requests.

The user input transfer request pending flag (FLXAnFRITS.UIRP) changes from 1 to 0 when the requested input transfer is completed. As next the pending transfers are processed.

21.3.16.2 Output Data Transfer

When the output data transfer function is enabled, received messages (either in dedicated message buffers or in the FlexRay receive FIFO) are transferred to the Local RAM/Global RAM by the output data handler. The output data handler can also transfer the message buffer content to the Local RAM/ Global RAM on application request. When enabled the output handler is also capable to initiate a transfer when the message buffer status has changed.

(1) Activation and deactivation

The output data transfer function should be activated before usage. The activation of the output transfer hander will initialize the FIFO put and get index pointer and FIFO fill level (FLXAnFROTS.FGIDX and FLXAnFROTS.FFL) to zero, set the bits FLXAnFROTS.FDA, FLXAnFROTS.OWP, FLXAnFROTS.FWP and FLXAnFROTS.UORP to 0. Also the interrupt status flags FLXAnFROTS.OTIS, FLXAnFROTS.FIS, FLXAnFROTS.OWIS and FLXAnFROTS.FWIS) are set to 0.

The activation has no influence to the data available flags (FLXAnFRDAi.DA) which are related to the dedicated buffers; these flags have to be cleared by the application.

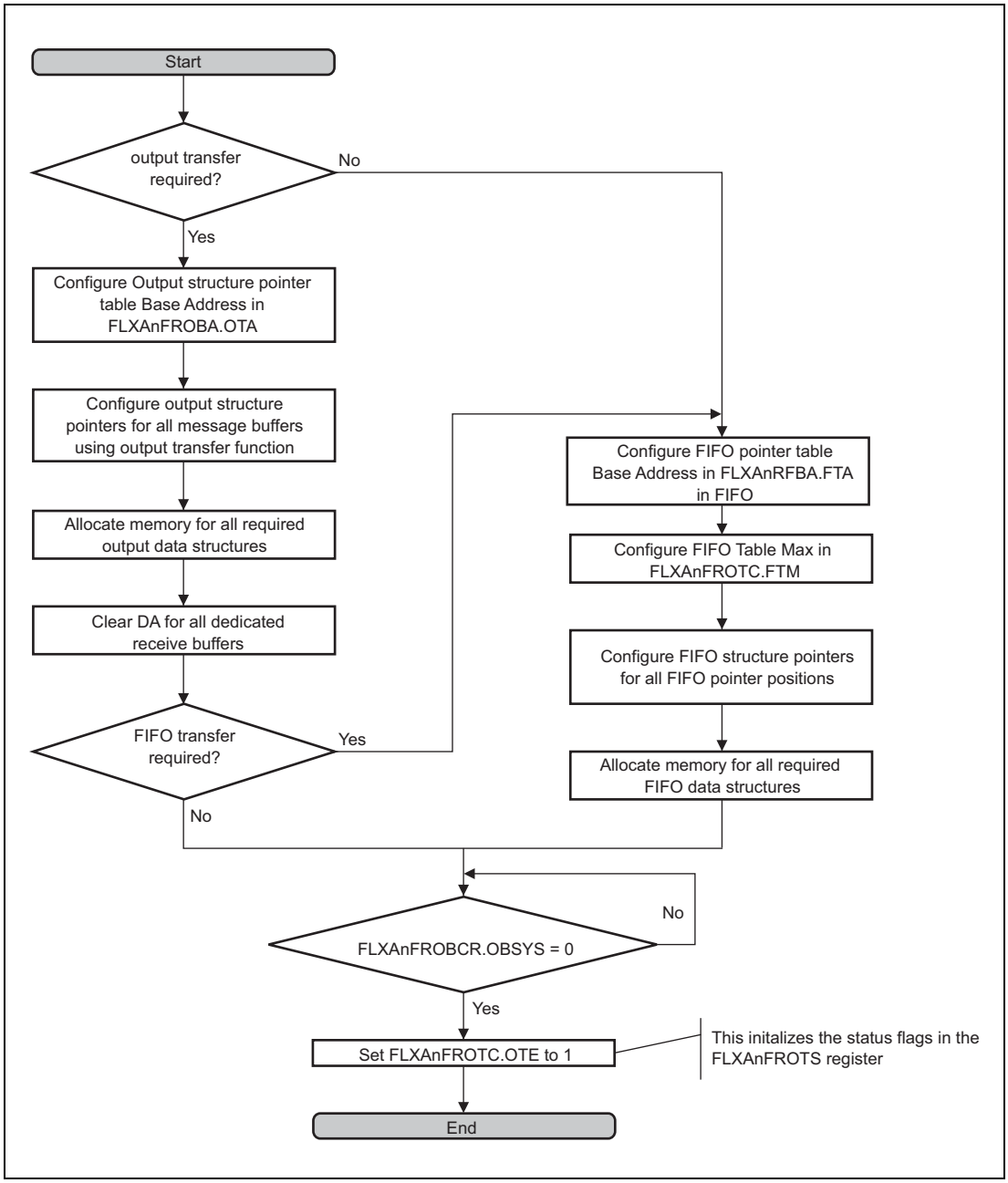


Figure 21.26 Output Transfer Enable Flow

A deactivation request of the output data transfer function can be made at any time. An ongoing transfer will be completed and the completion of this transfer will be flagged. During this time FLXAnFROTS.OTS remains 1.

When FLXAnFROTS.OTS changes from 1 to 0, the output transfer function is deactivated. The data available status flags and the FIFO get index are still maintained when the output transfer function is disabled.

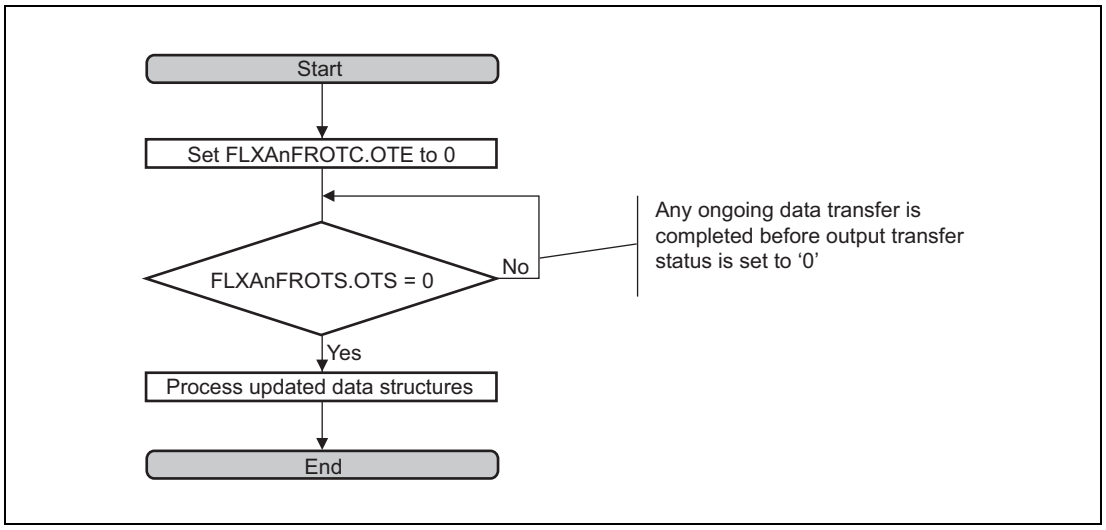


Figure 21.27 Output Transfer Disable Flow

(2) Output transfer data structure

The data in the Local RAM/Global RAM is stored in an output data structure. The location of the output data structures are determined by output data structure pointers also located in the Local RAM/ Global RAM. The output data structure and indexing is visualized in **Figure 21.28**.

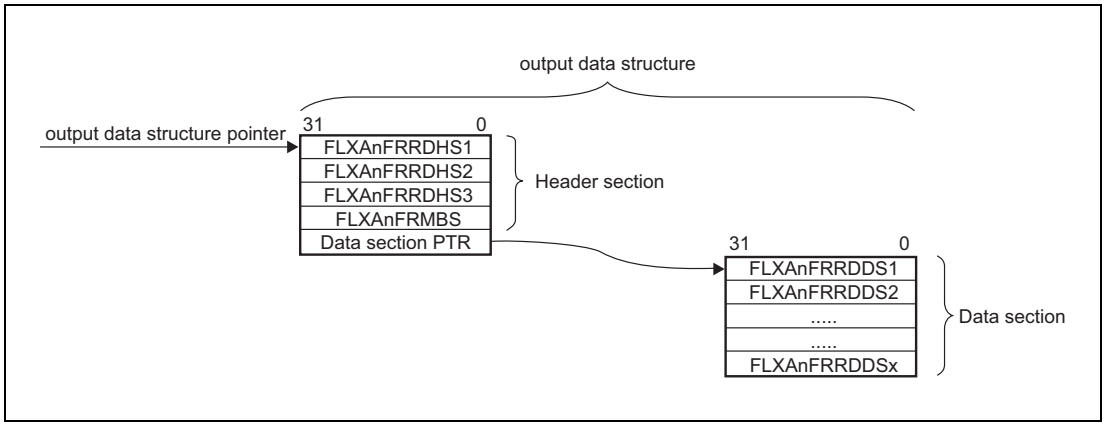


Figure 21.28 Output Data Structure

The output data structure consists of two sections, the header and data section. The header section consists of FLXAnFRRDHS1, FLXAnFRRDHS2, FLXAnFRRDHS3, FLXAnFRMBS and the data section pointer. FLXAnFRRDHS1 is the first element of the structure and has to be aligned to a 32 bit address. The data structure pointer is a reference to the address of FLXAnFRRDHS1. For information about the bit alignment and bit function within the header section refer to **Section 21.3.13.1, Header Partition**.

FLXAnFRRDDS1 is the first element of the data section. The data section pointer is a reference to the address of FLXAnFRRDDS1 and has to be aligned to a 32 bit address. The byte order for the FlexRay payload data in the output data structure is determined by the bit BEC in the FLXAnFROC register. For information about the payload data alignment within the data section refer to **Section 21.3.13.2, Data Partition**.

The length of the data section as well as the total structure size to be allocated in the Local RAM/ Global RAM depends on the configured payload length (FLXAnFRRDHS2.PLC) of the related message buffer. In case the configured payload length is an odd number of words or the received payload length (FLXAnFRRDHS2.PLR) is smaller than the configured payload length, the remaining data words in the Local RAM/Global RAM are unused and cannot be used by the application.

The output data structure is identical for all three kinds of output transfers. In case only the header section is transferred the data section pointer is not evaluated by the output handler and the data section remains unchanged.

(3) Output pointer table

For the output data transfer function the application needs to set up an output pointer table in the Local RAM/Global RAM. The location of the first element of this table should be programmed into the output pointer table base address (FLXAnFROBA.OTA). This base address has to be aligned to a 32 bit address.

The size of the output pointer table is defined by the maximum of: the last configured dedicated message buffer and the highest message buffer number which will be used for the user output transfer request.

The output pointer table holds pointers (output data structure pointers) to the Local RAM/Global RAM location where a memory space is reserved for the target message buffer content (header section and data section).

There is a fixed linear relationship between the address of the entries in the output pointer table and the number of the related message buffers (see **Figure 21.29**): the output pointer table starts with the entry for message buffer number 0 at the address configured in FLXAnFROBA.OTA and continues in ascending order for each following message buffer number, by 32 bit aligned address (e.g. message buffer 1 at address OTA + 4, message buffer 2 at address OTA + 8, etc.) for all possible message buffers.

When a set ND bit is the only transfer condition (FLXAnFROTC.OTCS is set to 0) only message buffers configured as a dedicated receive buffer or that will be used for user output transfer requests need have valid pointer entries.

When a set ND bit or a set MBC bit is the transfer condition (FLXAnFROTC.OTCS is set to 1) all dedicated receive buffer and dedicated transmit buffers need to have valid pointer entries.

(4) FIFO output pointer table

The FlexRay module internal FIFO can be extended by a queued buffer structure in the Local RAM/ Global RAM.

If the FlexRay module internal FIFO is used the application needs to setup the FIFO output pointer table. The location of the first element of this table is identified by the FIFO pointer table base address (FLXAnFRFBA.FTA). This base address has to be aligned to a 32 bit address.

The size of the FIFO pointer table and hence the maximum number of messages that can be added to the queue, is defined by FIFO Table Max (FLXAnFROTC.FTM).

The FIFO pointer table holds pointers (output data structure pointers) to the Local RAM/Global RAM location where a memory space is reserved the target message buffer content (header section and data section). For each table entry a data pointer shall be configured in this table.

(5) Transfer function of dedicated message buffers

To use this transfer function the output transfer has to be activated (see **Section 21.3.16.2 (1), Activation and deactivation**). The activation process requires to setup the output pointer table (see **Section 21.3.16.2 (3), Output pointer table**) in order to specify the destination location (output data structures) for the data to transfer. **Figure 21.29** shows how the output pointer table references the output data structures.

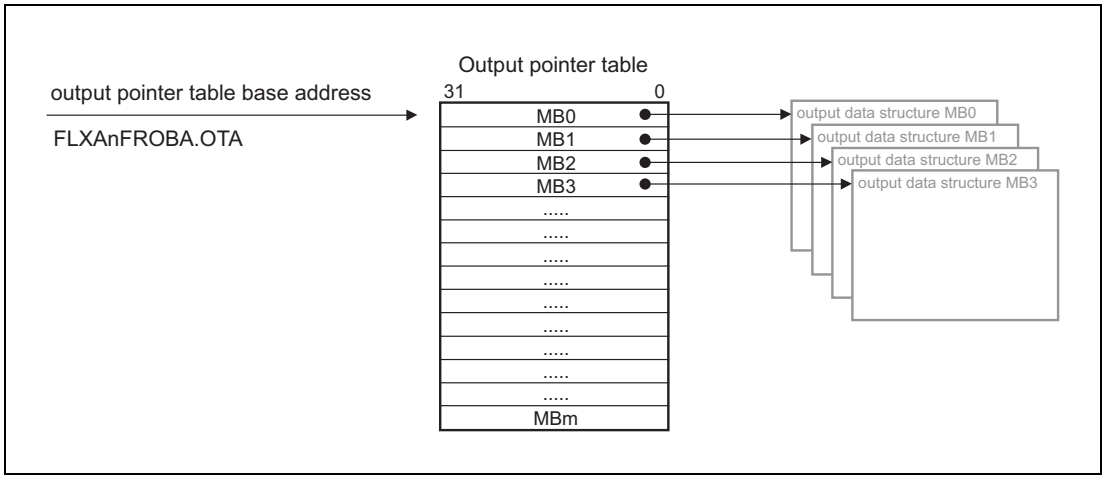


Figure 21.29 Output Data Structure and Indexing

With FLXAnFROTC.OTCS the output transfer condition can be selected between the 'New data only mode' and the 'New data and status changed mode'.

In the 'New data only mode' an output data transfer is initiated when a valid FlexRay data frame has been stored into a dedicated receive buffer which causes the related ND flag in the FLXAnFRNDAT register to set. The ND flag in the FLXAnFRNDAT register is automatically set to 0 during the transfer procedure. The header section is also transferred and hence the MBC flag in the FLXAnFRMBSC register is set to 0.

In the 'New data and status changed mode' an output data transfer is initiated as described in the 'New data only mode'. In addition an output data transfer is initiated when only the message buffer status has been changed which causes the related MBC flag in the FLXAnFRMBSC register to be set. In this case only the header section is transferred. The MBC flag in the FLXAnFRMBSC register is automatically set to 0 during the transfer procedure.

After transferring the message buffer data from the FlexRay internal message RAM to the output data structure the corresponding data available flag in the FLXAnFRDAi (i = 0 to 3) registers is set to 1. The update of the output data structure is also flagged by the setting of the output transfer interrupt status flag (FLXAnFROTS.OTIS).

As long as the data available flag remains 1 the corresponding output data structure will not be updated.

In the case

- the data available flag is 1 and a valid received message was stored or
- when FLXAnFROTC.OTCS is 1 and the message buffer status was updated,

the output transfer warning interrupt flag (FLXAnFROTS.OWIS) is set to 1 notifying the application that new data is available but the output data structure transfer cannot be processed. In addition FLXAnFROTS.OWP is set to 1 that continuously flags that status of the output transfer warning condition.

If a valid receive message in the FlexRay internal message RAM is overwritten by an additional receive message, the message lost flag (FLXAnFRMBS.MLST) is set to 1. This flag can be evaluated after the message buffer has been transferred into an output data structure.

Following sections are giving a guidance how output data structures can be handled.

(a) Data section copy method

One option is to copy the information from the output data structure to a different location of the Local RAM/Global RAM and then release the output data structure by clearing the related data available flag. The application should use the copied information for further processing.

(b) Data structure pointer method

A different option is to modify the output data structure pointer in the output pointer table and to release the output data structure by clearing the related data available flag. The changed output data pointer should refer to a free data structure. The application should use the old data structure for further processing.

(c) Data section pointer method

A third option is to modify the data section pointer in the output data structure and to release the output data structure by clearing the related data available flag. The changed data section pointer should refer to a free memory area. The application should use the old data section for further processing by forwarding the data section pointer.

(6) Transfer function of FIFO message buffers

To use this buffer transfer function the output transfer has to be activated (see **Section 21.3.16.2 (1), Activation and deactivation**). The activation process requires the set up of the FIFO pointer table (see **Section 21.3.16.2 (4), FIFO output pointer table**) in order to specify a location in the Local RAM/Global RAM reserved for the storage of the required output data structures.

A FIFO data transfer is initiated when a valid FlexRay data frame has been stored in the FlexRay internal FIFO.

After transfers from the internal FIFO to the output data structure, the FIFO interrupt status flag (FLXAnFROTS.FIS) and FLXAnFROTS.FDA are set to 1. The bit FLXAnFROTS.FIS can be used as an interrupt source. The bit FLXAnFROTS.FDA indicates that the FIFO is not empty.

Up to FLXAnFROTS.FTM output structures configured in the FLXAnFROTC register can be queued.

The transfer to the extended FIFO buffer structure is controlled by index pointers. This put index is controlled by the FIFO transfer handler and is incremented after transferring a message to the output data structure.

The FIFO reception handler also maintains a get index which is flagged in FLXAnFROTS.FGIDX. The value of this get index is known by the application by either reading the status or maintaining a software variable. The get index (the value after a reset is zero) is incremented by one when the application releases the oldest entry of the FIFO queue by writing 1 to FLXAnFROTS.FDA. By comparing the put index and the get index the FIFO handler knows about the current fill level of the queued buffer structure.

The current FIFO fill level is flagged in FLXAnFROTS.FFL. When FLXAnFROTS.FDA is 1, there is at least one entry in the FIFO queue.

In case the queued buffer structure in the Local RAM/Global RAM is full (FLXAnFROTS.FFL = FLXAnFROTC.FTM+1), no further transfers are initialized, new messages remain in the FlexRay internal FIFO and the FIFO transfer warning interrupt status flag (FLXAnFROTS.FWIS) is set to 1.

In case the FlexRay internal FIFO structure becomes full, messages in the FlexRay internal FIFO structure may get overwritten. The related status flags and configuration registers of the FlexRay core module can be used to generate desired warning notifications.

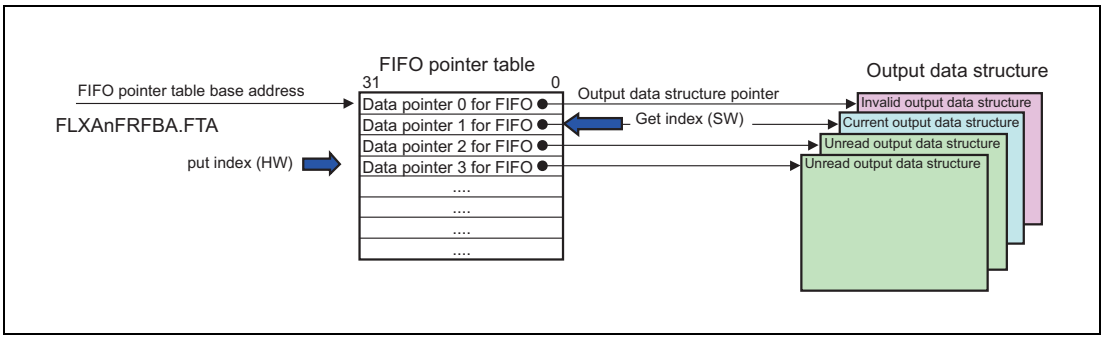


Figure 21.30 FIFO Pointer Table

(7) Transfer function of user output transfer requests

To use this transfer function the output transfer has to be activated (see **Section 21.3.16.2 (1), Activation and deactivation**). The activation process requires to setup the output pointer table (see **Section 21.3.16.2 (3), Output pointer table**) in order to specify the location in the Local RAM/ Global RAM reserved for the transfer of the data (output data structures).

The application is capable, by using FLXAnFRUOR.UMBNR, to request a transfer of dedicated message buffer to an output data structure. Except in CONFIG state, message buffers which are part of the FlexRay internal FIFO should not be requested.

The header section is always transferred to the output data structure. The transfer of the data section can be enabled by setting FLXAnFRUOR.URDS to 1. The selected message buffer content is stored in the output data structure location determined by the pointers in the output pointer table.

The data available status and transfer blocking by the DA bits in the FLXAnFRDAi register is also used for the user requested transfers. Therefore the DA bit in the FLXAnFRDAi register related to the requested buffer number (FLXAnFRUOR.UMBNR) should be released before making the transfer request.

After writing to FLXAnFRUOR.UMBNR, the bit FLXAnFROTS.UORP is set to 1 to indicate that there is a pending user transfer request. When the transfer has been processed the bit FLXAnFROTS.UORP is set to 0, the bit FLXAnFROTS.OTIS is set to 1 and the DA bit in the FLXAnFRDAi register related to the requested buffer number (FLXAnFRUOR.UMBNR) is set to 1.

User output transfer requests cannot be queued. The application should check the bit FLXAnFROTS.UORP before writing to FLXAnFRUOR.UMBNR.

User output transfer requests should not be made for message buffers which are pending in the input transfer queue.

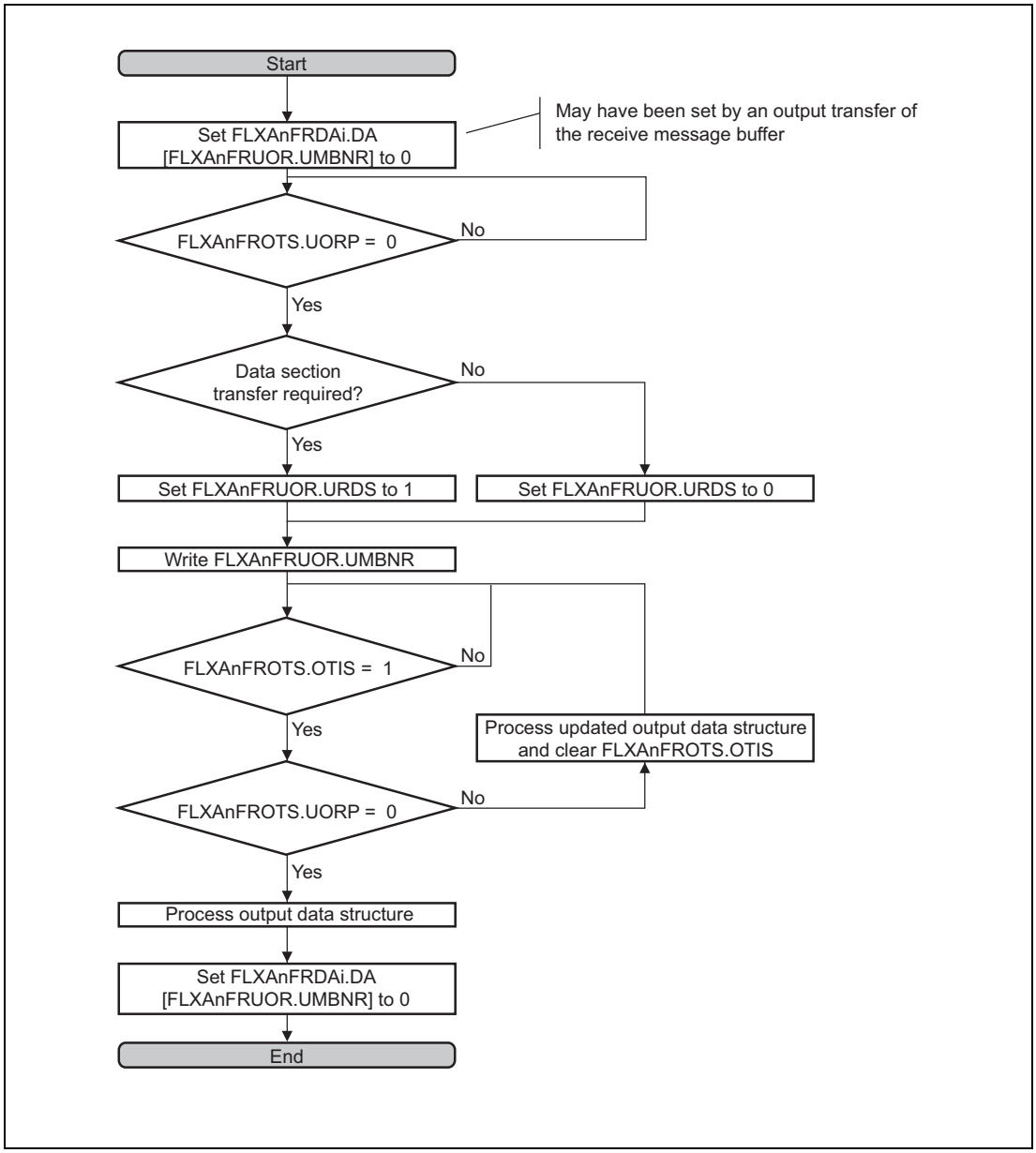


Figure 21.31 User Output Transfer Request Flow

Note that it may be possible that the data structure addressed by a user request is being updated due to a receive message buffer update (which causes the bit DA in the FLXAnFRDAi register being set). This set DA flags inhibits the user output transfer request. Therefore polling FLXAnFROTS.UORP is not a secure method to identify when the transfer of a requested message buffer has been completed. The bits FLXAnFROTS.OTIS or the DA bit in the FLXAnFRDAi register can be used instead. The exact flow depends on the software architecture.

21.3.16.3 Data Structure Transfer Scheduling

Cyclically the different types of transfer requests are checked. In order to guarantee a certain transfer time the different types of transfers have different priorities.

Use requested input transfers have highest priority followed by the transfer of data structures committed into the active input transfer queue. No new output transfer will be started as long as there is a pending input transfer request.

The three output transfer request types are checked in a specific order:

(1) All dedicated message buffers in ascending order

When FLXAnFROTC.OTCS is set to 0, set flags in the FLXAnFRNDATn register are causing a transfer of the message buffer to the output data structure if the destination area is free (DA bit in the FLXAnFRDAi register is 0).

When FLXAnFROTC.OTCS is set to 1, set flags in the FLXAnFRNDATn register or set flags in the FLXAnFRMBSCn register are causing a transfer of the message buffer to the output data structure if the destination area is free (DA bit in the FLXAnFRDAi register is 0).

(2) FlexRay internal FIFO

When the FlexRay internal FIFO is not empty and there is a free destination area, one FIFO message is transferred into the output data structure addressed by the FIFO pointer table.

(3) User output request

If there is a pending user output transfer request, one message buffer is transferred into the corresponding output data structure.

The check sequence is suspended when an input transfer occurs.

21.3.16.4 Behavior in Case of Data Transfer Access Error

The memory areas accessed by the data transfer function may be protected by a memory protection unit (MPU). When the MPU flags an access to a protected address caused by an input or output transfer, an access error event is generated and the related bit in the FLXAnFRAES register is set.

The ongoing transfer is immediately terminated but succeeding transfers are processed and may generate further access errors. Any following access errors are only flagged in FLXAnFRAES.MAE. The other status flags are not updated.

(1) Access error during input transfer

When an access error occurs during an input transfer:

- The ongoing transfer is immediately terminated. The FlexRay internal message RAM will not be updated
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.IAE is set to 1
- The input pointer table index is flagged in FLXAnFRAES.EIDX
- In case of an normal input transfer, the related DA flag in the FLXAnFRDAi register is set to 0
- In case of an user input transfer, request FLXAnFRITS.UIRP is set to 0

With the given status information the application is able to identify and correct the faulty data structure.

In addition the application needs to clear the input access error flag (FLXAnFRAES.IAE).

(2) Access error during output transfer

When an access error occurs during an output transfer:

- The ongoing transfer is immediately terminated but the update of the data structure may have started.
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.OAE is set to 1
- The output pointer table index is flagged in FLXAnFRAES.EIDX
- In case of an normal output transfer, the related DA flag in the FLXAnFRDAi register remains 0 and no output transfer, interrupt is generated
- In case of an user output transfer request FLXAnFROTS.UORP is set to 0

With the given status information the application is able to identify and correct the faulty data structure. The data structure in the Local RAM/Global RAM cannot be treated as valid.

In addition the application needs to clear the output access error flag (FLXAnFRAES.OAE).

The FlexRay module internal transfer of the message buffer is completed before the Local RAM/ Global RAM access error is detected. The output transfer will not be re-initiated. To avoid loss of data, the application can perform a user output transfer request of this message buffer to a correct Local RAM/Global RAM location.

(3) Access error during FIFO transfer

When an access error occurs during an FIFO transfer:

- The ongoing transfer is immediately terminated
- The address, the FlexRay module wanted to access to, is captured in the FLXAnFRAEA register
- FLXAnFRAES.FAE is set to 1
- The FIFO pointer table index is flagged in FLXAnFRAES.EIDX
- The FIFO index pointer are not changed and hence the FIFO status flags are unchanged

With the given status information the application is able to identify and correct the faulty data structure.

In addition the application needs to clear the FIFO access error flag (FLXAnFRAES.FAE).

The data in the Local RAM/Global RAM cannot be treated as valid and is not released to the application. The message cannot be recovered.

21.3.16.5 Behaviors in Case of RAM Read Errors

The FlexRay internal message RAM has an ECC checking mechanism. In case an uncorrectable RAM read error occurs, the application has to analyze the status in the FLXAnFRMHDS register and react as described in **Section 21.3.16.3, Data Structure Transfer Scheduling**. The input and output transfer handler reacts also on these errors detected in the message RAM when the error is related to an active transfer.

In addition, the Temporary Buffer RAM A and Temporary Buffer RAM B have an ECC checking mechanism as well. An uncorrectable RAM read errors does not impact the data transfer functionality but have to be handled as described in **Section 21.3.13.1 (4), Message Buffer Status FLXAnFRMBS (word 3)**.

In all cases, data causing a read error is never transferred to the Local RAM/Global RAM. If there is no recovery available in the application, the message is lost.

(1) Read error during transfer from TBF to MBF

This internal transfer is done for each valid FlexRay message received.

A read error can only occur when reading the header section in the FlexRay Message RAM (see read error flags in FLXAnFRMHDS). In this case, the message buffer needs to be re-configured.

For dedicated receive message buffers, the related FLXAnFRNDAT flag will not get set. Consequently the affected message buffer will not be transferred to the output data structure.

For the FlexRay internal FIFO buffers, the FLXAnFRNDAT flag is not set but the FlexRay internal FIFO put index is incremented. Due to this, a transfer procedure from the FlexRay internal FIFO buffer to the output buffer is started. However, if the read error is still present in the header section, updating of the output data structure will not start (see **Section 21.3.16.5 (2), Read error during transfer from MBF to OBF**); thus the data in the Local RAM/Global RAM remains correct.

Note that the correction or any other reconfiguration of FIFO related to message buffers while there are pending FIFO transfers may result in incorrect data in the Local RAM/Global RAM. It is strongly recommended to deactivate the output data transfer before starting the reconfiguration and flush the FlexRay internal FIFO before reactivation of the output data transfer.

(2) Read error during transfer from MBF to OBF

This internal transfer is done for every output data transfer (dedicated reception, FIFO, user requested).

A read error can occur in the header and data section (see read error flags in FLXAnFRMHDS). In both cases the message gets lost. If the error is located in the header section, the message buffer needs to be re-configured. If the error is located in the data section, the error is corrected with the next data section update.

When a read error occurs during the transfer from the message RAM to the output buffer, the output data structure will not be updated and the data available will not be set to 1. The FIFO put index and the FIFO fill level are not changed also.

In case of user output transfer request, FLXAnFROTS.UORP is set to 0 even if there was no update of the output data structure.

(3) Read error during transfer from IBF to MBF

This internal transfer is done for every input data transfer.

A read error can occur only when there is no update of the header section requested (the bit LHS in FLXAnFRWRHS4 is set to 0) due to the reading of the header section from the message RAM (see read error flags in FLXAnFRMHDS). In this case, the message buffer needs to be re-configured.

When a read error occurs during the input data transfer, the actually transferred message in the input queue gets lost.

(4) Message RAM read errors

Read errors when reading the header section are flagged in the FLXAnFRMHDS register.

Depending on the buffer type and set buffer protection, a reconfiguration of the message buffer may not be possible.

The input transfer function cannot be used to reconfigure a locked message buffer using the method described in **Section 21.3.13.4 (3), Temporary Unlocking of Header Section**.

Before reconfiguring a locked buffer, the user should disable the input transfer function and the output transfer function.

21.3.16.6 Data Transfer Timings

The transfer timing between the Local RAM/Global RAM and the FlexRay internal message buffers is composed of two elements. The first element is the FlexRay core internal transfer time between the interface buffer and the message RAM. The second element is the transfer time required between the Local RAM/Global RAM and the interface buffer.

The concept of the input- and output handler allows only one active transfer between the Local RAM/ Global RAM and input buffer as well as only one active transfer between the input buffer and the message RAM. In case two requests are made at the same time, the input transfer task is handled at first.

Due to this concept constrains the calculation of transfer times does not consider a parallel transfer but the worst case scenario whereby an output transfer is started only after all input transfers are completed.

(1) FlexRay core internal transfer time

The number of clock cycles required for a transfer is given in FlexRay core module clock cycles (bus clock). The transfer time differs between the input and output transfer.

The time required for a transfer between the interface buffer and the FlexRay internal message RAM is given by [ERAY_ADD]

$$cycles_{req\_MHD[PL]} = (numberOfConcurrentTasks) \times (setupTime + (DataWords_{req[PL]}) + 75)$$

Equation 3

The number of data words (32 bit) can be calculated by

$$DataWords_{req[PL]} = 4 + floor\left(\frac{PLC + 1}{2}\right)$$

Equation 4

Whereby 4 is the number of header section words (32 bit) which needs to be always accessed even if the header section is not updated.

As there can be in total three parallel tasks in the FlexRay internal message handler but the number of concurrent tasks is unknown, not only for a worst case consideration the number of concurrent tasks must be set to 3. Beside the setup time and the number of words to be transferred a message handler internal delay time of 75 bus clock cycles must be added. The maximum number of payload to be transferred is in worst case 64.

The worst case for an input transfer is different from the worst case for an output transfer.

For an input transfer the setup time required is 6 bus clock cycles and therefore

$$cycles_{req\_MHD\_input[64]} = 3 \times (6 + 68 + 75) = 447$$

Equation 5 Message hander internal input transfer time (worst case)

For an output transfer the setup time required is 2 bus clock cycles and therefore

$$cycles_{req\_MHD\_output[64]} = 3 \times (2 + 68 + 75) = 435$$

Equation 6 Message handler internal output transfer time (worst case)

(2) System bus transfer time

The number of bus clock cycles required for a transfer is given in bus clock cycles and is identical for all kind of transfers between the Local RAM/Global RAM and the FlexRay module.

The time required by the transfer handler for a transfer between the Local RAM/Global RAM and the FlexRay interface buffer is given by

$$cycles_{req\_SB[PL]} = throughput \times (systemTime + DataWords_{req[PL]}) + setupTime$$

Equation 7 General system bus transfer time formula

The number of payload words (32 bit) can be calculated by using Equation 4.

The throughput depends on the utilization of the system bus and depends on the application. However, the data transfer handler has the highest priority for accessing the Local RAM/Global RAM and the data throughput factor can be assumed as XXXX even in worst case scenarios. The system time (read of the data pointer from Local RAM/Global RAM) can be set to 1 in all cases. The transfer setup time can be set to 3 in all cases.

Therefore the system bus transfer time (given in bus clock cycles) is given by

$$cycles_{req\_SB[64]} = throughput \times (2 + 68) + 3 = XXXX \times (2 + 68) + 3$$

Equation 8 System bus transfer time (worst case)

In addition to the system bus transfer time a FlexRay module internal processing time needs to be taken into account. The number of clock cycles required for the internal processing time required is given in FlexRay core module clock cycles (bus clock) and is different between input transfers and output transfers but can be treated as constants.

$$cycles_{req\_THD\_input} = 2 + 4 = 6$$

Equation 9

Whereby 2 cycles are required to initiate the FlexRay module internal transfer from the input buffer to the message RAM. In addition 4 internal processing cycles are required.

$$cycles_{req\_THD\_output} = 3 + 4 = 7$$

Equation 10

Whereby 3 cycles are required to initiate the FlexRay module internal transfer from the input buffer to the message RAM. In addition 4 internal processing cycles are required.

(3) Summary required input transfer time

The worst case time (as a function of the transferred payload) required to transfer data between the Local RAM/Global RAM to the FlexRay internal message buffer can be calculated by

$$t\_input_{[PL]} = t\_chi \times (cycles_{req\_THD\_input} + cycles_{req\_MHD\_input[PL]}) + t\_sys \times (cycles_{req\_SB[PL]})$$
$$t\_input_{[PL]} = t\_chi \times (6 + 3 \times [81 + DataWords_{[PL]}]) + t\_sys \times (throughput \times [2 + DataWords_{[PL]}] + 3)$$

Equation 11 Input transfer time

With

$$t\_sys = (f(sysclk))^{-1}$$

and.

$$t\_chi = (f(clkp2\_flx))^{-1}$$

(4) Summary required output transfer time

The worst case time (as a function of the transferred payload) required to transfer data from the FlexRay internal message buffer to the Local RAM/Global RAM can be calculated by

$$t\_output_{[PL]} = t\_chi \times (cycles_{req\_THD\_output} + cycles_{req\_MHD\_output[PL]}) + t\_sys \times (cycles_{req\_SB[PL]})$$
$$t\_output_{[PL]} = t\_chi \times (7 + 3 \times [77 + DataWords_{[PL]}]) + t\_sys \times (throughput \times [2 + DataWords_{[PL]}] + 3)$$

Equation 12 Output transfer time

With

$$t\_sys = (f(sysclk))^{-1}$$

and.

$$t\_chi = (f(clkp2\_flx))^{-1}$$

21.3.17 Byte Alignment

The alignment of the bytes received by the FlexRay protocol and the alignment of the bytes required by the application may be different. The FlexRay module provides with FLXAnFROC.BEC a byte alignment function to support different byte ordering styles.

Figure 21.32 shows the payload byte alignment in a FlexRay frame.

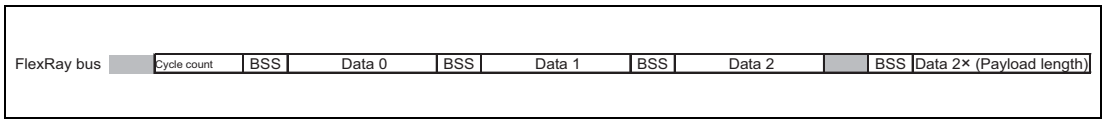


Figure 21.32 Byte Alignment on the FlexRay Bus

21.3.17.1 Little Endian Alignment

When FLXAnFROC.BEC is 0, the byte alignment is set to Little Endian.

(1) FLXAnFRNMVm (m = 1 to 3)

The byte alignment of the NMV bytes is as below.

Bit	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
FLXAnFRNMV1	Data 3	Data 2	Data 1	Data 0
FLXAnFRNMV2	Data 7	Data 6	Data 5	Data 4
FLXAnFRNMV3	Data 11	Data 10	Data 9	Data 8

(2) FLXAnFRWRDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay input buffer and the input data structure is as below.

FLXAnFRWRDSxMD[7:0] = $Data_{4x-4}$

FLXAnFRWRDSxMD[15:8] = $Data_{4x-3}$

FLXAnFRWRDSxMD[23:16] =$Data_{4x-2}$

FLXAnFRWRDSxMD[31:24] = $Data_{4x-1}$

Transmission order on the FlexRay bus is FLXAnFRWRDSx.MD[7:0], FLXAnFRWRDSx.MD [15:8], FLXAnFRWRDSx.MD [23:16], FLXAnFRWRDSx.MD [31:24] with the most significant bit (MSB) transmitted first.

(3) FLXAnFRRDDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay output buffer and the output data structure is as below.

FLXAnFRRDDSxMD[7:0] = $Data_{4x-4}$

FLXAnFRRDDSxMD[15:8] = $Data_{4x-3}$

FLXAnFRRDDSxMD[23:16] =$Data_{4x-2}$

FLXAnFRRDDSxMD[31:24] = $Data_{4x-1}$

Reception order on the FlexRay bus is FLXAnFRRDDSx.MD[7:0], FLXAnFRRDDSx.MD [15:8], FLXAnFRRDDSx.MD [23:16], FLXAnFRRDDSx.MD [31:24] with the most significant bit (MSB) received first.

21.3.17.2 Big Endian Alignment

When FLXAnFROC.BEC is 1, the byte alignment is set to Big Endian.

(1) FLXAnFRNMVm (m = 1 to 3)

The byte alignment of the NMV bytes is as below.

Bit	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
FLXAnFRNMV1	Data 0	Data 1	Data 2	Data 3
FLXAnFRNMV2	Data 4	Data 5	Data 6	Data 7
FLXAnFRNMV3	Data 8	Data 9	Data 10	Data 11

(2) FLXAnFRWRDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay input buffer and the input data structure is as below.

FLXAnFRWRDSxMD[7:0] = $Data_{4x-1}$

FLXAnFRWRDSxMD[15:8] = $Data_{4x-2}$

FLXAnFRWRDSxMD[23:16] =$Data_{4x-3}$

FLXAnFRWRDSxMD[31:24] = $Data_{4x-4}$

Transmission order on the FlexRay bus is FLXAnFRWRDSx.MD[31:24], FLXAnFRWRDSx.MD[23:16], FLXAnFRWRDSx.MD[15:8], FLXAnFRWRDSx.MD[7:0] with the most significant bit (MSB) transmitted first.

(3) FLXAnFRRDDSx (x = 1 to 64)

The byte alignment for the message payload in the FlexRay output buffer and the output data structure is as below.

FLXAnFRRDDSxMD[7:0] = $Data_{4x-1}$

FLXAnFRRDDSxMD[15:8] = $Data_{4x-2}$

FLXAnFRRDDSxMD[23:16] =$Data_{4x-3}$

FLXAnFRRDDSxMD[31:24] = $Data_{4x-4}$

Reception order on the FlexRay bus is FLXAnFRRDDSx.MD[31:24], FLXAnFRRDDSx.MD[23:16], FLXAnFRRDDSx.MD[15:8], FLXAnFRRDDSx.MD[7:0] with the most significant bit (MSB) received first.

21.4 Detection and Correction of Errors in FlexRay RAM

21.4.1 ECC for the FlexRay RAM

This product incorporates RAM for the FlexRay module and an ECC circuit for this RAM.

Table 21.113 FlexRay RAM and ECC Correspondence

Target	ECC Unit Name
Message RAM (MRAM)	ECCFLXA0
Temporary buffer (TBFA)	ECCFLXA0T0
Temporary buffer (TBFB)	ECCFLXA0T1

Table 21.114 gives an outline of the ECC functions for the FlexRay RAM.

Table 21.114 List of the ECC Functions for the FlexRay RAM

Item	Outline of Functions
ECC error detection/ correction	The RAM is checked for ECC errors. The following options are selectable. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection The ECC error detection/correction can be disabled by using through mode. With the initial settings, error detection/correction is enabled.
Error notification	When an ECC 2-bit error is generated, an interrupt signal is generated. • Error notification can be enabled or disabled when an ECC 2-bit error is detected. In the initial setting, 2-bit error notification is enabled. However, if an interrupt is masked by the FEINTFMSK register, an interrupt processing is not executed.
Error status	Monitoring for the detection of ECC 2-bit errors and for the detection of ECC1-bit errors is available. A bit for clearing the error status is provided.

CAUTION

When ECC error detection/correction is performed, initialize the RAM before it is used.

21.4.2 Interrupt Request

Table 21.115 shows RAM ECC interrupt request of the Flex Ray.

INTECCDFLRAM is the total (logical OR) of the interrupt signals of three ECCs (ECCFLXA0, ECCFLXA0T0, and ECCFLXA0T1) that the FlexRay RAM supports.

Table 21.115 FlexRay ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit interrupt signal	Description	Name	DMA Trigger Number
—	FLXA0 2-bit ECC error interrupt	INTECCDFLRAM	—

21.4.3 List of Registers

FlexRay registers are listed in the following table.

Table 21.116 Registers

Unit	Register	Symbol	Address
ECCFLXA0	ECC Control Register	ECCFLXA0CTL	$FFC72000_H$
ECCFLXA0	ECC Test Mode Control Register	ECCFLXA0TMC	$FFC72004_H$
ECCFLXA0	ECC Redundant Bit Data Control Test Register	ECCFLXA0TRC	$FFC72008_H$
ECCFLXA0	ECC Redundant Bit Input/Output Replacement Buffer Register	ECCFLXA0ERDB	$FFC72008_H$
ECCFLXA0	ECC Encode Test Register	ECCFLXA0ECRD	$FFC72009_H$
ECCFLXA0	ECC 7-Bit Redundant Bit Data Hold Test Register	ECCFLXA0HORD	$FFC7200A_H$
ECCFLXA0	ECC Decode Syndrome Data Register	ECCFLXA0SYND	$FFC7200B_H$
ECCFLXA0	ECC Encode/Decode Input/Output Replacement Test Register	ECCFLXA0TED	$FFC7200C_H$
ECCFLXA0T0	ECC Control Register	ECCFLXA0T0CTL	$FFC73000_H$
ECCFLXA0T0	ECC Test Mode Control Register	ECCFLXA0T0TMC	$FFC73004_H$
ECCFLXA0T0	ECC Redundant Bit Data Control Test Register	ECCFLXA0T0TRC	$FFC73008_H$
ECCFLXA0T0	ECC Redundant Bit Input/Output Replacement Buffer Register	ECCFLXA0T0ERDB	$FFC73008_H$
ECCFLXA0T0	ECC Encode Test Register	ECCFLXA0T0ECRD	$FFC73009_H$
ECCFLXA0T0	ECC 7-Bit Redundant Bit Data Hold Test Register	ECCFLXA0T0HORD	$FFC7300A_H$
ECCFLXA0T0	ECC Decode Syndrome Data Register	ECCFLXA0T0SYND	$FFC7300B_H$
ECCFLXA0T0	ECC Encode/Decode Input/Output Replacement Test Register	ECCFLXA0T0TED	$FFC7300C_H$
ECCFLXA0T1	ECC Control Register	ECCFLXA0T1CTL	$FFC73010_H$
ECCFLXA0T1	ECC Test Mode Control Register	ECCFLXA0T1TMC	$FFC73014_H$
ECCFLXA0T1	ECC Redundant Bit Data Control Test Register	ECCFLXA0T1TRC	$FFC73018_H$
ECCFLXA0T1	ECC Redundant Bit Input/Output Replacement Buffer Register	ECCFLXA0T1ERDB	$FFC73018_H$
ECCFLXA0T1	ECC Encode Test Register	ECCFLXA0T1ECRD	$FFC73019_H$
ECCFLXA0T1	ECC 7-Bit Redundant Bit Data Hold Test Register	ECCFLXA0T1HORD	$FFC7301A_H$
ECCFLXA0T1	ECC Decode Syndrome Data Register	ECCFLXA0T1SYND	$FFC7301B_H$
ECCFLXA0T1	ECC Encode/Decode Input/Output Replacement Test Register	ECCFLXA0T1TED	$FFC7301C_H$

21.4.4 ECCFLXA0CTL/ECCFLXA0T0CTL/ECCFLXA0T1CTL — FlexRay ECC Control Register

The ECCFLXA0CTL/ECCFLXA0T0CTL/ECCFLXA0T1CTL register controls the mode of the ECC and the status for FlexRay.

Bits 7, 5, and 4 should be set (written) while the FlexRay operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be 01_B.

Access: This register can be read or written in 16-bit units.

Address: See **Table 21.116, Registers**.

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	EMCA0	—	—	—	ECER2C	ECER1C	—	ECTHM	—	EC1ECP	EC2EDIC	—	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	—
R/W	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R	R/W	R	R/W	R/W	R	R	R	R

Note 1. These bits are always read as 0.

Table 21.117 ECCFLXA0CTL/ECCFLXA0T0CTL/ECCFLXA0T1CTL Register Contents (1/2)

Bit Position	Bit Name	Function
15	EMCA1	Access Control Bits 1 and 0 to ECC Mode Selection These bits specify whether updating the ECTHM bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 01_B, writing to bit 7 is enabled.
14	EMCA0	
13 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10	ECER2C	2-Bit ECC Error Detection Flag Clear This bit clears 2-bit error detection flags of ECER2F (bit 2). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between this bit writing and ECER2F bit setting occurs, writing to this bit has a priority.
9	ECER1C	1-Bit ECC Error Detection Correction Accumulation Flag Clear This bit clears 1-bit error detection/correction flags of ECER1F (bit 1). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between this bit writing and ECER1F bit setting occurs, writing to this bit has a priority.
8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTHM	ECC Function through Mode Selection Set this bit to select whether to enable or disable the ECC function. Setting this bit to 1 disables the ECC function. When writing to this bit, 0 and 1 respectively must be written to the EMCA1 and EMCA0 bits at the same time. Set this bit to 1 to disable the ECC function. 0: Passing through mode is disabled (normal operation mode). 1: Passing through mode is enabled. (ECC function disable)
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	EC1ECP	1-Bit Error Correction Enable This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.

Table 21.117 ECCFLXA0CTL/ECCFLXA0T0CTL/ECCFLXA0T1CTL Register Contents (2/2)

Bit Position	Bit Name	Function
4	EC2EDIC	2-Bit Error Detection Interrupt Control This bit controls whether to generate an interrupt when 2-bit error is detected. 0: When 2-bit error is detected, an INTECCDFLRAM interrupt will not be generated. 1: When 2-bit error is detected, an INTECCDFLRAM interrupt will be generated. (the value after reset)
3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	ECER2F	2-Bit Error Detection Flag This flag indicates whether 2-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). When 2-bit error interrupt is enabled (EC2EDIC = 1) and this flag is set, an ECC 2-bit error interrupt (INTECCDFLRAM) is output. Write 1 to the ECER2C bit (bit 10) to clear the flag. When through mode is enable (ECTHM = 1), this bit is cleared. If 2-bit error is detected again while this bit is set, an interrupt will not be generated. 0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred. This bit is read-only. Writing 0 or 1 does not change internal state.
1	ECER1F	1-Bit Error Detection/Correction Flag This flag indicates whether 1-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). Write 1 to the ECER1C bit (bit 9) to clear the flag. When through mode is enabled (ECTHM = 1), this bit is cleared. 0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred. This bit is read-only. Writing 0 or 1 does not change internal state.
0	ECEMF	ECC Error Message Flag This flag indicates whether an error exists in the current read data bus. This bit is updated whenever the RAM outputs data. Because the initial value of the RAM data is undefined, If the RAM is read before initialization, this bit may be set. When through mode is enabled (ECTHM = 1) and there is no 1-bit error in decode circuit input data, this bit is cleared. 0: The current RAM output data does not have bit errors. 1: The current RAM output data have bit errors.

CAUTION

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.

We recommend initializing the RAM before clearing bits 2 and 1.

21.4.5 ECCFLXA0TMC/ECCFLXA0T0TMC/ECCFLXA0T1TMC — FlexRay ECC Test Mode Control Register

The ECCFLXA0TMC/ECCFLXA0T0TMC/ECCFLXA0T1TMC register is used to switch to the test mode, and this register is for test mode control.

This register can be used when FlexRay is not accessed to RAM.

When writing to bit 7, ETMA1 and ETMA0 need to be 10_B.

Access: This register can be read or written in 16-bit units.

Address: See **Table 21.116, Registers**.

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA1	ETMA0	—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are always read as 0.

Table 21.118 ECCFLXA0TMC/ECCFLXA0T0TMC/ECCFLXA0T1TMC Register Contents (1/2)

Bit Position	Bit Name	Function
15	ETMA1	Access Control Bits 1 and 0 to ECC Test Mode These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 10_B, writing to bits 7 is enabled.
14	ETMA0	
13 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	ECTMCE	ECC Test Mode Enable This bit specifies whether to enable access to test control bits of the test registers and this register. When writing to this bit, (1, 0) should be written to (ETMA1, ETMA0) at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. Test registers: ECCFLXA0TED / ECCFLXA0T0TED / ECCFLXA0T1TED, ECCFLXA0TRC / ECCFLXA0T0TRC / ECCFLXA0T1TRC, ECCFLXA0SYND / ECCFLXA0T0SYND / ECCFLXA0T1SYND, ECCFLXA0HORD / ECCFLXA0T0HORD / ECCFLXA0T1HORD, ECCFLXA0ECRD / ECCFLXA0T0ECRD / ECCFLXA0T1ECRD, ECCFLXA0ERDB / ECCFLXA0T0ERDB / ECCFLXA0T1ERDB Register test control bits: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	ECTRRS	ECC RAM Read Test Mode Selection This bit selects the targets for reading when the ECCFLXA0TED*1 and ECCFLXA0ERDB*2 registers are read. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Read value of the ECCFLXA0TED *1register will be the write value of the ECCFLXA0TED*1 register. Read value of the ECCFLXA0ERDB*2 register will be the write value of the ECCFLXA0ERDB*2 register. 1: Read value of the ECCFLXA0TED*1 register can read RAM data. Read value of the ECCFLXA0ERDB*2 register will be the ECC Data to be written to RAM.

Table 21.118 ECCFLXA0TMC/ECCFLXA0T0TMC/ECCFLXA0T1TMC Register Contents (2/2)

Bit Position	Bit Name	Function
3	ECREOS	ECC Redundant Bit Output Data Selection This bit specifies which is output to the ECC to be stored in RAM, the ECC data generated for write data or the value of the ECCFLXA0ERDB*2 register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: ECC data is generated for write data is stored in RAM. 1: The value of ECCFLXA0ERDB*2 Register is stored in RAM.
2	ECENS	ECC Encoder Input Selection This bit specifies data written to RAM or the value of the ECCFLXA0TED*1 register as the input to the ECC encoder. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: ECC data is generated from write data to RAM 1: ECC data is generated from register value of the ECCFLXA0TED*1.
1	ECDCS	ECC Bit Error Detection/Correction Flag This bit specifies which data is for generation of syndrome code and error detection, RAM data or the value of ECCFLXA0TED*1. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Syndrome code generation and error detection are performed from RAM Data. 1: Syndrome code generation and error detection are performed from ECCFLXA0TED*1 register value.
0	ECREIS	ECC Redundant Bit Input Data Selection This bit specifies which ECC data is for generation of syndrome code and error detection, ECC data stored in RAM or the value of the ECCFLXA0ERDB*2 . Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Syndrome code generation and error detection are performed from ECC data stored in RAM. 1: Syndrome code generation and error detection are performed from ECCFLXA0ERDB*2 register value.

Note 1. The same applies to ECCFLXA0T0TMC and ECCFLXA0T1TMC.

Note 2. The same applies to ECCFLXA0T0ERDB and ECCFLXA0T1ERDB.

21.4.6 ECCFLXA0TED/ECCFLXA0T0TED/ECCFLXA0T1TED — FlexRay ECC Encode/Decode Input/Output Replacement Test Register

In ECC test mode, this register handles test data.

This register value is used to generate ECC data or syndrome code.

This register can be accessed when ECC test mode in enabled (ECCFLXA0TMC*[1].ECTMCE = 1). When ECCFLXA0TMC*[1].ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when FlexRay is not accessed to RAM.

Note 1. The same applies to ECCFLXA0T0TMC and ECCFLXA0T1TMC.

Access: This register can be read or written in 32-bit units.

Address: See **Table 21.116, Registers**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.119 ECCFLXA0TED/ECCFLXA0T0TED/ECCFLXA0T1TED Register Contents

Bit Position	Bit Name	Function
31 to 0	ECEDB[31:0]	When ECCFLXA0TMC*[1].ECENS = 1, these bits generate ECC data from value of this register and store the register value in RAM. When ECCFLXA0TMC*[1].ECDCS = 1, these bits generate syndrome code from the value of the register and store the register value in ECC decode syndrome data register (ECCFLXA0SYND*[2]). In addition, when ECCFLXA0TMC*[1].ECTRRS = 1, RAM data [31:0] instead of written data is read for the value of this register.

Note 1. The same applies to ECCFLXA0T0TMC and ECCFLXA0T1TMC.

Note 2. The same applies to ECCFLXA0T0SYND and ECCFLXA0T1SYND.

21.4.7 ECCFLXA0TRC/ECCFLXA0T0TRC/ECCFLXA0T1TRC — FlexRay ECC Redundant Bit Data Control Test Register

This register is a test register for ECC data in ECC test mode and contains four 8-bit registers in each ECC unit, ECCFLXA0YND / ECCFLXA0T0YND / ECCFLXA0T1YND, ECCFLXA0HORD / ECCFLXA0T0HORD / ECCFLXA0T1HORD, ECCFLXA0ECRD / ECCFLXA0T0HORD / ECCFLXA0T1HORD, ECCFLXA0ERDB / ECCFLXA0T0ERDB / ECCFLXA0T1ERDB.

This register can be accessed when ECC test mode is enabled (ECCFLXA0TMC*1.ECTMCE = 1). When ECCFLXA0TMC*1.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when FlexRay is not accessed to RAM.

Note 1. The same applies to ECCFLXA0T0TMC and ECCFLXA0T1TMC.

Access: This register can be read or written in 32-bit units.

Address: See **Table 21.116, Registers**.

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECCFLXA0SYND/ECCFLXA0T0SYND/ECCFLXA0T1SYND (see **Section 21.4.8**)								ECCFLXA0HORD/ECCFLXA0T0HORD/ECCFLXA0T1HORD (see **Section 21.4.9**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECCFLXA0ECRD/ECCFLXA0T0ECRD/ECCFLXA0T1ECRD (see **Section 21.4.10**)								ECCFLXA0ERDB/ECCFLXA0T0ERDB/ECCFLXA0T1ERDB (see **Section 21.4.11**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

21.4.8 ECCFLXA0SYND/ECCFLXA0T0SYND/ECCFLXA0T1SYND — FlexRay ECC Decode Syndrome Data Register

This register is a read-only register for storing generated syndrome data in ECC test mode.

Writing to this register is ignored.

This register is read-only when ECC test mode is enabled (ECCFLXA0TMC*1.ECTMCE = 1). When ECC test mode is disabled (ECCFLXA0TMC*1.ECTMCE = 0), 00_H is read.

Note 1. The same applies to ECCFLXA0T0TMC and ECCFLXA0T1TMC.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 21.116, Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SYND6	SYND5	SYND4	SYND3	SYND2	SYND1	SYND0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 21.120 ECCFLXA0SYND/ECCFLXA0T0SYND/ECCFLXA0T1SYND Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	SYND[6:0]	These bits store generated syndrome code as needed.

21.4.9 ECCFLXA0HORD/ECCFLXA0T0HORD/ECCFLXA0T1HORD — FlexRay ECC 7-Bit Redundant Bit Data Hold Test Register

This register is for storing ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can be accessed only when ECC test mode is enabled (ECCFLXA0TMC*1.ECTMCE = 1). When ECC test mode is disabled (ECCFLXA0TMC*1.ECTMCE = 0), 00_H is read.

Note 1. The same applies to ECCFLXA0T0TMC and ECCFLXA0T1TMC.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 21.116, Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	HORD6	HORD5	HORD4	HORD3	HORD2	HORD1	HORD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 21.121 ECCFLXA0HORD/ECCFLXA0T0HORD/ECCFLXA0T1HORD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	HORD[6:0]	These bits store ECC code for read RAM data as needed. When ECCFLXA0TMC*1.ECTRRS =1 and if ECCFLXA0TED*2 register is read, ECC code is stored.

Note 1. The same applies to ECCFLXA0T0TMC and ECCFLXA0T1TMC.

Note 2. The same applies to ECCFLXA0T0TED and ECCFLXA0T1TED.

21.4.10 ECCFLXA0ECRD/ECCFLXA0T0ECRD/ECCFLXA0T1ECRD — FlexRay ECC Encode Test Register

This register is a read-only register for storing generated ECC data for read RAM data in ECC test mode.

Writing to this register is ignored.

This register can be accessed only when ECC test mode is enabled (ECCFLXA0TMC*1.ECTMCE = 1). When ECC test mode is disabled (ECCFLXA0TMC*1.ECTMCE = 0), 00_H is read.

Note 1. The same applies to ECCFLXA0T0TMC and ECCFLXA0T1TMC.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 21.116, Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ECRD6	ECRD5	ECRD4	ECRD3	ECRD2	ECRD1	ECRD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 21.122 ECCFLXA0ECRD/ECCFLXA0T0ECRD/ECCFLXA0T1ECRD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ECRD[6:0]	These bits can read ECC data generated at the time of RAM data writing and can read ECC data for data written in the ECCFLXA0TED*1 register when ECCFLXA0TMC*2.ECENS =1.

Note 1. The same applies to ECCFLXA0T0TED and ECCFLXA0T1TED.

Note 2. The same applies to ECCFLXA0T0TMC and ECCFLXA0T1TMC.

21.4.11 ECCFLXA0ERDB/ECCFLXA0T0ERDB/ECCFLXA0T1ERDB — FlexRay ECC Redundant Bit Input/Output Replacement Register

In ECC test mode, this register handles test data.

This register value can be handled as generated ECC data at the time of writing to RAM or as read ECC data at the time of reading RAM data.

This register can be accessed when ECC test mode in enabled (ECCFLXA0TMC*1.ECTMCE = 1). When ECCFLXA0TMC*1.ECTMCE = 0, writing to this register is ignored and 00_H is read.

Note 1. The same applies to ECCFLXA0T0TMC and ECCFLXA0T1TMC.

Access: This register can be read or written in 8-bit units.

Address: See **Table 21.116, Registers**.

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ERDB6	ERDB5	ERDB4	ERDB3	ERDB2	ERDB1	ERDB0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.123 ECCFLXA0ERDB/ECCFLXA0T0ERDB/ECCFLXA0T1ERDB Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	ERDB[6:0]	These bits can store this register value as ECC data when ECCFLXA0TMC*1.ECREOS = 1. When the register is read while ECCFLXA0TMC*1.ECREIS =1, the value read from these bits is ECC data read from the RAM. When ECCFLXA0TMC*1.ECTRRS = 1, ECC data to be stored in RAM will be read for this register value instated of written data.

Note 1. The same applies to ECCFLXA0T0TMC and ECCFLXA0T1TMC.

21.4.12 SELB_READTEST — ECCREAD Test Select Register

SELB_READTEST is used to check read/write access to the FLXAn ECC registers.

For detail, see **Section 16.7.11, SELB_READTEST — ECCREAD Test Select Register**

Section 22 Window Watchdog Timer (WDTA)

This section contains a generic description of the Window Watchdog Timer (WDTA).

The first part of this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of WDTA.

22.1 Features of RH850/F1M WDTA

22.1.1 Number of Units and Channels

This microcontroller has the following number of WDTA units.

Table 22.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	2		
Name	WDTAn (n = 0, 1)		

Table 22.2 Index

Index	Description
n	Throughout this section, the individual window watchdog timer units are identified by the index "n"; for example, WDTAnWDTE(n = 0,1) is the WDTAn enable register.

22.1.2 Register Base Address

WDTAn base addresses are listed in the following table.

WDTAn register addresses are given as offsets from the base addresses.

Table 22.3 Register Base Addresses

Base Address Name	Base Address
<WDTA0_base>	FFED 0000_H
<WDTA1_base>	FFED 1000_H

22.1.3 Clock Supply

The WDTAn clock supply is shown in the following table.

Table 22.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
WDTA0	WDTATCKI	CKSCLK_AWDTA
	Register access clock	CPUCLK2
WDTA1	WDTATCKI	LS IntOSC
	Register access clock	CPUCLK2

22.1.4 Interrupt Requests

WDTAn interrupt requests are listed in the following table.

Table 22.5 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
WDTA0			
INTWDTAn	WDTA0 75% interrupt	40	—
WDTA1			
INTWDTAn	WDTA1 75% interrupt	41	—

Table 22.6 Interrupt Requests (FE Level Non-Maskable Interrupts)

Unit Interrupt Signal	Description	Interrupt Name	DMA Trigger Number
WDTA0			
WDTAnTNMI	WDTA0 FENMI interrupt (in the WDTA error detection mode with an NMI request)	WDTA0NMI	—
WDTA1			
WDTAnTNMI	WDTA1 FENMI interrupt (in the WDTA error detection mode with an NMI request)	WDTA1NMI	—

22.1.5 Reset Sources

WDTAn reset sources are listed in the following table. WDTAn is initialized by these reset sources.

Table 22.7 Reset Sources

Unit Name	Reset Source
WDTA0	Reset sources (AWORES)
WDTA1	All reset sources (ISORES)

Note: WDTA1 is stopped in STOP mode.

22.2 Overview

22.2.1 Functional Overview

WDTA has the following functions:

- Selection of the operation mode after reset, by using the option bytes

Enabling/disabling of WDTA, starting/stopping of the counter after reset, setting of the counter overflow time, and enabling/disabling of the VAC function can be selected. WDTA startup options to be set by the option bytes are described in **Table 22.8**.

- WDTA trigger function

Writing an activation code to the WDTA trigger register starts WDTA and restarts the counter. Activation codes include fixed activation codes and variable activation codes (VAC function). In a variable activation code, a different value from the previous time (variable value) is written to the WDTA trigger register, which causes the counter to be restarted.

- 75% interrupt request signal

An interrupt request signal can be generated when the WDTA counter reaches 75% of the overflow interval time (this function can be enabled or disabled by the setting of WDTAnMD.WDTAnWIE).

- Window function

The period during which writing to the WDTA trigger register is valid (window-open period) can be set. Writing to the WDTA trigger register during a period other than the window-open period causes an error.

- WDTA error detection function

When an error is detected, a non-maskable interrupt request or an internal reset is generated.

For details about the error sources, see **Section 22.5.3, WDTA Error Detection**.

Table 22.8 WDTA Start-Up Options

Start-Up Option	Function	Description	Option Byte
OPWDEN	WDTA setting	Enables/disables the WDTA. 0: WDTA is disabled 1: WDTA is enabled	• WDTA0: OPBT0.OPBT0[19] • WDTA1: OPBT0.OPBT0[23]
OPWDOVF[2:0]	Overflow interval time reset value setting	Specifies the reset value of the overflow interval time control bits WDTAnMD.WDTAnOVF[2:0].	• WDTA0/WDTA1: OPBT0.OPBT0[18:16]
OPWDRUN	Start mode setting	Specifies the start mode. 0: Software trigger start mode 1: Default start mode For details, see **Section 22.5.1, WDTA after Reset Release**.	• WDTA0: OPBT0.OPBT0[20] • WDTA1: OPBT0.OPBT0[24]
OPWDVAC	Variable activation code selection	Specifies the trigger register for the generation of counter re-start triggers to keep the counter from overflowing. 0: WDTAnWDTE (fixed) 1: WDTAnEVAC (variable) When WDTAnWDTE is selected, the value to be written to the register (activation code) is fixed (ACH). When WDTAnEVAC is selected, the activation code to be written to the register is variable. For details, see **Section 22.5.2, WDTA Trigger** and **22.5.2.1, Calculating an Activation Code when the VAC Function is Used**.	• WDTA0: OPBT0.OPBT0[22] • WDTA1: OPBT0.OPBT0[26]

NOTE

For the option byte settings, see **Section 36.9, Option Bytes**.

22.2.2 Block Diagram

Figure 22.1 shows the main components of the WDTA.

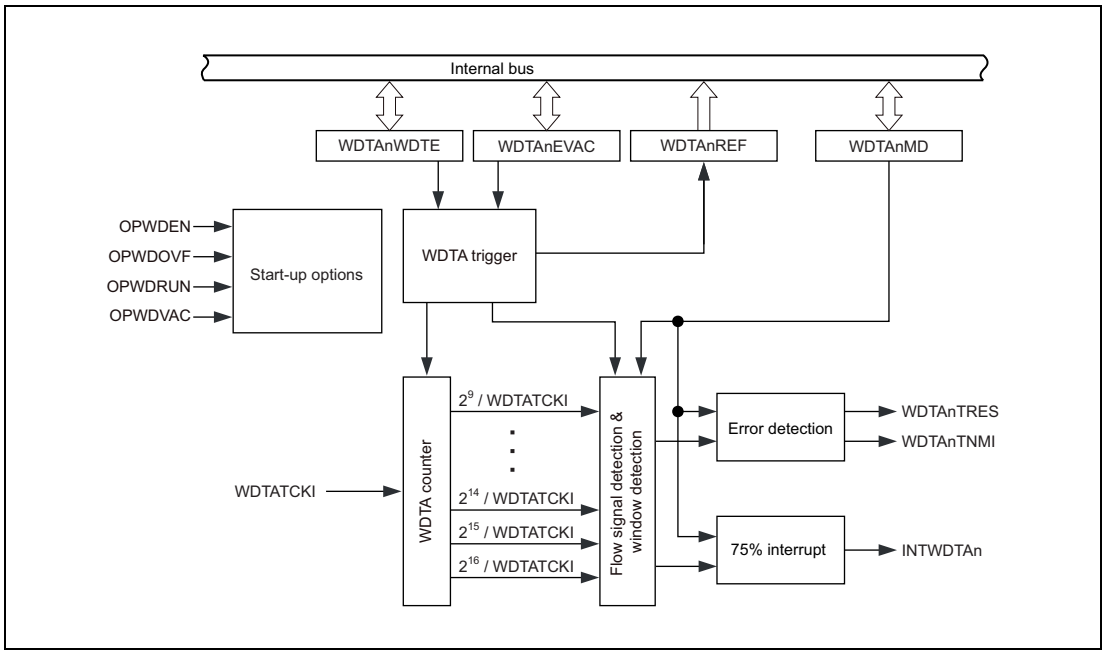


Figure 22.1 Block Diagram of Window Watchdog Timer A

22.3 Registers

22.3.1 List of Registers

WDTA registers are listed in the following table.

For details about <WDTAn_base>, see **Section 22.1.2, Register Base Address**.

Table 22.9 List of Registers

Module	Register	Symbol	Address
WDTAn	WDTA enable register	WDTAnWDTE	<WDTAn_base> + 0000_H
WDTAn	WDTA enable VAC register	WDTAnEVAC	<WDTAn_base> + 0004_H
WDTAn	WDTA reference value register	WDTAnREF	<WDTAn_base> + 0008_H
WDTAn	WDTA mode register	WDTAnMD	<WDTAn_base> + $000C_H$

22.3.2 WDTAnWDTE — WDTA Enable Register

This register is the WDTA trigger register when the VAC function is not used (start-up option OPWDVAC = 0).

Writing AC_H to this register generates a WDTA trigger and starts or restarts the WDTA counter. See **Section 22.5.2, WDTA Trigger**, for details.

The behavior of this register depends on the setting of the start-up option (OPWDVAC), see **Table 22.12, WDTAnWDTE Behavior**.

WDTA0 is initialized by AWORES.

WDTA1 is initialized by a reset of any type.

Access: This register can be read or written in 8-bit units.

Address: <WDTAn_base> + 0000_H

Value after reset: The initial value depends on the start-up options (OPWDEN, OPWDRUN, and OPWDVAC). See **Table 22.11, Values of WDTAnRUN7 after Reset**.

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

Table 22.10 WDTAnWDTE Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTAnRUN [7:0]	Writing the fixed activation code (AC_H) generates the WDTA trigger and starts/ restarts the WDTAn counter. Writing a value other than AC_H generates an error. The WDTAn cannot be stopped once it is started. See **Table 22.12, WDTAnWDTE Behavior**, when reading from or writing to these bits.

The WDTAnRUN7 bit is only valid if WDTA is enabled (OPWDEN = 1) and the VAC function is disabled (OPWDVAC = 0). **Table 22.11** lists the values of the WDTAnRUN7 bit after reset according to the start-up options.

Table 22.11 Values of WDTAnRUN7 after Reset

Start-Up Options			Start Mode	Value of WDTAnRUN7 after Reset
OPWDEN	OPWDVAC	OPWDRUN		
1	0	1	Default start	1
		0	Software trigger start	0

The behavior of WDTAnWDTE during read/write accesses depends on the OPWDVAC setting, as shown in **Table 22.12, WDTAnWDTE Behavior**.

Table 22.12 WDTAnWDTE Behavior

OPWDVAC	Description	WDTAnWDTE	
		Read	Write
0	The VAC function is disabled. WDTAnWDTE is enabled.	$2C_H$ is returned (in software trigger start mode, before the activation of WDTAn). AC_H is returned (after the activation of WDTAn).	WDTA trigger Write AC_H *1.
1	The VAC function is enabled. WDTAnWDTE is disabled.	$2C_H$ is returned.	Writing is ignored.

Note 1. Any other write value will cause an error.

22.3.3 WDTAnEVAC — WDTA Enable VAC Register

This register is the WDTA trigger register when the VAC function is used (start-up option OPWDVAC = 1).

Writing a correct activation code to this register generates a WDTA trigger and starts or restarts the WDTA counter. For details, see **Section 22.5.2, WDTA Trigger**. For details about the activation codes when the VAC function is used, see **Section 22.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

The behavior of this register depends on the setting of the start-up option (OPWDVAC). See **Table 22.15, WDTAnEVAC Behavior**.

WDTA0 is initialized by AWORES.

WDTA1 is initialized by a reset of any type.

Access: This register can be read or written in 8-bit units.

Address: <WDTAn_base> + 0004_H

Value after reset: The initial value depends on the start-up options (OPWDEN, OPWDRUN and OPWDVAC). See **Table 22.14, Values of WDTAnEVAC7 after Reset**.

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

Table 22.13 WDTAnEVAC Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTAnEVAC [7:0]	Writing a variable activation code generates the WDTA trigger and starts/ restarts the WDTAn counter. Writing an incorrect activation code generates an error. The WDTAn cannot be stopped once it is started. See **Table 22.15, WDTAnEVAC Behavior**, when reading from or writing to these bits.

The WDTAnEVAC7 bit is only valid if WDTA is enabled (OPWDEN = 1) and the VAC function is enabled (OPWDVAC = 1). **Table 22.14** lists the values of the WDTAnEVAC7 bit after reset according to the start-up options.

Table 22.14 Values of WDTAnEVAC7 after Reset

Start-Up Options			Start Mode	Value of WDTAnEVAC7 after Reset
OPWDEN	OPWDVAC	OPWDRUN		
1	1	1	Default start	1
		0	Software trigger start	0

The behavior of WDTAnEVAC during read/write accesses depends on the OPWDVAC setting, as shown in **Table 22.15**.

Table 22.15 WDTAnEVAC Behavior

OPWDVAC	Description	WDTAnEVAC	
		Read	Write
0	The VAC function is disabled. WDTAnEVAC is disabled.	$2C_H$ is returned.	Writing is ignored.
1	The VAC function is enabled. WDTAnEVAC is enabled.	$2C_H$ is returned (in software trigger start mode, before the activation of WDTAn). The variable activation code written last is returned (after the activation of WDTAn).	Write the variable activation code*1 For details, see **Section 22.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

Note 1. Any other write value will cause an error.

22.3.4 WDTAnREF — WDTA Reference Value Register

This register contains the reference value for calculating the activation code of the VAC function. It is automatically updated after every trigger operation. See **Section 22.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

If the VAC function is disabled (OPWDVAC = 0), reading this register returns 00_H.

WDTA0 is initialized by AWORES.

WDTA1 is initialized by a reset of any type.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <WDTAn_base> + 0008_H

Value after reset: 00_H

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

Table 22.16 WDTAnREF Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTAnREF[7:0]	Reference value for activation code calculation for the VAC function

22.3.5 WDTAnMD — WDTA Mode Register

This register specifies the overflow interval time, 75% interrupt enable/disable, the error mode, and the window-open period.

The value of this register can be updated only once after reset release and before the first trigger is generated. The updated value will be effective after the WDTA trigger register is written to next.

Updating this register after the first WDTA trigger is generated generates an error, but an error does not occur if the same value has been written to it.

WDTA0 is initialized by AWORES.

WDTA1 is initialized by a reset of any type.

Access: This register can be read or written in 8-bit units.

Address: <WDTAn_base> + $000C_H$

Value after reset: The initial value depends on the start-up options (OPWDOVF[2:0]). See **Table 22.8, WDTA Start-Up Options**.

Bit	7	6	5	4	3	2	1	0
	—	WDTAnOVF[2:0]			WDTAnWIE	WDTAnERM	WDTAnWS[1:0]	
Value after reset	0	*1	*1	*1	0	1	1	1
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The WDTAnOVF[2:0] value after reset can be set by the start-up options OPWDOVF[2:0].

Table 22.17 WDTAnMD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 4	WDTAnOVF[2:0]	Selects the overflow interval time.

WDTAn OVF2	WDTAn OVF1	WDTAn OVF0	Overflow Interval Time
0	0	0	2^9 / WDTATCKI
0	0	1	2^{10} / WDTATCKI
0	1	0	2^{11} / WDTATCKI
0	1	1	2^{12} / WDTATCKI
1	0	0	2^{13} / WDTATCKI
1	0	1	2^{14} / WDTATCKI
1	1	0	2^{15} / WDTATCKI
1	1	1	2^{16} / WDTATCKI

3	WDTAnWIE	Enables/disables the 75% interrupt request INTWDTAn. 0: INTWDTAn is disabled. 1: INTWDTAn is enabled.
2	WDTAnERM	Specifies the error mode. 0: NMI request mode 1: Reset mode
1, 0	WDTAnWS[1:0]	Selects the window-open period.

WDTAn WS1	WDTAn WS0	Window-Open Period
0	0	25%
0	1	50%
1	0	75%
1	1	100%

22.4 Interrupt Sources

WDTA checks the status of the WDTA counter value, detects illegal accesses to the WDTA-related registers, and generates an interrupt request. The following are WDTA interrupt requests:

(1) INTWDTAn (WDTA timer count 75% interrupt request)
An interrupt request signal is generated at 75% of the counter overflow time of the WDTA timer. An interrupt request signal can be enabled or disabled by using the WDTA mode register (WDTAnMD).

(2) WDTAnTNMI (WDTA error detection interrupt)
Detection of a WDTA error to generation of an NMI interrupt request. The WDTA mode register (WDTAnMD) can be used to switch between an NMI interrupt and a reset. For details about WDTA errors, see **Section 22.5.3, WDTA Error Detection**.

22.5 Functions

22.5.1 WDTA after Reset Release

22.5.1.1 Start Modes

There are two start modes (software start mode and default start mode) when WDTAn starts after reset release. The start mode can be selected by the start-up option.

The start mode selection is listed in **Table 22.18**.

Table 22.18 Start Modes

Start-Up Option OPWDRUN	Start Mode	Description
0	Software trigger	• The WDTA counter stops (0000_H) after reset release. • Writing an activation code to the WDTA trigger register starts WDTA.
1	Default	The WDTA counter starts after reset release.

22.5.1.2 WDTA Settings after Reset Release

(1) **Table 22.19** shows the WDTA settings after reset release.

Table 22.19 WDTA Settings after Reset Release

Function	Setting	Remark
WDTA enable/disable	Specified by start-up options	
Start mode		
VAC function		
WDTA overflow interval time	Specified by start-up options	Modification is possible only once by the setting of the WDTA mode register (WDTAnMD).
75% interrupt mode	75% interrupt disabled	
Behavior on error detection	Reset generation	
Window-open period	100%	

The setting of the WDTA mode register (WDTAnMD) is enabled when the first WDTA trigger is generated (writing an activation code to WDTAnWDTE and WDTAnEVAC). Perform the WDTAnMD register setting before a WDTA trigger is generated.

Setting of WDTA by using WDTAnMD is possible only once. If the value set to WDTAnMD is changed after a WDTA trigger is generated, an error occurs. However, an error does not occur if the same value is set.

22.5.1.3 Default Start Mode Timing

The default start mode timing and the changes to the WDTA settings are illustrated in **Figure 22.2**.

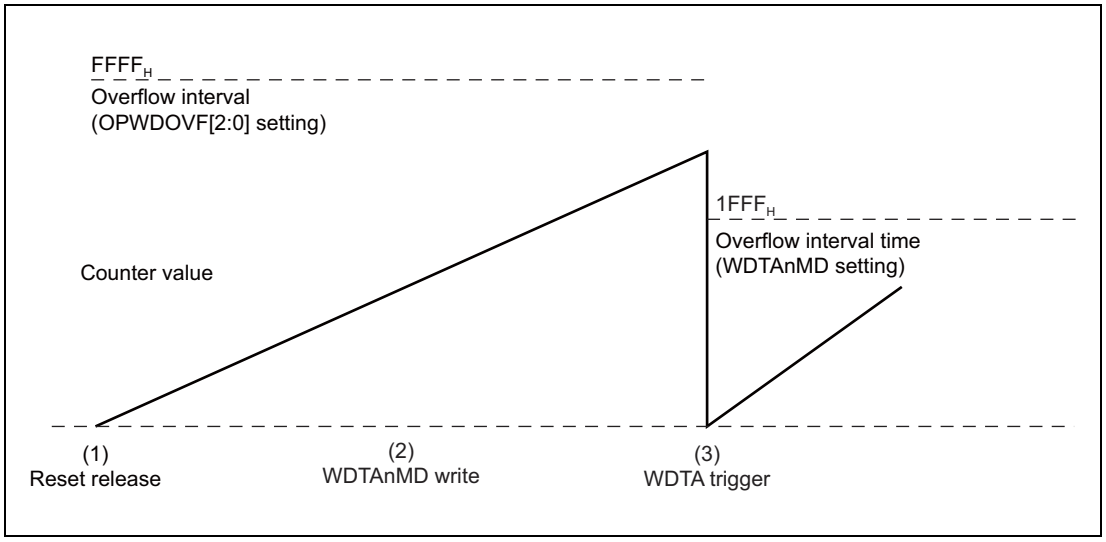


Figure 22.2 Timing Diagram of WDTA Start in Default Start Mode

The timing diagram shown in **Figure 22.2** shows the following behaviors:

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

Example: Overflow interval time after reset release
= 2^{16}/WDTATCKI (OPWDOVF[2:0] = 111_B)

(2) WDTAnMD is set before a WDTA trigger is generated. Note that the setting is not applied immediately.

(3) Write to the WDTA trigger register before the WDTA counter overflows.
The WDTAnMD setting is applied by the generation of a WDTA trigger.

Example: Overflow interval time after a WDTA trigger is generated
= 2^{13}/WDTATCKI

22.5.1.4 Software Trigger Start Mode Timing

The software trigger start mode timing and the changes to the WDTA settings are illustrated in **Figure 22.3**.

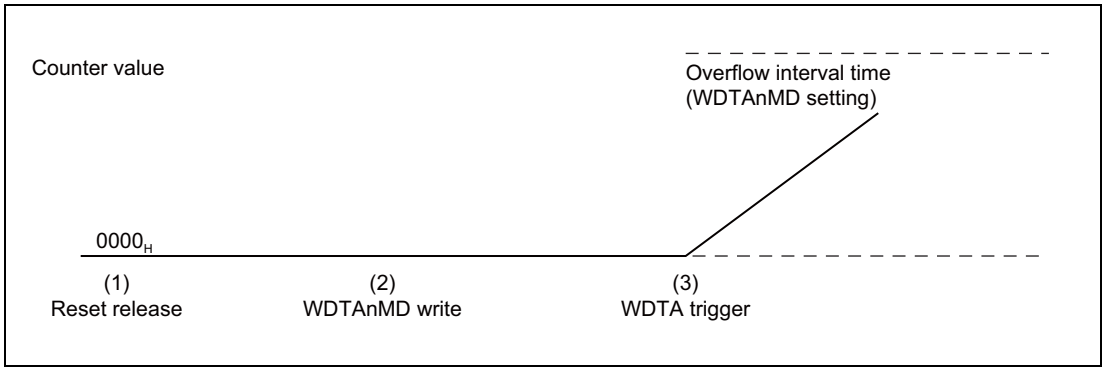


Figure 22.3 Timing Diagram of WDTA Start in Software Trigger Start Mode

The timing diagram shown in **Figure 22.3** shows the following behaviors:

(1) The WDTA counter value remains 0000_H until the first trigger is generated after reset release. The overflow interval time is set by start-up options, but this setting has no effect because the counter is not operating.

(2) WDTAnMD is set before a WDTA trigger is generated. Note that the setting is not applied immediately.

(3) The WDTA counter starts by the generation of a WDTA trigger.
The overflow interval time specified in WDTAnMD and other settings are applied.

22.5.2 WDTA Trigger

Writing a special value called an activation code to the WDTA enable register (WDTAnWDTE) and the WDTA enable VAC register (WDTAnEVAC) leads to generation of a WDTA trigger.

The WDTA trigger has the following functions:

- Starting the WDTA counter in software trigger start mode
- Restarting the WDTA counter
- Setting the WDTA mode specified by the WDTAnMD register (only for the first WDTA trigger after reset release)

The WDTA trigger register, which generates a WDTA trigger, is specified by the start-up option OPWDVAC.

Table 22.20 lists the WDTA trigger registers and activation codes.

Table 22.20 WDTA Trigger Registers and Activation Codes

Type of Activation Code	Trigger Register	Activation Code
Fixed (OPWDVAC = 0)	WDTAnWDTE	AC_H
Variable (OPWDVAC = 1)	WDTAnEVAC	For details, see **22.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

CAUTION

- **When writing the processing to clear WDTA successively, the clear processing below is not acknowledged for the following period:**
  "12 × CPU clock*1 cycles + 6 × WDT clock*2 cycles"
- **After writing the processing to clear WDTA and then changing to standby mode, the clear processing below is not acknowledged for the following period after return from stand-by mode:**
  "6 × CPU clock*1 cycles + 3 × WDT clock*2 cycles"

Note 1. CPU clock: Clock selected by CKSC_CPUCLKS_CTL and CKSC_CPUCLKD_CTL
Note 2. WDT clock: Clock selected by CKSC_AWDTAD_CTL in WDTA0.
LS IntOSC in WDTA1.

22.5.2.1 Calculating an Activation Code when the VAC Function is Used

Use the following expression to calculate the variable activation code (ExpectWDTE) to be set in the WDTA trigger register (WDTAnEVAC) when the VAC function is used, by using the WDTA reference value register (WDTAnREF):

$$\text{ExpectWDTE} = AC_H - \text{WDTAnREF (previous)}$$

Note that the value in the WDTAnREF register is updated every time a start-code is written to the trigger register WDTAnEVAC. Use the following expression to calculate the updated value of the WDTAnREF register:

WDTAnREF (following) = (rotate the value of ExpectWDTE to the left by 1 bit)

Table 22.21 lists the variable activation codes according to the number of WDTA triggers.

Table 22.21 Expected Variable Activation Code Development

No*1	WDTAnREF (Previous)		ExpectWDTE (AC_H - WDTAnREF)		WDTAnREF (Following)	
0	0000 0000	00_H	1010 1100	AC_H	0101 1001	59_H
1	0101 1001	59_H	0101 0011	53_H	1010 0110	$A6_H$
2	1010 0110	$A6_H$	0000 0110	06_H	0000 1100	$0C_H$
…	…	…	…	…	…	…

Note 1. Number of triggers after reset

NOTE

Writing an incorrect activation code generates an error.

22.5.3 WDTA Error Detection

WDTA detects an error, including generation of the WDTA count overflow or illegal operations.

The following events are detected as errors:

- WDTA counter overflow
- Incorrect activation code is written to the WDTA trigger register
- Writing to the trigger register at a time outside the window-open period
- When the setting value in the WDTA mode register (WDTAnMD) is changed after the first WDTA trigger is generated
- When the value of the WDTA mode register (WDTAnMD) is changed twice before the first WDTA trigger is generated

22.5.3.1 WDTA Error Mode

When a WDTA error is detected, either an NMI interrupt or a reset is generated according to the setting of the WDTA error mode bit (WDTAnMD.WDTAnERM). The error mode bit after reset release is set to the reset mode.

- WDTAnMD.WDTAnERM = 0: NMI mode
- WDTAnMD.WDTAnERM = 1: Reset mode

Figure 22.4 shows the reset or NMI request generation when the counter overflows and default start mode is selected.

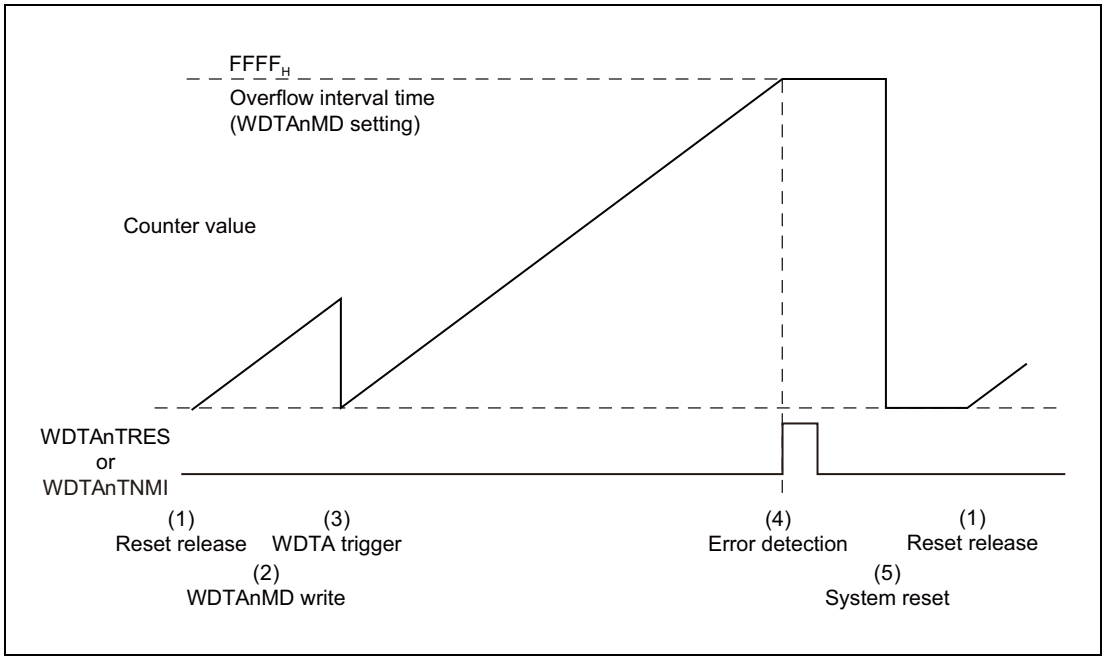


Figure 22.4 Timing Diagram of WDTA NMI Request or Reset Generation

The timing diagram shown in **Figure 22.4** shows the following behaviors:

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

(2) WDTAnMD is set before a WDTA trigger is generated.
In this case, 2^{16}/WDTATCKI is set for the overflow interval time.

(3) A WDTA trigger is generated, and the WDTAnMD setting is applied.

(4) When the counter overflows, an error is detected. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated.
The counter value remains unchanged until a system reset is performed.

(5) When the system is reset, the counter is cleared and stops until reset release.

22.5.4 75% Interrupt Request Signals

When the WDTA counter reaches 75% of the time set for the overflow interval, the interrupt request INTWDTAn is generated.

This function can be enabled or disabled by the WDTAnMD.WDTAnWIE register.

Figure 22.5 shows the 75% interrupt request generated under following conditions:

- Default start mode is selected.
- 75% interrupt request is enabled after the first WDTA trigger is generated
- WDTA overflow interval time is 2^{16}/WDTATCKI

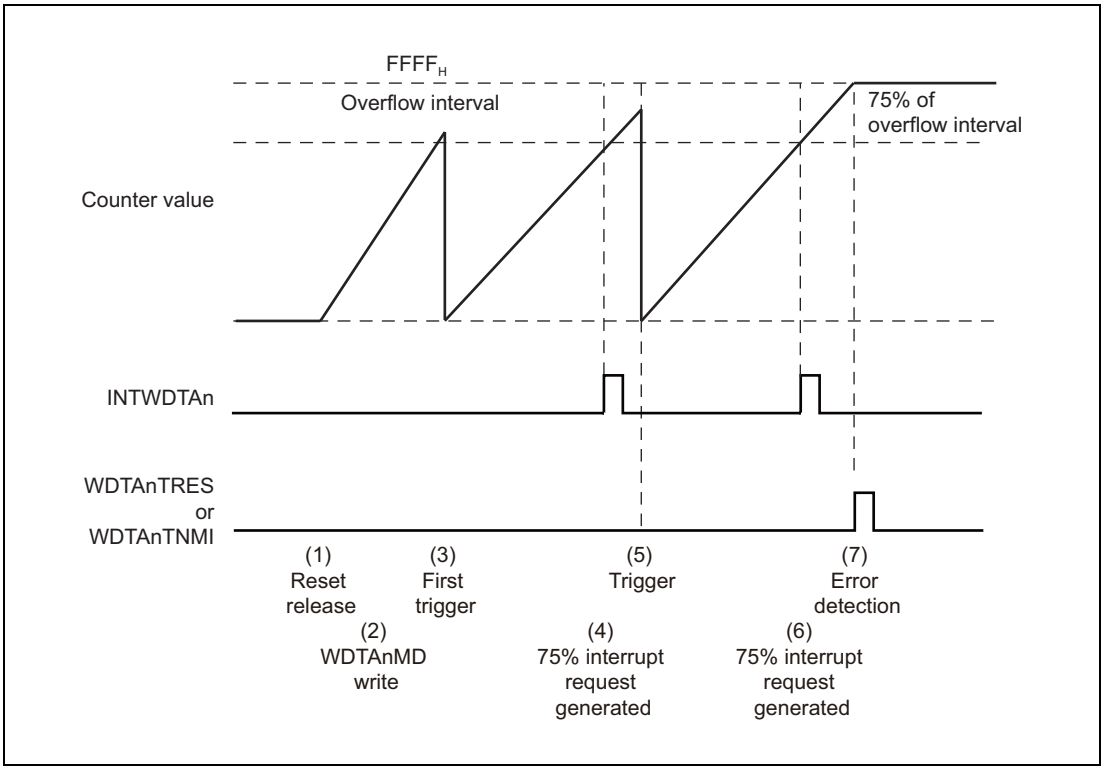


Figure 22.5 Timing Diagram of WDTA 75% Interrupt Request Signals

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

(2) WDTAnMD is set before a WDTA trigger is generated. In this case, 2^{16}/WDTATCKI is set for the overflow interval time.

(3) A WDTA trigger is generated, and the WDTAnMD setting is applied.

(4) When the WDTA counter reaches 75% of the overflow interval timer, interrupt request INTWDTAn is generated.

(5) The WDTA trigger restarts counting.

(6) When the WDTA counter reaches 75% of the overflow interval timer, interrupt request INTWDTAn is generated.

(7) When the counter overflows, an error is detected. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated. The counter value remains unchanged until a system reset is performed.

22.5.5 Window Function

The period when a WDTA trigger is valid (window-open period) can be set. If the window-open period is set to the value less than 100%, the generation of a WDTA trigger not in the window-open period causes an error. The window-open period after reset release is 100%. The period is set to the value by the WDTAnMD.WDTAnWS[1:0] setting after the first WDTA trigger is generated.

Figure 22.6 shows the behavior of the window function under the following conditions.

- Default start mode is selected.
- 25% window-open period is enabled after the first WDTA trigger is generated (WDTAnWS[1:0] = 00_B)
- WDTA overflow interval time is 2^{16}/WDTATCKI

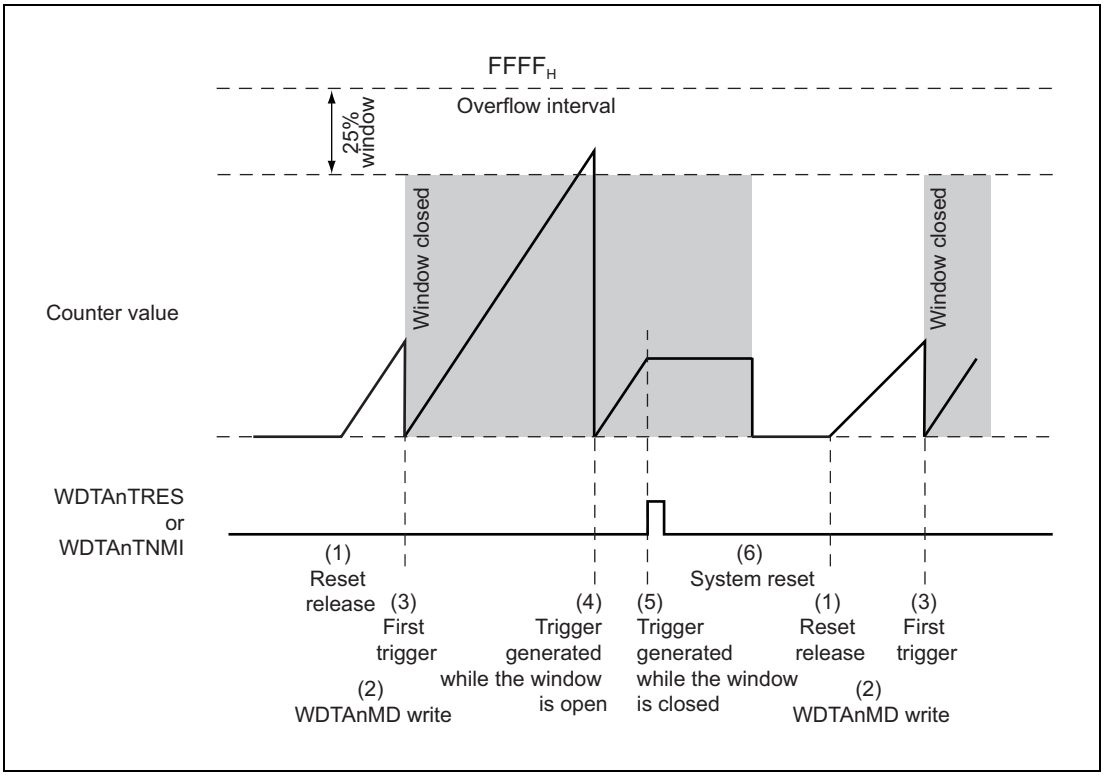


Figure 22.6 Timing Diagram of WDTA Window Function

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

(2) WDTAnMD is set before a WDTA trigger is generated. In this case, 2^{16}/WDTATCKI is set for the overflow interval time.

(3) A WDTA trigger is generated, and the WDTAnMD setting is applied.

(4) During the window-open period, the WDTA trigger restarts counting.

(5) During the window-closed period, an error is detected by the WDTA trigger. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated. The counter value remains unchanged until a system reset is performed.

(6) When the system is reset, the counter is cleared and stops until reset release.

Section 23 OS Timer (OSTM)

This section contains a generic description of the OS Timer (OSTM).

The first part of this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the OSTM.

23.1 Features of RH850/F1M OSTM

23.1.1 Number of Units

This microcontroller has the following number of units of the OSTM.

Table 23.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	5		
Name	OSTMn (n = 0 to 4)		

Table 23.2 Index

Index	Description
n	Throughout this section, the individual OSTM units are identified by the index “n” (n = 0 to 4); for example, OSTMnCNT is the OSTM counter register.

23.1.2 Register Base Address

OSTMn base addresses are listed in the following table.

OSTMn register addresses are given as offsets from the base addresses.

Table 23.3 Register Base Addresses

Base Address Name	Base Address
<OSTM0_base>	FFD7 0000_H
<OSTM1_base>	FFD7 0100_H
<OSTM2_base>	FFD7 0200_H
<OSTM3_base>	FFD7 0300_H
<OSTM4_base>	FFD7 0400_H

23.1.3 Clock Supply

The OSTM clock supply is shown in the following table.

Table 23.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
OSTMn	PCLK	CPUCLK2
	Register access clock	CPUCLK2

23.1.4 Interrupt Requests

OSTM interrupt requests are listed in the following table.

Table 23.5 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
OSTM0			
OSTMTINT	OSTM0 interrupt	84	—

Table 23.6 Interrupt Request (FE Level Maskable Interrupt Request)

Unit Interrupt Signal	Description	Interrupt Name	DMA Trigger Number
OSTMn			
OSTMTINT	OSTMn interrupt	INTOSTMn_FE	—

23.1.5 Reset Sources

OSTM reset sources are listed in the following table. OSTM is initialized by these reset sources.

Table 23.7 Reset Sources

Unit Name	Reset Source
OSTMn	All reset sources (ISORES)

23.2 Overview

OSTM is a 32-bit timer/counter.

It can be used in interval timer mode or in free-run compare mode. By selecting one of the operating modes, the count direction (up/down) can be specified and the generation of interrupt requests can be controlled.

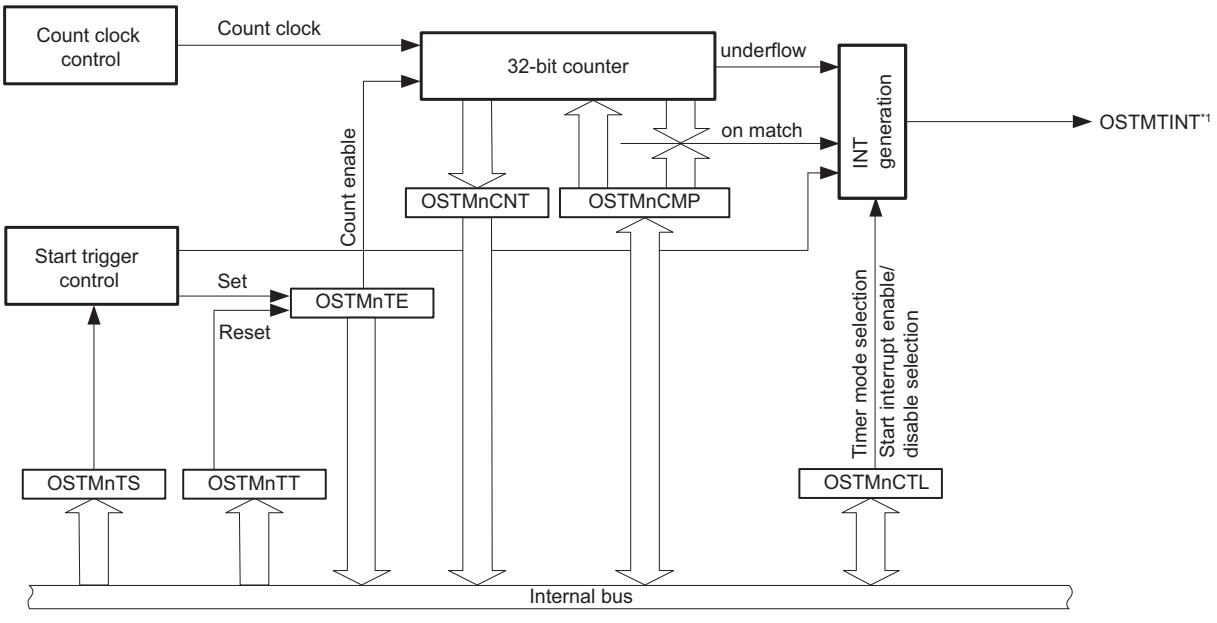
23.2.1 Functional Overview

OSTM has the following features.

- Two operating modes
  - Interval timer mode
  - Free-run compare mode
- OSTMTINT interrupt

23.2.2 Block Diagram

The following block diagram shows the main components of OSTM.



Note 1. OSTMTINT0 is assigned to FEINT and EIINT interrupt requests.
For details, see **Section 7.2.1.2, FE Level Maskable Interrupts** or **Section 7.2.1.3, EI Level Maskable Interrupts**.

Figure 23.1 Block Diagram of OSTM

23.2.3 Count Clock

The count clock used by OSTM is PCLK.

23.2.4 Interrupt Requests (OSTMTINT)

An OSTMTINT interrupt request is generated on counter underflow (interval timer mode) or when the counter matches the compare value (free-run compare mode).

An interrupt request can also be generated on starting and restarting of the counter. This is controlled by the OSTMnCTL.OSTMnMD0 bit.

This is illustrated in the following figure.

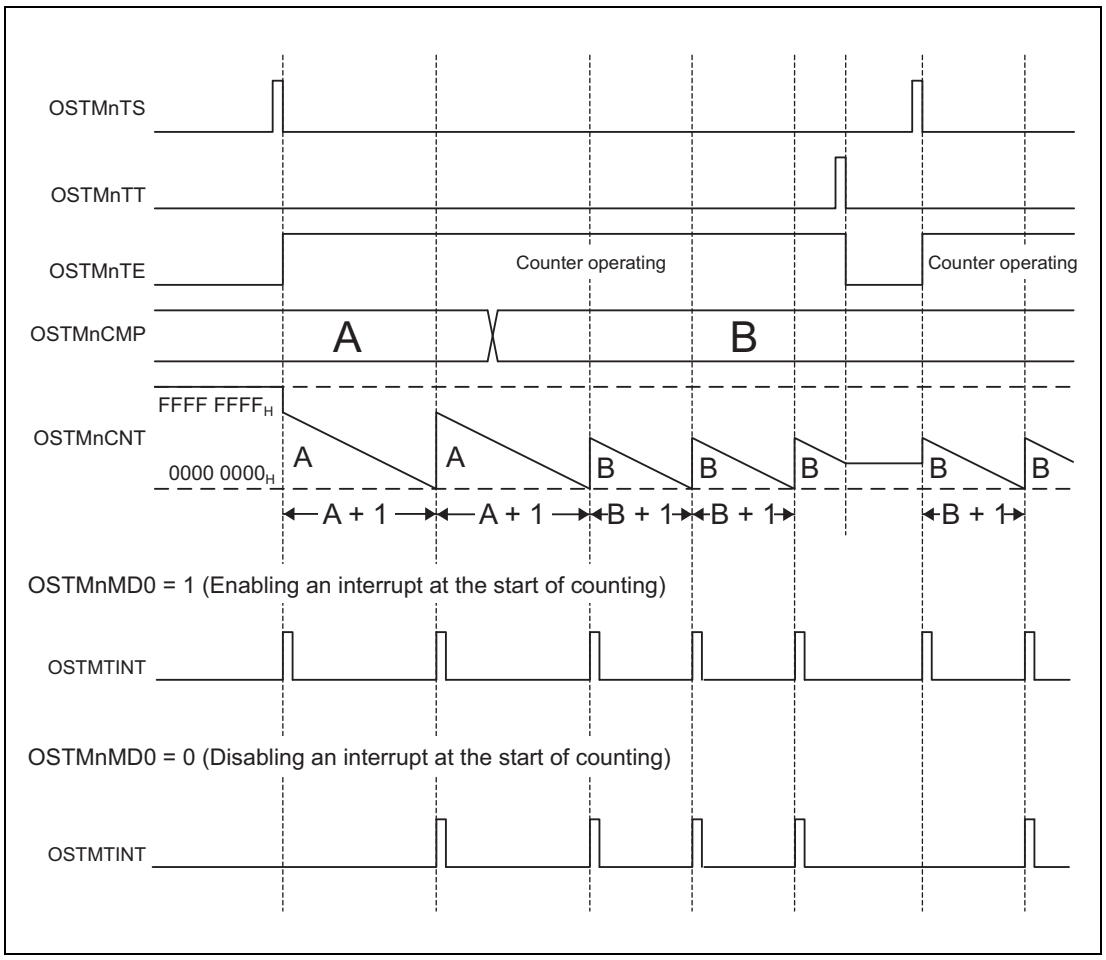


Figure 23.2 Generating an Interrupt when Counting Starts (in Interval Timer Mode)

23.3 Registers

23.3.1 List of Registers

OSTM registers are listed in the following table.

For details about <OSTMn_base>, see **Section 23.1.2, Register Base Address**.

Table 23.8 Registers

Module	Register	Symbol	Address
OSTMn	OSTMn compare register	OSTMnCMP	<OSTMn_base> + 00_H
OSTMn	OSTMn counter register	OSTMnCNT	<OSTMn_base> + 04_H
OSTMn	OSTMn count enable status register	OSTMnTE	<OSTMn_base> + 10_H
OSTMn	OSTMn count start trigger register	OSTMnTS	<OSTMn_base> + 14_H
OSTMn	OSTMn count stop trigger register	OSTMnTT	<OSTMn_base> + 18_H
OSTMn	OSTMn control register	OSTMnCTL	<OSTMn_base> + 20_H
OSTMn	OSTMn emulation register	OSTMnEMU	<OSTMn_base> + 24_H

23.3.2 OSTMnCMP — OSTMn Compare Register

This register stores the start value of the down-counter or the value with which the counter is compared, depending on the operating mode.

Access: This register can be read or written in 32-bit units.

Address: <OSTMn_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OSTMnCMP[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTMnCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.9 OSTMnCMP Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCMP [31:0]	• In interval timer mode: start value of the down-counter • In free-run compare mode: compare value

23.3.3 OSTMnCNT — OSTMn Counter Register

This register indicates the counter value of the timer.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <OSTMn_base> + 04_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OSTMnCNT[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTMnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.10 OSTMnCNT Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCNT [31:0]	Timer counter value

Table 23.11 lists the correspondence between the OSTM operating mode, counting direction, and start value. The start value indicates the value to be read after the operating mode is changed.

Table 23.11 Correspondence between Operating Mode, Counting Direction, and Start Value

Timer Operating Mode	OSTMnCTL.OSTMnMD1	Counting Direction	Start Value
Interval timer mode	0*1	Down	FFFF $FFFF_H$
Free-run compare mode	1	Up	0000 0000_H

Note 1. Value after reset.

23.3.4 OSTMnTE — OSTMn Count Enable Status Register

This register indicates whether the counter is enabled or disabled.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <OSTMn_base> + 10_H

Value after reset: 00_H

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

Table 23.12 OSTMnTE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	OSTMnTE	Indicates whether the counter is enabled or disabled: 0: Counter disabled 1: Counter enabled This bit is set to 1 in response to OSTMnTS.OSTMnTS being set to 1. Setting OSTMnTT.OSTMnTT to 1 resets this bit to 0.

NOTE

If the counter is disabled, the counter value retains its value.

If the counter is restarted:

- restarts counting down from the value in the OSTMnCMP register if it is in interval timer mode or
- restarts counting up from the counter value 0000 0000_H if it is in free-run compare mode.

23.3.5 OSTMnTS — OSTMn Count Start Trigger Register

This register starts the counter.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <OSTMn_base> + 14_H

Value after reset: 00_H

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

Table 23.13 OSTMnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	OSTMnTS	Starts the counter: 0: This setting is invalid. 1: Starts the counter and sets OSTMnTE.OSTMnTE = 1. • In interval timer mode, a forced restart is executed if this bit is set while OSTMnTE.OSTMnTE =1. • In free-run compare mode, setting this bit is ignored as long as OSTMnTE.OSTMnTE = 1.

23.3.6 OSTMnTT — OSTMn Count Stop Trigger Register

This register stops the counter.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <OSTMn_base> + 18_H

Value after reset: 00_H

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

Table 23.14 OSTMnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	OSTMnTT	Stops the counter: 0: This setting is invalid. 1: Stops the counter and clears the OSTMnTE.OSTMnTE bit.

23.3.7 OSTMnCTL — OSTMn Control Register

This register specifies the operating mode for the counter and controls the generation of OSTMTINT interrupt requests when counting starts.

Although this register is readable and writable, writing to it is only possible when OSTMnTE.OSTMnTE = 0; that is, the register becomes read only when OSTMnTE.OSTMnTE = 1.

Access: This register can be read or written in 8-bit units.

Address: <OSTMn_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OSTMnMD1	OSTMnMD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 23.15 OSTMnCTL Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	OSTMnMD1	Specifies the operating mode for the counter: 0: Interval timer mode 1: Free-run compare mode
0	OSTMnMD0	Controls the generation of OSTMTINT interrupt requests at the start of counting: 0: Interrupts when counting starts are disabled. 1: Interrupts when counting starts are enabled.

23.3.8 OSTMnEMU — OSTMn Emulation Register

This register controls operation in combination with SVSTOP.

Access: This register can be read or written in 8-bit units.
Only proceed with writing while the counter is stopped (OSTMnTE.OSTMnTE = 0 and EPC.SVSTOP = 0).

Address: <OSTMn_base> + 24_H

Value after reset: 00_H

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

Table 23.16 OSTMnEMU Register Contents

Bit Position	Bit Name	Function
7	OSTMnSVSDIS	When EPC.SVSTOP = 0 Supply of the count clock is continued when the debugger acquires control of the microcontroller (at breakpoints and so on) regardless of the value of this bit (1/0). When EPC.SVSTOP = 1 0: The count clock is stopped when the debugger acquires control of the microcontroller (at breakpoints and so on). 1: Supply of the count clock is continued when the debugger acquires control of the microcontroller (at breakpoints and so on).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

23.4 Operation

23.4.1 Starting and Stopping OSTM

OSTM is started and stopped as follows:

Starting the timer

OSTM is started by the following setting.

- Setting the OSTMnTS.OSTMnTS bit to 1

The OSTMnTE.OSTMnTE status bit is set to 1.

The counter starts to count up or down in accordance with the settings for operating mode. For details, see **Section 23.4.2, Interval Timer Mode** and **Section 23.4.3, Free-Run Compare Mode**.

Stopping the timer

Setting the OSTMnTT.OSTMnTT bit to 1 stops OSTM.

This also clears the OSTMnTE.OSTMnTE status bit.

23.4.2 Interval Timer Mode

In interval timer mode, OSTM can be used as a reference timer generating interrupt requests at fixed intervals.

23.4.2.1 Basic Operation in Interval Timer Mode

In interval timer mode, the timer counts down from the value specified in the OSTMnCMP register. An OSTMTINT interrupt request is generated when the counter underflows (reaches 0000 0000_H).

To select interval timer mode, set OSTMnCTL.OSTMnMD1 = 0.

The OSTMnCMP register can be rewritten at any time. If it is rewritten during count operation, the counter loads the new OSTMnCMP value when it next reaches 0000 0000_H. Then the counter continues operating with the new value.

OSTMTINT period

The period of OSTMTINT is:

- OSTMTINT generation period = count clock period × (OSTMnCMP + 1)

The following figure shows the basic operation of OSTM when counter-start interrupts are enabled in interval timer mode.

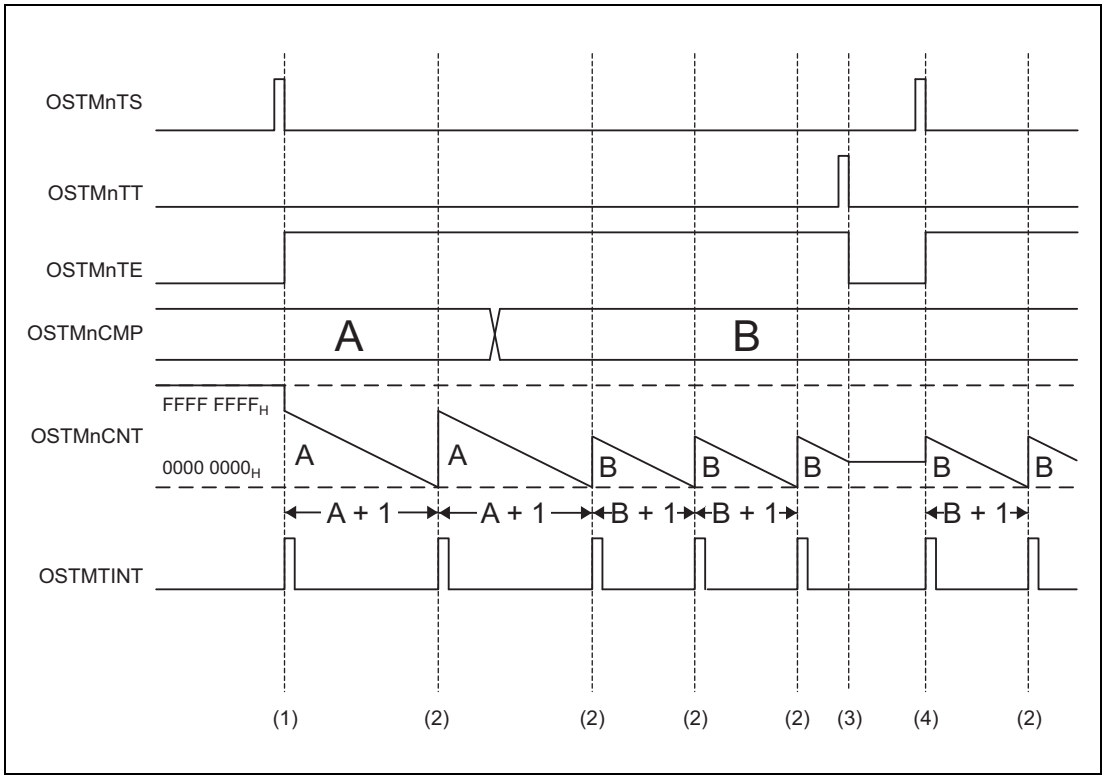


Figure 23.3 Timing Diagram of OSTM in Interval Timer Mode

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter.
The counter starts counting down from the value of OSTMnCMP.
If OSTMnCTL.OSTMnMD0 = 1, OSTMTINT interrupt requests are generated at the start of counting. The OSTMnCNT register indicates the counter value.

(2) When the counter reaches 0000 0000$_H$, an OSTMTINT interrupt request is generated. The counter loads the new start value from OSTMnCMP and continues counting down.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.

(4) When counting is restarted (OSTMnTS.OSTMnTS = 1), the counter loads the new start value from OSTMnCMP and starts counting down.

Forced restart

The counter is forcibly restarted by setting OSTMnTS.OSTMnTS = 1 during counting.

The counter loads the start value from the OSTMnCMP register and continues to count down.

The following figure shows the forced restart of the OS Timer in interval timer mode, with counter-start interrupts enabled (OSTMnCTL.OSTMnMD0 = 1).

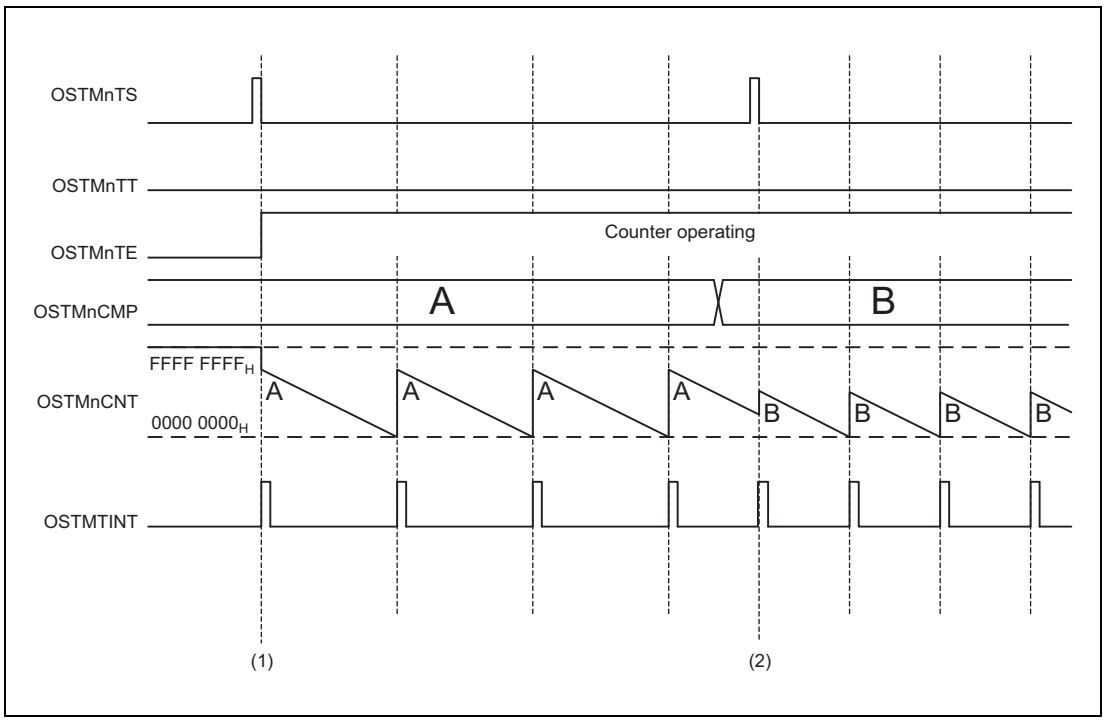


Figure 23.4 Timing Diagram of Forced Restart in Interval Timer Mode

The timing diagram above shows the following operations:

(1) The counter is started and stopped as described under **Figure 23.3, Timing Diagram of OSTM in Interval Timer Mode**.

(2) Setting OSTMnTS.OSTMnTS to 1 restarts the counter while counting is in progress (i.e. while OSTMnTE.OSTMnTE = 1).
The counter immediately restarts counting down, starting with the current value of OSTMnCMP.
When OSTMnCTL.OSTMnMD0 = 1, an OSTMTINT interrupt request is generated when counting starts.

23.4.2.2 Operation when OSTMnCMP = 0000 0000_H

When OSTMnCMP = 0000 0000_H, OSTM behaves as follows.

- When the counter is enabled, the OSTMTINT interrupt request is always set to 1.

The following figure shows operations of OSTM when OSTMnCMP = 0000 0000_H, and counter-start interrupts are enabled.

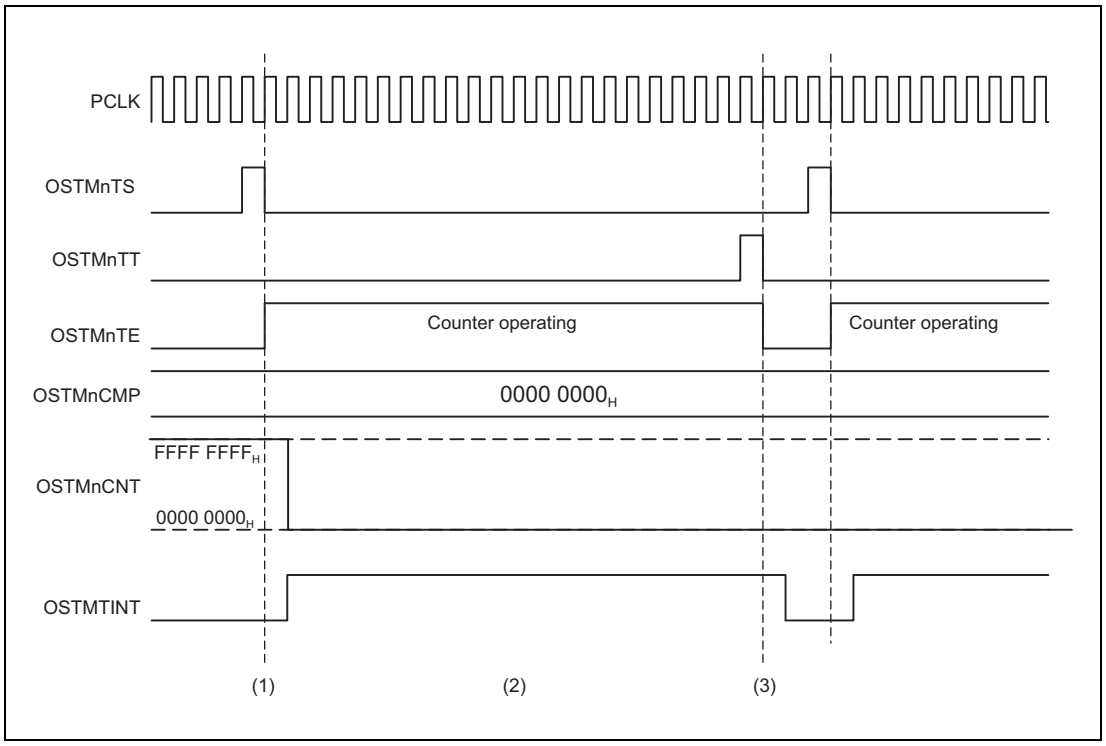


Figure 23.5 Timing Diagram when OSTMnCMP = 0000 0000_H in Interval Timer Mode

The timing diagram above shows the following operations:

(1) The counter is reloaded with the value in OSTMnCMP as soon as it starts counting, so the value 0000 0000_H is retained in OSTMnCMP.

(2) The OSTMTINT interrupt request is continuously asserted.

(3) After the counter stops, the OSTMTINT interrupt request signal is deasserted.

When interrupts on starting of the counter are disabled, no interrupt is generated when counting starts.

23.4.2.3 Setting Procedure for Interval Timer Mode

The setting procedure in interval timer mode after reset release is described below:

Setting procedure

(1) Set the start value of the counter in the OSTMnCMP register.

(2) Select interval timer mode by setting the OSTMnCTL.OSTMnMD1 bit to 0.

(3) Enable or disable interrupts when counting starts (OSTMnCTL.OSTMnMD0).

23.4.3 Free-Run Compare Mode

23.4.3.1 Basic Operation in Free-Run Compare Mode

In free-run compare mode, the counter counts up from 0000 0000_H to FFFF $FFFF_H$. When the value of the OSTMnCMP register matches the current counter value, an OSTMTINT interrupt request is output.

In free-run compare mode, set OSTMnCTL.OSTMnMD1 = 1.

New values can be written to the OSTMnCMP register at any time.

The following figure shows the basic operation of OSTM in free-run compare mode with the start of counting enabled (OSTMnCTL.OSTMnMD0 = 1).

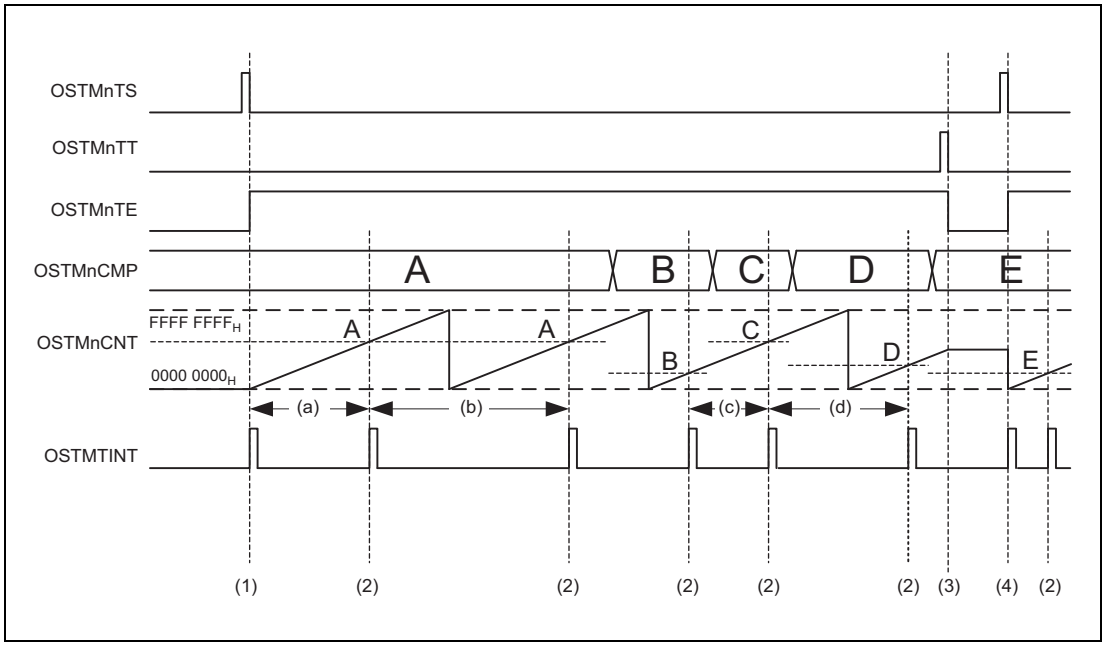


Figure 23.6 Timing Diagram of OSTM in Free-Run Compare Mode

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1.
The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter counts up from 0000 0000_H to FFFF $FFFF_H$. The OSTMnCNT register indicates the counter value.
When OSTMnCTL.OSTMnMD0 = 1, an OSTMTINT interrupt request is generated at the start of counting.

(2) When the current counter value matches the value in the OSTMnCMP register, an OSTMTINT interrupt request is generated.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter.
The counter retains its current value until it is restarted.

(4) Counting by the counter restarts from 0000 0000_H when OSTMnTS.OSTMnTS = 1.

OSTMTINT period

The OSTMTINT generation period is different depending on the starting time. If OSTMnCMP is rewritten during operation, the period is changed according to the size of the new and old compare values.

Table 23.17 OSTMTINT Generation Timing

Old Value for Comparison	New Value for Comparison	Counter Value at Time of Rewriting	Period of OSTMTINT Generation	Label in Timing Diagram
Counter starts			(A + 1) × count clock period	(a)
A	A	No rewriting	(FFFF FFFF$_H$ + 1) × count clock period	(b)
B	C > B	B < counter value < C	(C – B) × count clock period	(c)
C	D < C	Counter value > D,C	(FFFF FFFF$_H$ – C + D + 1) × count clock period	(d)

Forced restart

Forced restart is not performed when the OSTMnTS. OSTMnTS bit is set during counting. The counter ignores the attempted setting and continues counting.

23.4.3.2 Operation when OSTMnCMP = 0000 0000$_H$

The following figure shows the operation of OSTM when OSTMnCMP = 0000 0000$_H$, and counter-start interrupts are enabled (OSTMnCTL.OSTMnMD0 = 1).

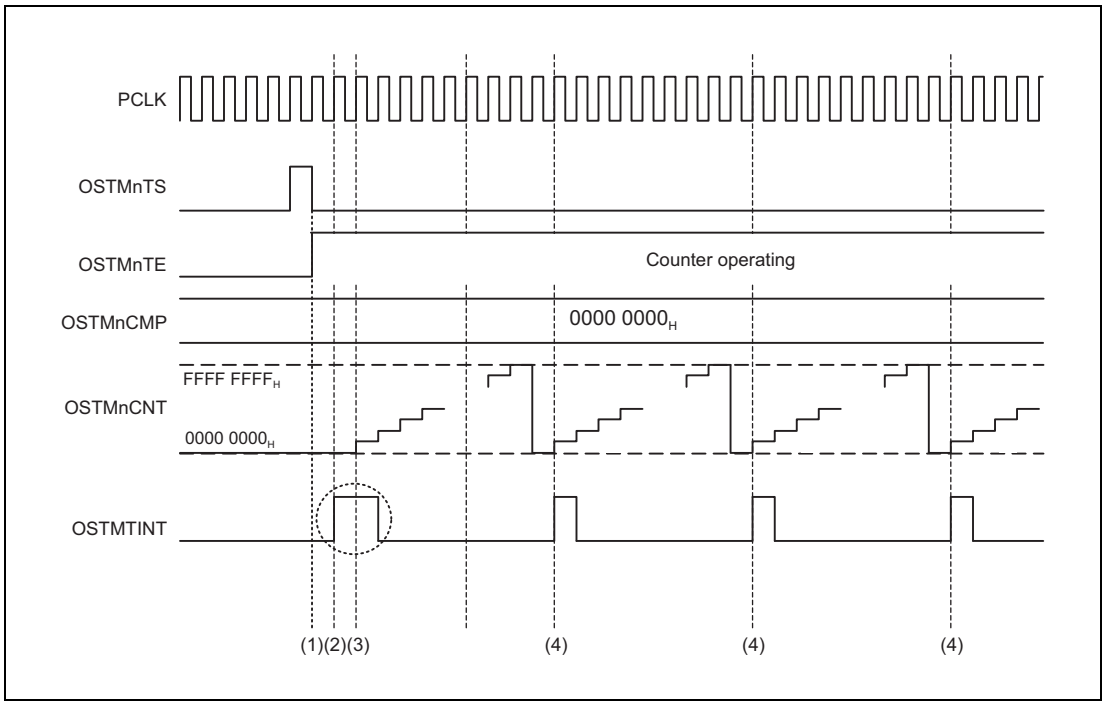


Figure 23.7 Timing Diagram when OSTMnCMP = 0000 0000$_H$ in Free-Run Compare Mode

The timing diagram above shows the following operations.

(1) Once the counter starts, it counts up from 0000 0000$_H$ to FFFF FFFF$_H$.

(2) An OSTMTINT interrupt request is generated when counting starts.

(3) If the current counter value matches OSTMnCMP, an OSTMTINT interrupt request is generated. If OSTMnCMP = 0000 0000_H as shown above, OSTMTINT is generated over two clock cycles.

(4) An OSTMTINT interrupt request is generated every (FFFF $FFFF_H$ +1) clock cycles.

When interrupts on starting of the counter are disabled, no interrupt is generated when counting starts.

23.4.3.3 Setting Procedure for Free-Run Compare Mode

The setting procedure in free-run compare mode after reset release is described below:

Setting procedure

(1) Set the compare value in the OSTMnCMP register.

(2) Select free-run compare mode by setting the OSTMnCTL.OSTMnMD1 bit to 1.

(3) Enable or disable interrupts when counting starts by the OSTMnCTL.OSTMnMD0 bit.

Section 24 Timer Array Unit B (TAUB)

This section contains a generic description of the timer array unit B (TAUB).

The first part of this section describes the RH850/F1M specific features such as the number of units and the register base addresses. The remainder of the section describes the functions and registers of the TAUB.

24.1 Features of RH850/F1M TAUB

24.1.1 Number of Units and Channels

This microcontroller has the following number of TAUB units.

Table 24.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	1	2	
Name	TAUBn (n = 0)	TAUBn (n = 0, 1)	

TAUBn has the following number of channels of timers.

Table 24.2 TAUBn Unit Configurations and Channels

Unit Name (Channel Name) TAUBn	Channels per Unit	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
TAUB0	16	√	√	√
TAUB1	16	—	√	√

Table 24.3 Indices

Index	Description
n	Throughout this section, the individual TAUB units are identified by the index “n”; for example, TAUBnTOM is the TAUBn channel output mode register.
m	The TAUB has 16 channels. Throughout this section, the individual channels are identified by the index “m” (m = 0 to 15), thus a certain channel is denoted as CHm. The even numbered channels (m = 0, 2, 4, 6, 8, 10, 12, 14) are denoted as CHm_even. The odd numbered channels (m = 1, 3, 5, 7, 9, 11, 13, 15) are denoted as CHm_odd.

24.1.2 Register Base Address

TAUBn base addresses are listed in the following table.

TAUBn register addresses are given as offsets from the base addresses.

Table 24.4 Register Base Addresses

Name	Base Address
<TAUB0_base>	FFE3 0000_H
<TAUB1_base>	FFE3 1000_H

24.1.3 Clock Supply

The TAUBn clock supply is shown in the following table.

Table 24.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
TAUBn	PCLK	CKSCLK_IPERI2
	Register access clock	CKSCLK_IPERI2

24.1.4 Interrupt Requests

TAUBn interrupt requests are listed in the following table.

Table 24.6 Interrupt Requests (1/2)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
TAUB0			
INTTAUB0I0	Channel 0 interrupt	142	33
INTTAUB0I1	Channel 1 interrupt	143	92
INTTAUB0I2	Channel 2 interrupt	144	34
INTTAUB0I3	Channel 3 interrupt	145	93
INTTAUB0I4	Channel 4 interrupt	146	35
INTTAUB0I5	Channel 5 interrupt	147	94
INTTAUB0I6	Channel 6 interrupt	148	36
INTTAUB0I7	Channel 7 interrupt	149	95
INTTAUB0I8	Channel 8 interrupt	150	96
INTTAUB0I9	Channel 9 interrupt	151	37
INTTAUB0I10	Channel 10 interrupt	152	97
INTTAUB0I11	Channel 11 interrupt	153	38
INTTAUB0I12	Channel 12 interrupt	154	98
INTTAUB0I13	Channel 13 interrupt	155	39
INTTAUB0I14	Channel 14 interrupt	156	99
INTTAUB0I15	Channel 15 interrupt	157	40
TAUB1			
INTTAUB1I0	Channel 0 interrupt	256	52
INTTAUB1I1	Channel 1 interrupt	257	115
INTTAUB1I2	Channel 2 interrupt	258	53
INTTAUB1I3	Channel 3 interrupt	259	116
INTTAUB1I4	Channel 4 interrupt	260	54
INTTAUB1I5	Channel 5 interrupt	261	117
INTTAUB1I6	Channel 6 interrupt	262	55
INTTAUB1I7	Channel 7 interrupt	263	118
INTTAUB1I8	Channel 8 interrupt	264	119
INTTAUB1I9	Channel 9 interrupt	265	56
INTTAUB1I10	Channel 10 interrupt	266	120
INTTAUB1I11	Channel 11 interrupt	267	57
INTTAUB1I12	Channel 12 interrupt	268	121

Table 24.6 Interrupt Requests (2/2)

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
INTTAUB1I13	Channel 13 interrupt	269	58
INTTAUB1I14	Channel 14 interrupt	270	122
INTTAUB1I15	Channel 15 interrupt	271	59

24.1.5 Reset Sources

TAUBn reset sources are listed in the following table. TAUBn is initialized by these reset sources.

Table 24.7 Reset Sources

Unit Name	Reset Source
TAUBn	All reset sources (ISORES)

24.1.6 External Input/Output Signals

External input/output signals of TAUBn are listed below.

Table 24.8 External Input/Output Signals (1/2)

TAUB Signal	Description	Alternative Port Pin Signal
TAUB0		
TAUBTTIN0	Channel 0 input*1	TAUB0I0
TAUBTTIN1	Channel 1 input*1	TAUB0I1
TAUBTTIN2	Channel 2 input*1	TAUB0I2
TAUBTTIN3	Channel 3 input*1	TAUB0I3
TAUBTTIN4	Channel 4 input*1	TAUB0I4
TAUBTTIN5	Channel 5 input*1	TAUB0I5
TAUBTTIN6	Channel 6 input*1	TAUB0I6
TAUBTTIN7	Channel 7 input*1	TAUB0I7
TAUBTTIN8	Channel 8 input*1	TAUB0I8
TAUBTTIN9	Channel 9 input*1	TAUB0I9
TAUBTTIN10	Channel 10 input*1	TAUB0I10
TAUBTTIN11	Channel 11 input*1	TAUB0I11
TAUBTTIN12	Channel 12 input*1	TAUB0I12
TAUBTTIN13	Channel 13 input*1	TAUB0I13
TAUBTTIN14	Channel 14 input*1	TAUB0I14
TAUBTTIN15	Channel 15 input*1	TAUB0I15
TAUBTTOUT0	Channel 0 output	TAUB0O0
TAUBTTOUT1	Channel 1 output	TAUB0O1
TAUBTTOUT2	Channel 2 output	TAUB0O2
TAUBTTOUT3	Channel 3 output	TAUB0O3
TAUBTTOUT4	Channel 4 output	TAUB0O4
TAUBTTOUT5	Channel 5 output	TAUB0O5
TAUBTTOUT6	Channel 6 output	TAUB0O6

Table 24.8 External Input/Output Signals (2/2)

TAUB Signal	Description	Alternative Port Pin Signal
TAUBTTOUT7	Channel 7 output	TAUB0O7
TAUBTTOUT8	Channel 8 output	TAUB0O8
TAUBTTOUT9	Channel 9 output	TAUB0O9
TAUBTTOUT10	Channel 10 output	TAUB0O10
TAUBTTOUT11	Channel 11 output	TAUB0O11
TAUBTTOUT12	Channel 12 output	TAUB0O12
TAUBTTOUT13	Channel 13 output	TAUB0O13
TAUBTTOUT14	Channel 14 output	TAUB0O14
TAUBTTOUT15	Channel 15 output	TAUB0O15
TAUB1		
TAUBTTIN0	Channel 0 input*1	TAUB1I0
TAUBTTIN1	Channel 1 input*1	TAUB1I1
TAUBTTIN2	Channel 2 input*1	TAUB1I2
TAUBTTIN3	Channel 3 input*1	TAUB1I3
TAUBTTIN4	Channel 4 input*1	TAUB1I4
TAUBTTIN5	Channel 5 input*1	TAUB1I5
TAUBTTIN6	Channel 6 input*1	TAUB1I6
TAUBTTIN7	Channel 7 input*1	TAUB1I7
TAUBTTIN8	Channel 8 input*1	TAUB1I8
TAUBTTIN9	Channel 9 input*1	TAUB1I9
TAUBTTIN10	Channel 10 input*1	TAUB1I10
TAUBTTIN11	Channel 11 input*1	TAUB1I11
TAUBTTIN12	Channel 12 input*1	TAUB1I12
TAUBTTIN13	Channel 13 input*1	TAUB1I13
TAUBTTIN14	Channel 14 input*1	TAUB1I14
TAUBTTIN15	Channel 15 input*1	TAUB1I15
TAUBTTOUT0	Channel 0 output	TAUB1O0
TAUBTTOUT1	Channel 1 output	TAUB1O1
TAUBTTOUT2	Channel 2 output	TAUB1O2
TAUBTTOUT3	Channel 3 output	TAUB1O3
TAUBTTOUT4	Channel 4 output	TAUB1O4
TAUBTTOUT5	Channel 5 output	TAUB1O5
TAUBTTOUT6	Channel 6 output	TAUB1O6
TAUBTTOUT7	Channel 7 output	TAUB1O7
TAUBTTOUT8	Channel 8 output	TAUB1O8
TAUBTTOUT9	Channel 9 output	TAUB1O9
TAUBTTOUT10	Channel 10 output	TAUB1O10
TAUBTTOUT11	Channel 11 output	TAUB1O11
TAUBTTOUT12	Channel 12 output	TAUB1O12
TAUBTTOUT13	Channel 13 output	TAUB1O13
TAUBTTOUT14	Channel 14 output	TAUB1O14
TAUBTTOUT15	Channel 15 output	TAUB1O15

Note 1. When channel input pins are to be used, noise filters must be set for the corresponding port pin functions. For details, see **Section 2.12, Noise Filter & Edge/Level Detector**.

24.2 Overview

24.2.1 Functional Overview

The TAUB has the following functions:

- Independent channel operation function (operated using a single channel)
- Synchronous channel operation function (operated using a master channel and multiple slave channels)

The TAUB is used to perform various count or timer operations and to output different signals depending on the results of the operation. It has 16 channels, each of which contains a prescaler for count clock generation, 16 bit counter TAUBnCNTm and 16 bit data register TAUBnCDRm to hold a count start value and compare value respectively.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate either independently or in combination with other channels (synchronously). For a combination of one master channel and more than one slave channel, the slave channels depend on the master channel.

When a channel is operated independently, it can be operated independent of all other channels. The synchronous operation function is implemented to combine channel groups (consisted of a master channel and slave channels).

Several rules apply to the settings of channels.

24.2.2 Terms

In this section, the following terms are used:

- Independent channel operation function/synchronous channel operation function

  TAUB has 16 channels, and provides an independent channel operation function whereby individual channels operate independently and a synchronous channel operation function whereby multiple channels operate in combination.

  – With the independent operation function, any channel can be used independently of all other channels.
  – With the synchronous channel operation function, channel groups (consisting of a master channel and slave channels) can operated in combination.

Several rules apply to the settings of channels.

- Channel group

  In the synchronous channel operation function, all channels that depend on each other are referred to as a “channel group”.
  A channel group has one master channel and one or more slave channels.

- Upper/lower channel

  Depending on the channel number m, a channel with a smaller number or with a larger number is referred to as “upper” or “lower” channel, respectively.

  – Upper channel: Channel with a smaller channel number
  – Lower channel: Channel with a larger channel number

  Example:
  For channel 5, channel 3 is an upper channel and channel 9 is a lower channel. Channel 0 is the highest channel and channel 15 is the lowest channel.

The following describes the functional blocks:

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK3 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- TAUBTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operating mode (selected with the TAUBnCMORm.TAUBnMD[4:0] bits)
- Counter start enable (TAUBnTS. TAUBnTSm) and counter stop (TAUBnTT. TAUBnTTm) When counter start is enabled, status flag TAUBnTE. TAUBnTEm is set.
- Count direction (up/down) (can be controlled by the master channel)

Trigger selector

The counter starts automatically when it is enabled (TAUBnTE.TAUBnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger:

- INTTAUBnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUBTTOUTm generation unit.

Simultaneous rewrite controller

Simultaneous rewrite control is a special function that can be used in synchronous operation functions. The data registers of all channels in a channel group (TAUBnCDRm) can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUBnTO Controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

24.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 24.9 Functional List of TAUB Operations

Operation Function	Setting Example
Independent Channel Operation Functions	**Section 24.12**
Interval Timer Function	**Section 24.12.1**
TAUBTTINm Input Interval Timer Function	**Section 24.12.2**
Clock Divide Function	**Section 24.12.3**
External Event Count Function	**Section 24.12.4**
One-Pulse Output Function	**Section 24.12.5**
TAUBTTINm Input Pulse Interval Measurement Function	**Section 24.12.6**
TAUBTTINm Input Signal Width Measurement Function	**Section 24.12.7**
TAUBTTINm Input Position Detection Function	**Section 24.12.8**
TAUBTTINm Input Period Count Detection Function	**Section 24.12.9**
TAUBTTINm Input Pulse Interval Judgment Function	**Section 24.12.10**
TAUBTTINm Input Signal Width Judgment Function	**Section 24.12.11**
Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)	**Section 24.12.12**
Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)	**Section 24.12.13**
Independent Channel Simultaneous Rewrite Functions	**Section 24.13**
Simultaneous Rewrite Trigger Generation Function Type 1	**Section 24.13.1**
Synchronous Channel Operation Functions	**Section 24.14**
PWM Output Function	**Section 24.14.1**
One-Shot Pulse Output Function	**Section 24.14.2**
Delay Pulse Output Function	**Section 24.14.3**
A/D Conversion Trigger Output Function Type 1	**Section 24.14.4**
Triangular Wave PWM Output Function	**Section 24.14.5**
Triangular Wave PWM Output Function with Dead Time	**Section 24.14.6**
A/D Conversion Trigger Output Function Type 2	**Section 24.14.7**

24.2.4 Input/Output Interrupt Request Signals

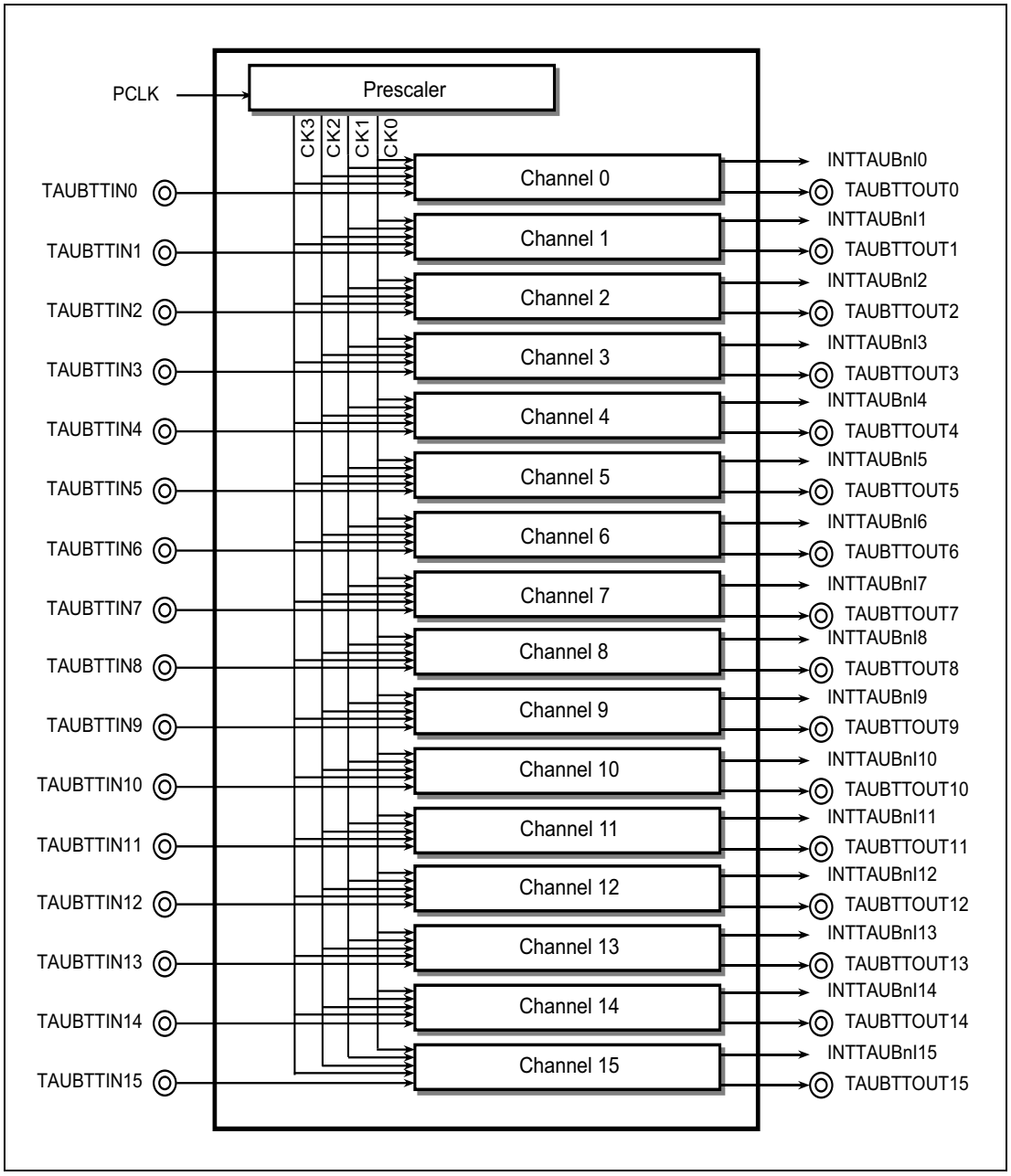


Figure 24.1 TAUB Input/Output and Interrupt Request Signals

24.2.5 Block Diagram

The following figure shows the main components of the TAUB.

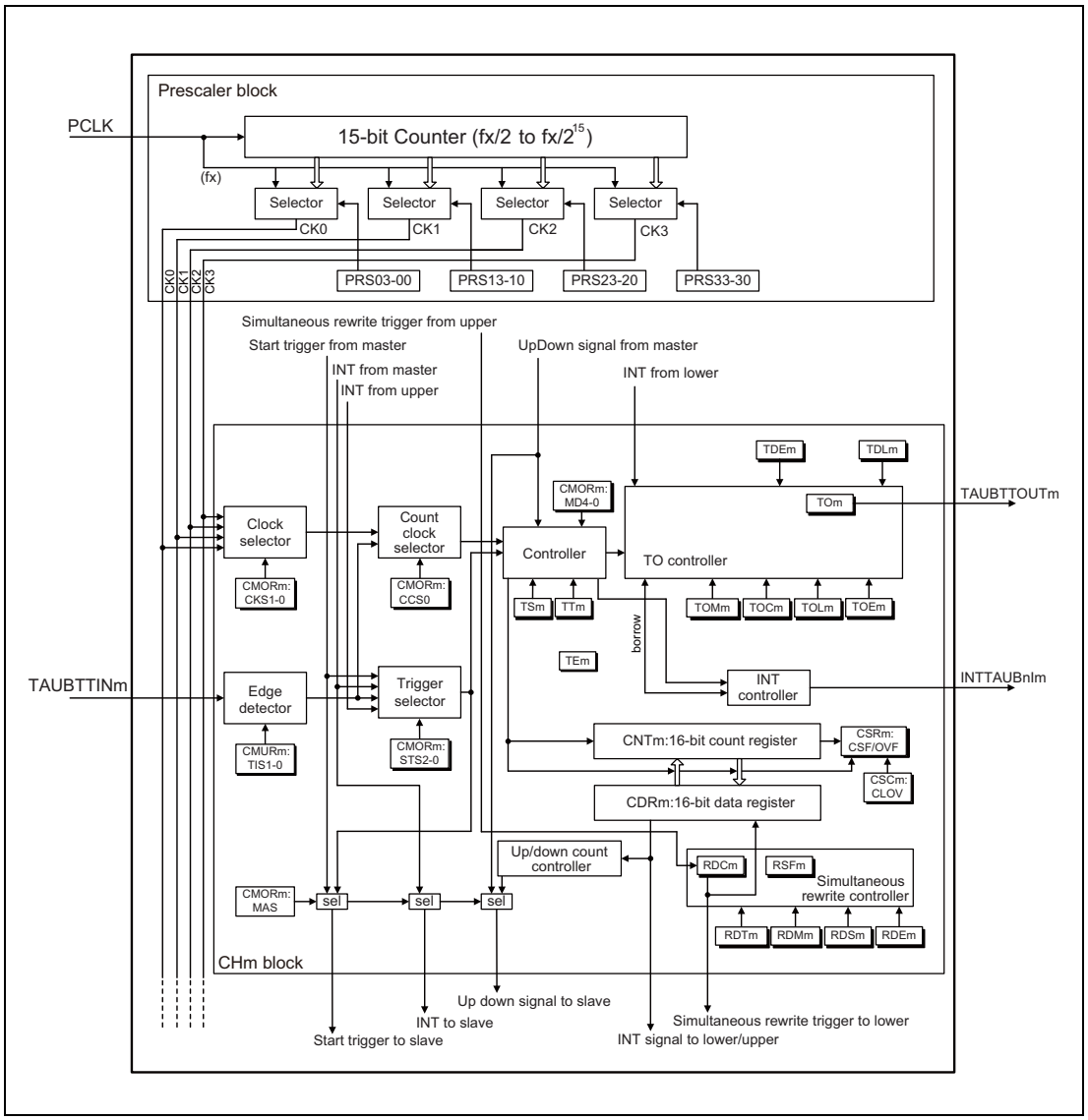


Figure 24.2 Block Diagram of the TAUB

The prefix “TAUBn” has been omitted from the register names for the sake of clarity in the above figure.

24.2.6 Description of Blocks

The following describes the functional blocks:

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK3 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- TAUBTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUBnCMORm.TAUBnMD[4:0])
- Counter start enable (TAUBnTS.TAUBnTSm) and counter stop (TAUBnTT.TAUBnTTm)
  When counter start is enabled, status flag TAUBnTE.TAUBnTEm is set.
- Count direction (up/down) (can be controlled by the master channel)

Trigger selector

Depending on the selected operation mode, the counter starts automatically when it is enabled (TAUBnTE.TAUBnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger:

- Input of TAUBTTINm valid edge
- INTTAUBnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUBTTOUTm generation unit

Simultaneous rewrite controller

Simultaneous rewrite control is a special function that can be used in synchronous operation functions. The data registers (TAUBnCDRm) of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUBnTO controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

24.3 Registers

24.3.1 List of Registers

TAUB registers are listed in the following table.

For details about <TAUBn_base>, see **Section 24.1.2, Register Base Address**.

Table 24.10 List of Registers

Module Name	Register Name	Symbol	Address
TAUBn prescaler registers			
TAUBn	TAUBn prescaler clock select register	TAUBnTPS	<TAUBn_base> + 240_H
TAUBn control registers			
TAUBn	TAUBn channel data register m	TAUBnCDRm	<TAUBn_base> + 0_H + m × 4_H
TAUBn	TAUBn channel counter register m	TAUBnCNTm	<TAUBn_base> + 80_H + m × 4_H
TAUBn	TAUBn channel mode OS register m	TAUBnCMORm	<TAUBn_base> + 200_H + m × 4_H
TAUBn	TAUBn channel mode user register m	TAUBnCMURm	<TAUBn_base> + $C0_H$ + m × 4_H
TAUBn	TAUBn channel status register m	TAUBnCSRm	<TAUBn_base> + 140_H + m × 4_H
TAUBn	TAUBn channel status clear trigger register m	TAUBnCSCm	<TAUBn_base> + 180_H + m × 4_H
TAUBn	TAUBn channel start trigger register	TAUBnTS	<TAUBn_base> + $1C4_H$
TAUBn	TAUBn channel enable status register	TAUBnTE	<TAUBn_base> + $1C0_H$
TAUBn	TAUBn channel stop trigger register	TAUBnTT	<TAUBn_base> + $1C8_H$
TAUBn output registers			
TAUBn	TAUBn channel output enable register	TAUBnTOE	<TAUBn_base> + $5C_H$
TAUBn	TAUBn channel output register	TAUBnTO	<TAUBn_base> + 58_H
TAUBn	TAUBn channel output mode register	TAUBnTOM	<TAUBn_base> + 248_H
TAUBn	TAUBn channel output configuration register	TAUBnTOC	<TAUBn_base> + $24C_H$
TAUBn	TAUBn channel output active level register	TAUBnTOL	<TAUBn_base> + 040_H
TAUBn	TAUBn channel dead time output enable register	TAUBnTDE	<TAUBn_base> + 250_H
TAUBn	TAUBn channel dead time output level register	TAUBnTDL	<TAUBn_base> + 54_H
TAUBn reload data registers			
TAUBn	TAUBn channel reload data enable register	TAUBnRDE	<TAUBn_base> + 260_H
TAUBn	TAUBn channel reload data mode register	TAUBnRDM	<TAUBn_base> + 264_H
TAUBn	TAUBn channel reload data control CH select register	TAUBnRDS	<TAUBn_base> + 268_H
TAUBn	TAUBn channel reload data control register	TAUBnRDC	<TAUBn_base> + $26C_H$
TAUBn	TAUBn channel reload data trigger register	TAUBnRDT	<TAUBn_base> + 44_H
TAUBn	TAUBn channel reload status register	TAUBnRSF	<TAUBn_base> + 48_H
TAUBn emulation register			
TAUBn	TAUBn emulation register	TAUBnEMU	<TAUBn_base> + 290_H

24.3.2 Details of TAUBn Prescaler Registers

24.3.2.1 TAUBnTPS — TAUBn Prescaler Clock Select Register

This register specifies the CK0, CK1, CK2, and CK3 clocks for all channels of the PCLK prescaler.

Access: This register can be read or written in 16-bit units.

Address: <TAUBn_base> + 240_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnPRS3[3:0]				TAUBnPRS2[3:0]				TAUBnPRS1[3:0]				TAUBnPRS0[3:0]			
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.11 TAUBnTPS Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	TAUBnPRS3 [3:0]	Specifies the CK3 clock.

TAUBnPRS3[3:0]	CK3 clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

These bits can only be rewritten when all counters using CK3 are stopped (TAUBnTE.TAUBnTEm = 0).

Table 24.11 TAUBnTPS Register Contents (2/3)

Bit Position	Bit Name	Function
11 to 8	TAUBnPRS2 [3:0]	Specifies the CK2 clock.

TAUBnPRS2[3:0]	CK2 Clock
0000_B	$PCLK/2^0$
0001_B	$PCLK/2^1$
0010_B	$PCLK/2^2$
0011_B	$PCLK/2^3$
0100_B	$PCLK/2^4$
0101_B	$PCLK/2^5$
0110_B	$PCLK/2^6$
0111_B	$PCLK/2^7$
1000_B	$PCLK/2^8$
1001_B	$PCLK/2^9$
1010_B	$PCLK/2^{10}$
1011_B	$PCLK/2^{11}$
1100_B	$PCLK/2^{12}$
1101_B	$PCLK/2^{13}$
1110_B	$PCLK/2^{14}$
1111_B	$PCLK/2^{15}$

These bits can only be rewritten when all counters using CK2 are stopped (TAUBnTE.TAUBnTEm = 0).

7 to 4	TAUBnPRS1 [3:0]	Specifies the CK1 clock.

TAUBnPRS1[3:0]	CK1 Clock
0000_B	$PCLK/2^0$
0001_B	$PCLK/2^1$
0010_B	$PCLK/2^2$
0011_B	$PCLK/2^3$
0100_B	$PCLK/2^4$
0101_B	$PCLK/2^5$
0110_B	$PCLK/2^6$
0111_B	$PCLK/2^7$
1000_B	$PCLK/2^8$
1001_B	$PCLK/2^9$
1010_B	$PCLK/2^{10}$
1011_B	$PCLK/2^{11}$
1100_B	$PCLK/2^{12}$
1101_B	$PCLK/2^{13}$
1110_B	$PCLK/2^{14}$
1111_B	$PCLK/2^{15}$

These bits can only be rewritten when all counters using CK1 are stopped (TAUBnTE.TAUBnTEm = 0).

Table 24.11 TAUBnTPS Register Contents (3/3)

Bit Position	Bit Name	Function
3 to 0	TAUBnPRS0 [3:0]	Specifies the CK0 clock.

TAUBnPRS0[3:0]	CK0 Clock
0000_B	$PCLK/2^0$
0001_B	$PCLK/2^1$
0010_B	$PCLK/2^2$
0011_B	$PCLK/2^3$
0100_B	$PCLK/2^4$
0101_B	$PCLK/2^5$
0110_B	$PCLK/2^6$
0111_B	$PCLK/2^7$
1000_B	$PCLK/2^8$
1001_B	$PCLK/2^9$
1010_B	$PCLK/2^{10}$
1011_B	$PCLK/2^{11}$
1100_B	$PCLK/2^{12}$
1101_B	$PCLK/2^{13}$
1110_B	$PCLK/2^{14}$
1111_B	$PCLK/2^{15}$

These bits can only be rewritten when all counters using CK0 are stopped (TAUBnTE.TAUBnTEm = 0).

NOTE

The TAUBn clock input PCLK is specified in the first part of this section, **Section 24.1.3, Clock Supply**.

24.3.3 Details of TAUBn Control Registers

24.3.3.1 TAUBnCDRm — TAUBn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operation mode specified in TAUBnCMORm.TAUBnMD[4:1].

Access: This register can be read or written in 16-bit units.
- When this register functions as a capture register, only reading is possible. Write operation is ignored.
- When this register functions as a compare register, reading and writing is possible.

Address: <TAUBn_base> + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.12 TAUBnCDRm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnCDR [15:0]	Data register for the capture/compare value.

24.3.3.2 TAUBnCNTm — TAUBn Channel Counter Register

This register is the channel m counter register.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUBn_base> + 80_H + m × 4_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.13 TAUBnCNTm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnCNT [15:0]	16-bit counter value.

The read value depends on the counter, the operation mode change, and the values of the TAUBnTS.TAUBnTSm and TAUBnTT.TAUBnTTm bits.

The initial counter read value depends on the operation mode and how the counter was stopped:

- by a reset
- by a counter stop trigger (TAUBnTT.TAUBnTTm = 1)

The following table lists the initial counter read values after the counter has stopped (TAUBnTE.TAUBnTEm = 0) and re-enabled (TAUBnTS.TAUBnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUBnTS.TAUBnTSm = 1) for modes where the counter waits for a start trigger.

Table 24.14 TAUBnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUBnCNTm		
		Start Value*1	After Stop Trigger	After One Count
Interval timer mode	Count down	$FFFF_H$	Stop value	—
Judge mode	Count down	$FFFF_H$	Stop value	—
Capture mode	Count up	0000_H	Stop value	—
Event count mode	Count down	$FFFF_H$	Stop value	—
One-count mode	Count down	$FFFF_H$	Stop value	Stop value
Capture and one-count mode	Count up	0000_H	Stop value	Capture value + 1 (TAUBnCDRm)
Judge and one-count mode	Count down	$FFFF_H$	Stop value	TAUBnCNTm value – 1
Count-up/-down mode	Count up/down	$FFFF_H$	Stop value	—
Pulse one-count mode	Count down	$FFFF_H$	Stop value	0000_H
Count capture mode	Count up	0000_H	Stop value	—
Gate count mode	Count down	$FFFF_H$	Stop value	Stop value
Capture and gate count mode	Count up	0000_H	Stop value	Stop value

Note 1. The value set for TAUBnCNTm when operation mode is changed after reset is deasserted.

24.3.3.3 TAUBnCMORm — TAUBn Channel Mode OS Register

This register controls channel m operation.

Access: This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

Address: <TAUBn_base> + 200_H + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.15 TAUBnCMORm Register Contents (1/3)

Bit Position	Bit Name	Function
15, 14	TAUBnCKS [1:0]	Selects the operation clock. The operation clock is used for the TAUBTTINm input edge detection circuit. TAUBnCNTm can also be used as the count clock depending on the setting of theTAUBnCMORm.TAUBnCCS0 bit.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Selects the count clock for TAUBnCNTm counter:
11	TAUBnMAS	Specifies the channel as master or slave channel during synchronous channel operation: 0: Slave 1: Master This bit is only valid for even channels (CHm_even). For odd channels (CHm_odd), it is fixed to 0.

TAUBn CKS1	TAUBn CKS0	Selected Operation Clock
0	0	CK0
0	1	CK1
1	0	CK2
1	1	CK3

TAUBnCCS0	Selected Count Clock
0	Operation clock as specified by TAUBnCMORm.TAUBnCKS[1:0].
1	Valid edge of TAUBTTINm input signal

Table 24.15 TAUBnCMORm Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUBnSTS [2:0]</td><td>Selects the external start trigger:
<table>
<tr><th>TAUBn STS2</th><th>TAUBn STS1</th><th>TAUBn STS0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of the TAUBTTINm input signal. TAUBnCMURm.TAUBnTIS[1:0] specifies the valid edge.</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>The valid edge of the TAUBTTINm input signal is used as the start trigger and the reverse edge is used as the stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INTTAUBnIm is the start trigger of the master channel.</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>INTTAUBnIm of upper channel (m – 1) is the start trigger regardless of master setting</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Dead-time output signal of the TAUBTTOUTm generation unit</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Up/down output trigger signal of the master channel.</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>TAUBnCOS [1:0]</td><td>Specifies when the capture register TAUBnCDRm and the overflow flag TAUBnCSRm.TAUBnOVF of channel m are updated.
These bits are only valid if channel m is in capture function (capture mode and capture & one-count mode).
<table>
<tr><th>TAUBn COS1</th><th>TAUBn COS0</th><th>TAUBnCDRm</th><th>TAUBnCSRm.TAUBnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of a TAUBTTINm input valid edge.</td><td>Updated (cleared or set) upon detection of a TAUBTTINm input valid edge:
• If a counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is set.
• If no counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is cleared.</td></tr>
<tr><td>0</td><td>1</td><td>Set upon counter overflow and cleared by setting TAUBnCSCm.TAUBnCLOV to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated upon detection of a TAUBTTINm input valid edge and upon counter overflow:
• TAUBTTINm input valid edge: Counter value is written to TAUBnCDRm
• Overflow: $FFFF_H$ is written to TAUBnCDRm. The next TAUBTTINm input valid edge detection is ignored.</td><td>Not set.</td></tr>
<tr><td>1</td><td>1</td><td>Set upon counter overflow and cleared by setting TAUBnCSCm.TAUBnCLOV to 1.</td></tr>
</table></td></tr>
<tr><td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 24.15 TAUBnCMORm Register Contents (3/3)

Bit Position	Bit Name	Function
4 to 0	TAUBnMD [4:0]	Specifies the operation mode. For details, refer to the settings for individual functions.

TAUBn MD4	TAUBn MD3	TAUBn MD2	TAUBn MD1	TAUBn MD0	Description
0	0	0	0	1/0	Interval timer mode
0	0	0	1	1/0	Judge mode
0	0	1	0	1/0	Capture mode
0	0	1	1	0	Event count mode
0	1	0	0	1/0	One-count mode
0	1	0	1	1/0	Setting prohibited
0	1	1	0	0	Capture and one-count mode
0	1	1	1	1/0	Judge and one-count mode
1	0	0	0	0	Setting prohibited
1	0	0	1	0	Count-up/-down mode
1	0	1	0	1/0	Pulse one-count mode
1	0	1	1	1/0	Count capture mode
1	1	0	0	0	Gate count mode
1	1	0	1	0	Capture and gate count mode

Mode	Role of TAUBnMD0 Bit
Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUBnIm is output at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUBnIm is not output. 1: INTTAUBnIm is output.
Event count mode Count-up/-down mode	This bit should be set to 0 (INTTAUBnIm signal is not output at the beginning of count operation).
One-count mode Pulse one-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. **CAUTION** • In one-count mode, INTTAUBnIm signal is not output at the beginning of count operation. • In pulse one-count mode, INTTAUBnIm signal is output at the beginning of count operation.
Gate count mode	This bit should be set to 0 (start trigger detection during counting is disabled).
Capture and one-count mode Capture and gate count mode	This bit should be set to 0. **CAUTION** INTTAUBnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.
Judge mode Judge and one-count mode	Specifies INTTAUBnIm output timing. 0: When TAUBnCNTm ≤ TAUBnCDRm 1: When TAUBnCNTm > TAUBnCDRm

24.3.3.4 TAUBnCMURm — TAUBn Channel Mode User Register

This register specifies the type of valid edge detection used for the TAUBTTINm input.

Access: This register can be read or written in 8-bit units.

Address: <TAUBn_base> + $C0_H$ + m × 4_H

Value after reset: 00_H

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

Table 24.16 TAUBnCMURm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS [1:0]	Specifies the valid edge of the TAUBTTINm input:

TAUBn TIS1	TAUBn TIS0	Description
0	0	Falling edge
0	1	Rising edge
1	0	Rising and falling edges (low-width measurement selection). Start trigger: falling edge Stop trigger (capture): rising edge
1	1	Rising and falling edges (high-width measurement selection). Start trigger: rising edge Stop trigger (capture): falling edge

- Edge detection for TAUBTTINm input signals is performed based on the operation clock selected by TAUBnCMORm.TAUBnCKS[1:0].

24.3.3.5 TAUBnCSRm — TAUBn Channel Status Register

This register indicates the count direction and the overflow status of the counter for channel m.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUBn_base> + 140_H + m × 4_H

Value after reset: 00_H

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

Table 24.17 TAUBnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	TAUBnCSF	Indicates the count direction: 0: Counts up 1: Counts down The read value of this bit is only valid in the following mode: • Up/Down Count mode
0	TAUBnOVF	Indicates the counter overflow status: 0: No overflow occurred 1: Overflow occurred This bit is used only in the following modes: • Capture mode • Capture and one-count mode The function of this bit depends on the setting of control bits TAUBnCMORm.TAUBnCOS[1:0].

24.3.3.6 TAUBnCSCm — TAUBn Channel Status Clear Register

This register is a trigger register for clearing the overflow flag TAUBnCSRm.TAUBnOVF of channel m.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H

Address: <TAUBn_base> + 180_H + m × 4_H

Value after reset: 00_H

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

Table 24.18 TAUBnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAUBnCLOV	0: No function 1: Clears the overflow flag TAUBnCSRm.TAUBnOVF

24.3.3.7 TAUBnTS — TAUBn Channel Start Trigger Register

This register enables the counter for each channel.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H

Address: <TAUBn_base> + $1C4_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnTS15	TAUBnTS14	TAUBnTS13	TAUBnTS12	TAUBnTS11	TAUBnTS10	TAUBnTS09	TAUBnTS08	TAUBnTS07	TAUBnTS06	TAUBnTS05	TAUBnTS04	TAUBnTS03	TAUBnTS02	TAUBnTS01	TAUBnTS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 24.19 TAUBnTS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTSm	Enables the counter for channel m: 0: No function 1: Enables the counter and sets TAUBnTE.TAUBnTEm = 1. TAUBnTE.TAUBnTEm = 1 only enables counter. Whether the counter starts depends on the selected operation mode.

24.3.3.8 TAUBnTE — TAUBn Channel Enable Status Register

This register indicates whether counter is enabled or disabled.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUBn_base> + $1C0_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnTE15	TAUBnTE14	TAUBnTE13	TAUBnTE12	TAUBnTE11	TAUBnTE10	TAUBnTE09	TAUBnTE08	TAUBnTE07	TAUBnTE06	TAUBnTE05	TAUBnTE04	TAUBnTE03	TAUBnTE02	TAUBnTE01	TAUBnTE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.20 TAUBnTE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTEm	Indicates whether counter for channel m is enabled or disabled: 0: Counter disabled 1: Counter enabled Setting TAUBnTS.TAUBnTSm to 1 sets this bit to 1. Setting TAUBnTT.TAUBnTTm to 1 resets this bit to 0.

24.3.3.9 TAUBnTT — TAUBn Channel Stop Trigger Register

This register stops the counter for each channel.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H.

Address: <TAUBn_base> + $1C8_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnTT15	TAUBnTT14	TAUBnTT13	TAUBnTT12	TAUBnTT11	TAUBnTT10	TAUBnTT09	TAUBnTT08	TAUBnTT07	TAUBnTT06	TAUBnTT05	TAUBnTT04	TAUBnTT03	TAUBnTT02	TAUBnTT01	TAUBnTT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 24.21 TAUBnTT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTTm	Stops the counter of channel m: 0: No function 1: Stops the counter and resets TAUBnTE.TAUBnTEm. TAUBnCNTm, TAUBnTO.TAUBnTOm, and TAUBTTOUTm all retain the values they had before the counter was stopped.

24.3.4 Details of TAUBn Simultaneous Rewrite Registers

24.3.4.1 TAUBnRDE — TAUBn Channel Reload Data Enable Register

This register enables and disables simultaneous rewrite of the data register TAUBnCDRm/ TAUBnTOLm.

Access: This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0.

Address: <TAUBn_base> + 260_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDE15	TAUBn RDE14	TAUBn RDE13	TAUBn RDE12	TAUBn RDE11	TAUBn RDE10	TAUBn RDE09	TAUBn RDE08	TAUBn RDE07	TAUBn RDE06	TAUBn RDE05	TAUBn RDE04	TAUBn RDE03	TAUBn RDE02	TAUBn RDE01	TAUBn RDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.22 TAUBnRDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDEm	Enables/disables simultaneous rewrite of the data register of channel m: 0: Disables simultaneous rewrite 1: Enabled simultaneous rewrite

24.3.4.2 TAUBnRDS — TAUBn Channel Reload Data Control Channel Select Register

This register selects the control channel for simultaneous rewrite.

Access: This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0.

Address: <TAUBn_base> + 268_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDS15	TAUBn RDS14	TAUBn RDS13	TAUBn RDS12	TAUBn RDS11	TAUBn RDS10	TAUBn RDS09	TAUBn RDS08	TAUBn RDS07	TAUBn RDS06	TAUBn RDS05	TAUBn RDS04	TAUBn RDS03	TAUBn RDS02	TAUBn RDS01	TAUBn RDS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.23 TAUBnRDS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDSm	Specifies the channel to control the simultaneous rewrite trigger. 0: Master channel 1: Another upper channel

24.3.4.3 TAUBnRDM — TAUBn Channel Reload Data Mode Register

This register selects when the signal that controls simultaneous rewrite is generated.

Access: This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0.

Address: <TAUBn_base> + 264_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDM15	TAUBn RDM14	TAUBn RDM13	TAUBn RDM12	TAUBn RDM11	TAUBn RDM10	TAUBn RDM09	TAUBn RDM08	TAUBn RDM07	TAUBn RDM06	TAUBn RDM05	TAUBn RDM04	TAUBn RDM03	TAUBn RDM02	TAUBn RDM01	TAUBn RDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.24 TAUBnRDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDMm	Selects when the signal that triggers simultaneous rewrite is generated: 0: When the master channel counter starts counting 1: At the top of a triangular wave cycle These bits only apply when TAUBnRDE.TAUBnRDEm = 1 and TAUBnRDS.TAUBnRDSm = 0.

24.3.4.4 TAUBnRDC — TAUBn Channel Reload Data Control Register

This register specifies the channel that generates the INTTAUBnIm signal that triggers simultaneous rewrite.

Access: This register can be read or written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0

Address: <TAUBn_base> + $26C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDC15	TAUBn RDC14	TAUBn RDC13	TAUBn RDC12	TAUBn RDC11	TAUBn RDC10	TAUBn RDC09	TAUBn RDC08	TAUBn RDC07	TAUBn RDC06	TAUBn RDC05	TAUBn RDC04	TAUBn RDC03	TAUBn RDC02	TAUBn RDC01	TAUBn RDC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.25 TAUBnRDC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDCm	Specifies whether the channel generates a simultaneous rewrite trigger signal or not. 0: Does not operate as a simultaneous rewrite trigger channel. 1: Operates as a simultaneous rewrite trigger channel. These bits only apply when TAUBnRDE.TAUBnRDEm = 1 and TAUBnRDS.TAUBnRDSm = 1.

24.3.4.5 TAUBnRDT — TAUBn Channel Reload Data Trigger Register

This register triggers the simultaneous rewrite enabling state.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H.

Address: <TAUBn_base> + 044_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDT15	TAUBn RDT14	TAUBn RDT13	TAUBn RDT12	TAUBn RDT11	TAUBn RDT10	TAUBn RDT09	TAUBn RDT08	TAUBn RDT07	TAUBn RDT06	TAUBn RDT05	TAUBn RDT04	TAUBn RDT03	TAUBn RDT02	TAUBn RDT01	TAUBn RDT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 24.26 TAUBnRDT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDTm	Triggers the simultaneous rewrite enabling state: 0: No function. Writing 0 is ignored (the operation is not affected). 1: The simultaneous rewrite enabling flag (TAUBnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger. These bits only apply when: • TAUBnRDE.TAUBnRDEm = 1

24.3.4.6 TAUBnRSF — TAUBn Channel Reload Status Register

This flag register indicates the simultaneous rewrite status.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUBn_base> + 048_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RSF15	TAUBn RSF14	TAUBn RSF13	TAUBn RSF12	TAUBn RSF11	TAUBn RSF10	TAUBn RSF09	TAUBn RSF08	TAUBn RSF07	TAUBn RSF06	TAUBn RSF05	TAUBn RSF04	TAUBn RSF03	TAUBn RSF02	TAUBn RSF01	TAUBn RSF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.27 TAUBnRSF Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRSFm	Indicates the simultaneous rewrite status: 0: Indicates simultaneous rewrite is completed due to the generation of the simultaneous rewrite trigger. 1: Indicates the simultaneous rewrite trigger waiting state when simultaneous rewrite is enabled (TAUBnRDTm = 1).

24.3.5 Details of TAUBn Output Registers

24.3.5.1 TAUBnTOE — TAUBn Channel Output Enable Register

This register enables and disables independent channel output mode controlled by software.

Access: This register can be read or written in 16-bit units.

Address: <TAUBn_base> + $5C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOE15	TAUBn TOE14	TAUBn TOE13	TAUBn TOE12	TAUBn TOE11	TAUBn TOE10	TAUBn TOE09	TAUBn TOE08	TAUBn TOE07	TAUBn TOE06	TAUBn TOE05	TAUBn TOE04	TAUBn TOE03	TAUBn TOE02	TAUBn TOE01	TAUBn TOE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.28 TAUBnTOE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOEm	Enables/disables independent channel output mode: 0: Disables independent channel output mode (controlled by software) 1: Enables independent channel output mode

24.3.5.2 TAUBnTO — TAUBn Channel Output Register

This register specifies and reads the level of TAUBTTOUTm.

Access: This register can be read or written in 16-bit units.

Address: <TAUBn_base> + 58_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TO15	TAUBn TO14	TAUBn TO13	TAUBn TO12	TAUBn TO11	TAUBn TO10	TAUBn TO09	TAUBn TO08	TAUBn TO07	TAUBn TO06	TAUBn TO05	TAUBn TO04	TAUBn TO03	TAUBn TO02	TAUBn TO01	TAUBn TO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.29 TAUBnTO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOm	Specifies/reads the level of TAUBTTOUTm: 0: Low 1: High Only TAUBnTOm bits for which Independent Channel Output function is disabled (TAUBnTOEm = 0) can be written.

24.3.5.3 TAUBnTOM — TAUBn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

Address: <TAUBn_base> + 248_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOM15	TAUBn TOM14	TAUBn TOM13	TAUBn TOM12	TAUBn TOM11	TAUBn TOM10	TAUBn TOM09	TAUBn TOM08	TAUBn TOM07	TAUBn TOM06	TAUBn TOM05	TAUBn TOM04	TAUBn TOM03	TAUBn TOM02	TAUBn TOM01	TAUBn TOM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.30 TAUBnTOM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOMm	Specifies the channel output mode: 0: Independent channel output mode 1: Synchronous channel output mode

24.3.5.4 TAUBnTOC — TAUBn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUBnTOMm.

Access: This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

Address: <TAUBn_base> + $24C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOC15	TAUBn TOC14	TAUBn TOC13	TAUBn TOC12	TAUBn TOC11	TAUBn TOC10	TAUBn TOC09	TAUBn TOC08	TAUBn TOC07	TAUBn TOC06	TAUBn TOC05	TAUBn TOC04	TAUBn TOC03	TAUBn TOC02	TAUBn TOC01	TAUBn TOC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.31 TAUBnTOC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOCm	Specifies the output mode: 0: Operation mode 1 1: Operation mode 2 The output mode also depends on TAUBnTOM.TAUBnTOMm, as can be seen in the following table.

TOMm	TOCm	Description
0	0	Toggle mode: Toggles when INTTAUBnIm occurs.
	1	Set/reset mode: Set when INTTAUBnIm occurs upon count start and reset when INTTAUBnIm occurs due to detection of a match between TAUBnCNTm and TAUBnCDRm.
1	0	Synchronous Channel Operation Mode 1: Set when INT occurs on the master channel and reset when INT occurs on the slave channel.
	1	Synchronous Channel Operation Mode 2: Set when INTTAUBnIm occurs while the slave channel is counting down and reset when INTTAUBnIm occurs while the slave channel is counting up

24.3.5.5 TAUBnTOL — TAUBn Channel Output Level Register

This register specifies the output logic of the channel output bit (TAUBnTO.TAUBnTOm).

Access: This register can be read or written in 16-bit units.

Address: <TAUBn_base> + 040_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnTOL15	TAUBnTOL14	TAUBnTOL13	TAUBnTOL12	TAUBnTOL11	TAUBnTOL10	TAUBnTOL09	TAUBnTOL08	TAUBnTOL07	TAUBnTOL06	TAUBnTOL05	TAUBnTOL04	TAUBnTOL03	TAUBnTOL02	TAUBnTOL01	TAUBnTOL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.32 TAUBnTOL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOLm	Specifies the output logic of the channel m output bit (TAUBnTO.TAUBnTOm): 0: Positive logic (active high) 1: Negative logic (active low) The setting of these bits applies to all channel output modes other than independent channel output mode controlled by software and independent channel output mode 1.

24.3.6 Details of TAUBn Dead Time Output Registers

24.3.6.1 TAUBnTDE — TAUBn Channel Dead Time Output Enable Register

This register enables/disables dead time operation for each channel.

Access: This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

Address: <TAUBn_base> + 250_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TDE15	TAUBn TDE14	TAUBn TDE13	TAUBn TDE12	TAUBn TDE11	TAUBn TDE10	TAUBn TDE09	TAUBn TDE08	TAUBn TDE07	TAUBn TDE06	TAUBn TDE05	TAUBn TDE04	TAUBn TDE03	TAUBn TDE02	TAUBn TDE01	TAUBn TDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.33 TAUBnTDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTDEm	Enables/disables dead time control operation of channel m: 0: Disables dead time operation 1: Enables dead time operation The same settings must be set for the even and the odd slave channel that comprise a set. These bits only apply when: • TAUBnTOE.TAUBnTOEm, TAUBnTOM.TAUBnTOMm, TAUBnTOC.TAUBnTOCm = 1

24.3.6.2 TAUBnTDL — TAUBn Channel Dead Time Output Level Register

This register selects the phase period to which dead time is added.

Access: This register can be read or written in 16-bit units.

Address: <TAUBn_base> + 54_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TDL15	TAUBn TDL14	TAUBn TDL13	TAUBn TDL12	TAUBn TDL11	TAUBn TDL10	TAUBn TDL09	TAUBn TDL08	TAUBn TDL07	TAUBn TDL06	TAUBn TDL05	TAUBn TDL04	TAUBn TDL03	TAUBn TDL02	TAUBn TDL01	TAUBn TDL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.34 TAUBnTDL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTDLm	Selects the phase period to which dead time is added: 0: Positive phase period 1: Negative phase period These bits only apply when: • TAUBnTOE.TAUBnTOEm, TAUBnTOM.TAUBnTOMm, TAUBnTOC.TAUBnTOCm, TAUBnTDE.TAUBnTDEm = 1

24.3.7 TAUBn Emulation Register

24.3.7.1 TAUBnEMU — TAUBn Emulation Register

This register controls SVSTOP operations.

Access: This register can be read or written in 8-bit units.
Write to this register only when the counter is stopped (TAUBnTE.TAUBnTEm = 0) and when EPC.SVSTOP = 0.

Address: <TAUBn_base> + 290_H

Value after reset: 00_H

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

Table 24.35 TAUBnEMU Register Contents

Bit Position	Bit Name	Function
7	TAUBnSVSDIS	(When EPC.SVSTOP = 0) Regardless of the value of this bit (1/0), the count clock is continuously supplied when the debugger obtains the control of the microcontroller (e.g., at a breakpoint). (When EPC.SVSTOP = 1) 0: The count clock stops when the debugger obtains the control of the microcontroller (e.g., at a breakpoint). 1: The count clock is continuously supplied when the debugger obtains the control of the microcontroller (e.g., at a breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

24.4 General Operating Procedure

The following describes the general operating procedure for TAUBn.

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUBTTOUTm is also initialized and outputs a low level.

1. Set the TAUBnTPS register to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUBn function:
   - Set the operation mode
   - Set the channel output mode
   - Set any other control bits
3. Enable the counter by setting the TAUBnTS.TAUBnTSm bit to 1.
   The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.
4. If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUBnTT.TAUBnTTm bit to 1. The counter can be forcibly restarted by setting the TAUBnTS.TAUBnTSm bit to 1.
5. Stop the function by setting the TAUBnTT.TAUBnTTm bit to 1.

NOTE

- A detailed description of the required control bits and the operation of the individual functions is given below.
  - **Section 24.12, Independent Channel Operation Functions**
  - **Section 24.14, Synchronous Channel Operation Functions**
- The function can be changed while the counter is stopped (TAUBnTE.TAUBnTEm=0).

24.5 Concepts of Synchronous Channel Operation

The synchronous channel operation function is implemented using a combination of channel groups (consisted of master and slave channels).

Several rules apply to the settings of channels.

These rules are detailed in **Section 24.5.1, Rules of Synchronous Channel Operation Function**.

Two special features for synchronous channel operation are detailed in the following:

- **Section 24.5.2, Simultaneous Start and Stop of Synchronous Channel Counters**
- **Section 24.6, Simultaneous Rewrite**

24.5.1 Rules of Synchronous Channel Operation Function

Number of master and slave channels

- Only even channels (CH0, CH2, CH4, …) can be set as master channels.
  Any channel apart from CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and several slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 and the lower channels (CH3, CH4, CH5, …) can be set as slave channels.
- If multiple master channels are used, a range of slave channels that includes another master channel cannot be set for a given master channel.
  Example: If CH0 and CH4 are master channels, CH1 to CH3 can be set as slave channels for CH0, but CH5 to CH15 cannot.

Operation clock

- The same operation clock must be set for the master channel and its slave channel(s). This is achieved by setting the same value to the TAUBnCMORm.TAUBnCKS[1:0] bits for the slave and master channels.

The basic concepts of master/slave channel usage and operation clocks are illustrated in the following figure.

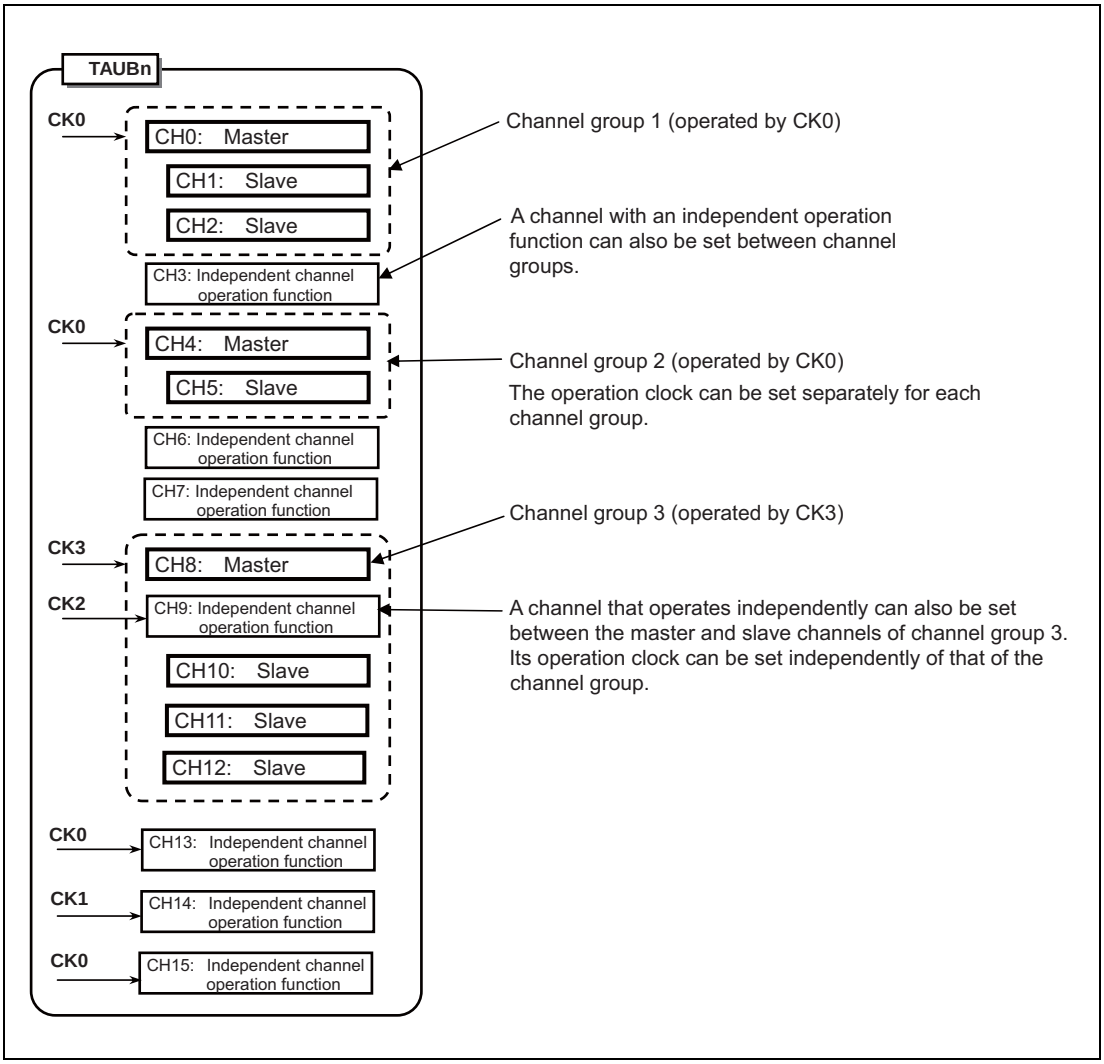


Figure 24.3 Grouping of the Channels and Assignment of Operation Clocks

24.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously within the same unit and between different units.

24.5.2.1 Simultaneous Start and Stop within the Same Unit

- To simultaneously start synchronized channels, the TAUBnTS.TAUBnTSm bits of the channels must be set at the same time.
- To simultaneously stop synchronized channels, the TAUBnTT.TAUBnTTm bits of the channels must be set at the same time.

Setting the TAUBnTS.TAUBnTSm bits to 1 sets the corresponding TAUBnTE.TAUBnTEm bits to 1, enabling counting. The exact time that it starts depends on the operation mode.

24.6 Simultaneous Rewrite

24.6.1 Introduction

Simultaneous rewrite refers to the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUBnCDRm and TAUBnTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered by:

- The counter on the master channel or upper channel (depending on the selected operation mode) reaching a certain value
- INTTAUBnIm being issued on the upper channel specified by TAUBnRDC.TAUBnRDCm

There are three methods for simultaneous rewrite. These are listed in the following table, along with how to specify them and when they cause simultaneous rewrite to be triggered.

Table 24.36 Simultaneous Rewrite Methods and when They are Triggered

Method	Trigger	TAUBnRDE.TAUBnRDEm	TAUBnRDS.TAUBnRDSm	TAUBnRDM.TAUBnRDMm
—	No simultaneous rewrite	0	0	0
A	The master channel (re)starts counting	1	0	0
B	The master channel has started counting, and starts counting down at the peak of triangular wave of the corresponding slave channel.	1	0	1
C1	INTTAUBnIm is generated on an upper channel specified by TAUBnRDC.TAUBnRDCm	1	1	0/1

The following table lists which of these three methods is available for each channel operation function. For more information about the individual channel operation functions, see the corresponding sections in **Section 24.13, Independent Channel Simultaneous Rewrite Functions** and **Section 24.14, Synchronous Channel Operation Functions**.

Table 24.37 Channel Functions and the Methods They Use for Simultaneous Rewrite

Functions	A	B	C1	TAUBnTOL.TAUBnTOLm
Simultaneous Rewrite Trigger Output Function Type 1			√	
PWM Output Function	√		√	√
One-Shot Pulse Output Function	√			
Delay Pulse Output Function	√			
Triangular Wave PWM Output Function		√	√	√
Triangular Wave PWM Output Function with Dead Time		√	√	
A/D Conversion Trigger Output Function Type 1	√		√	
A/D Conversion Trigger Output Function Type 2		√	√	

Note: √: Available, (Blank): Unavailable

24.6.2 How to Control Simultaneous Rewrite

The following figure shows the general procedure for simultaneous rewrite.
The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

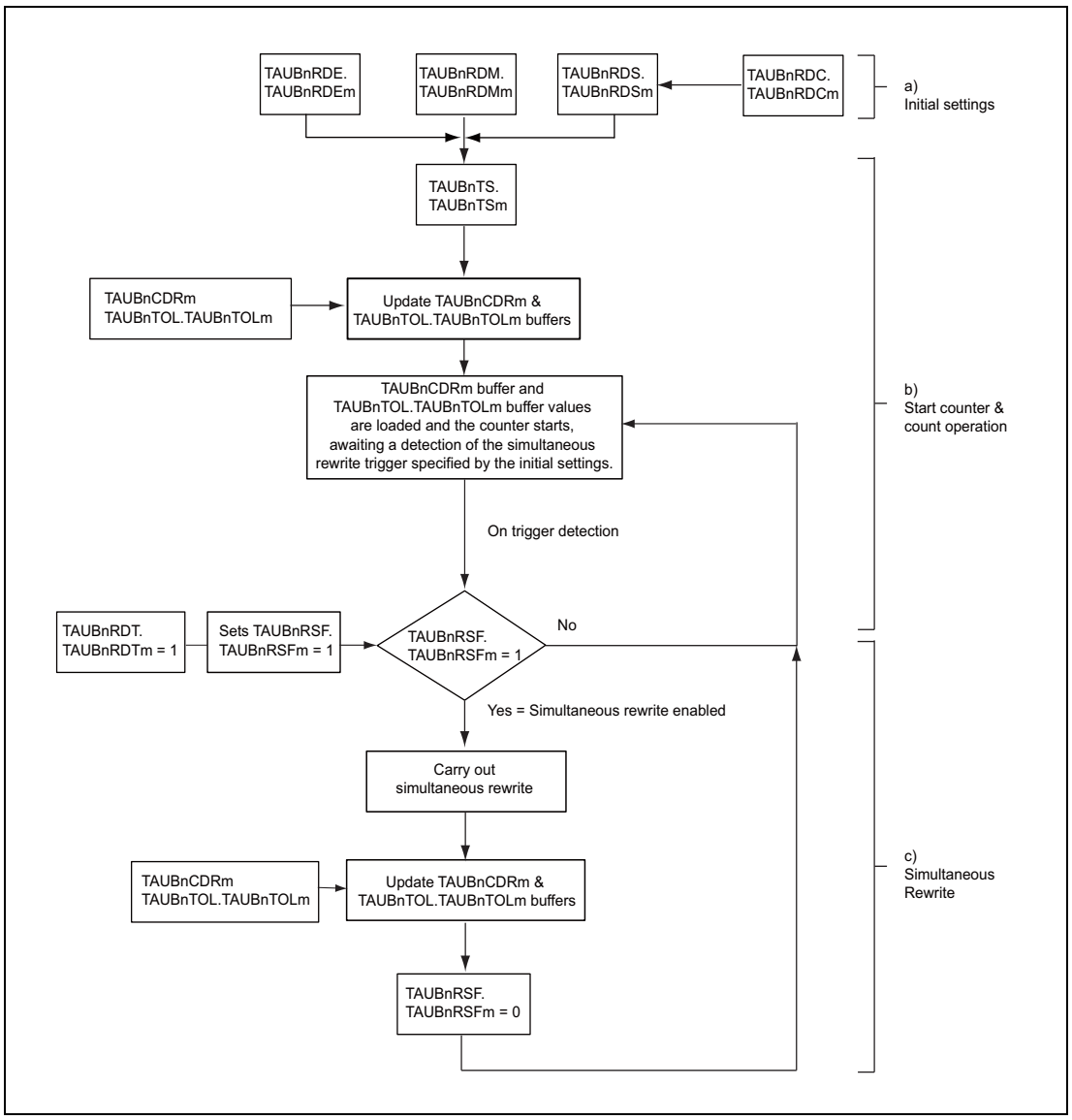


Figure 24.4 General Procedure for Simultaneous Rewrite

24.6.2.1 Initial Settings

- To enable simultaneous rewrite in channel m, set TAUBnRDE.TAUBnRDEm = 1.
- To select the type of simultaneous rewrite, set TAUBnRDM.TAUBnRDMm and TAUBnRDS.TAUBnRDSm according to the values in **Table 24.36, Simultaneous Rewrite Methods and when They are Triggered**.
- Specify a simultaneous rewrite trigger channel by using TAUBnRDC.TAUBnRDCm. (Prerequisite: TAUBnRDS.TAUBnRDSm has been set to the upper channel.)

24.6.2.2 Start Counter and Count Operation

- To start all the TAUBnCNTm counters in the channel group, set the corresponding TAUBnTS.TAUBnTSm bits to 1. TAUBnTOL.TAUBnTOLm and the values in the data registers (TAUBnCDRm) are loaded to the corresponding TAUBnTOL.TAUBnTOLm buffer (TAUBnTOL.TAUBnTOLm buf) and data buffer registers (TAUBnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUBnRDT.TAUBnRDTm) to 1 sets the reload flag (TAUBnRSF.TAUBnRSFm) to 1, enabling simultaneous rewrite. TAUBnRSF.TAUBnRSFm remains at 1 until simultaneous rewrite has taken place.
- When the specified trigger for simultaneous rewrite is detected, the TAUBnRSF.TAUBnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUBnRSF.TAUBnRSFm = 1). If it is, simultaneous rewrite is carried out.
  Otherwise, simultaneous rewrite is not carried out, and the system awaits the next simultaneous rewrite trigger detection.

24.6.2.3 Simultaneous Rewrite

- When simultaneous rewrite is enabled (TAUBnRSF.TAUBnRSFm = 1) and the simultaneous rewrite trigger is detected, the current values of the data registers are copied to their buffers. These values are then loaded to the corresponding counters and the values are applied the next time the counter starts or restarts.
- When simultaneous rewrite is finished, the TAUBnRSF.TAUBnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

24.6.3 Other General Rules of Simultaneous Rewrite

The following rules also apply:

- TAUBnRDE.TAUBnRDEm, TAUBnRDS.TAUBnRDSm, TAUBnRDM.TAUBnRDMm, and TAUBnRDC.TAUBnRDCm cannot be changed while the counter is in operation (TAUBnTE.TAUBnTEm = 1).
- TAUBnTOL.TAUBnTOLm can only be rewritten during operation when in PWM output function or triangular wave PWM output function mode. For all other functions, TAUBnTOL.TAUBnTOLm must be written before the counter starts. If it is rewritten in another function mode, TAUBTTOUTm outputs an invalid wave.
- When a simultaneous rewrite trigger is issued on an upper channel (TAUBnRDS.TAUBnRDSm = 1), the TAUBnRDC.TAUBnRDCm bit controls all the lower channels. This means that if the TAUBnRDC.TAUBnRDCm bits of CH2 and CH7 are set to 1 and the TAUBnRDC.TAUBnRDCm bits of other channels are set to 0, CH2 and CH7 serve as simultaneous rewrite trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.
- If simultaneous rewrite is enabled and an upper channel is selected for the simultaneous rewrite trigger generation channel (TAUBnRDE.TAUBnRDEm and TAUBnRDS.TAUBnRDSm = 1) but no upper channel is set (TAUBnRDC.TAUBnRDC[15:0] = 0), simultaneous rewrite cannot take place.

24.6.4 Types of Simultaneous Rewrite

In the following section, the three simultaneous rewrite methods are explained using timing diagrams.

24.6.4.1 Simultaneous Rewrite when the Master Channel (Re)Starts Counting (Method A)

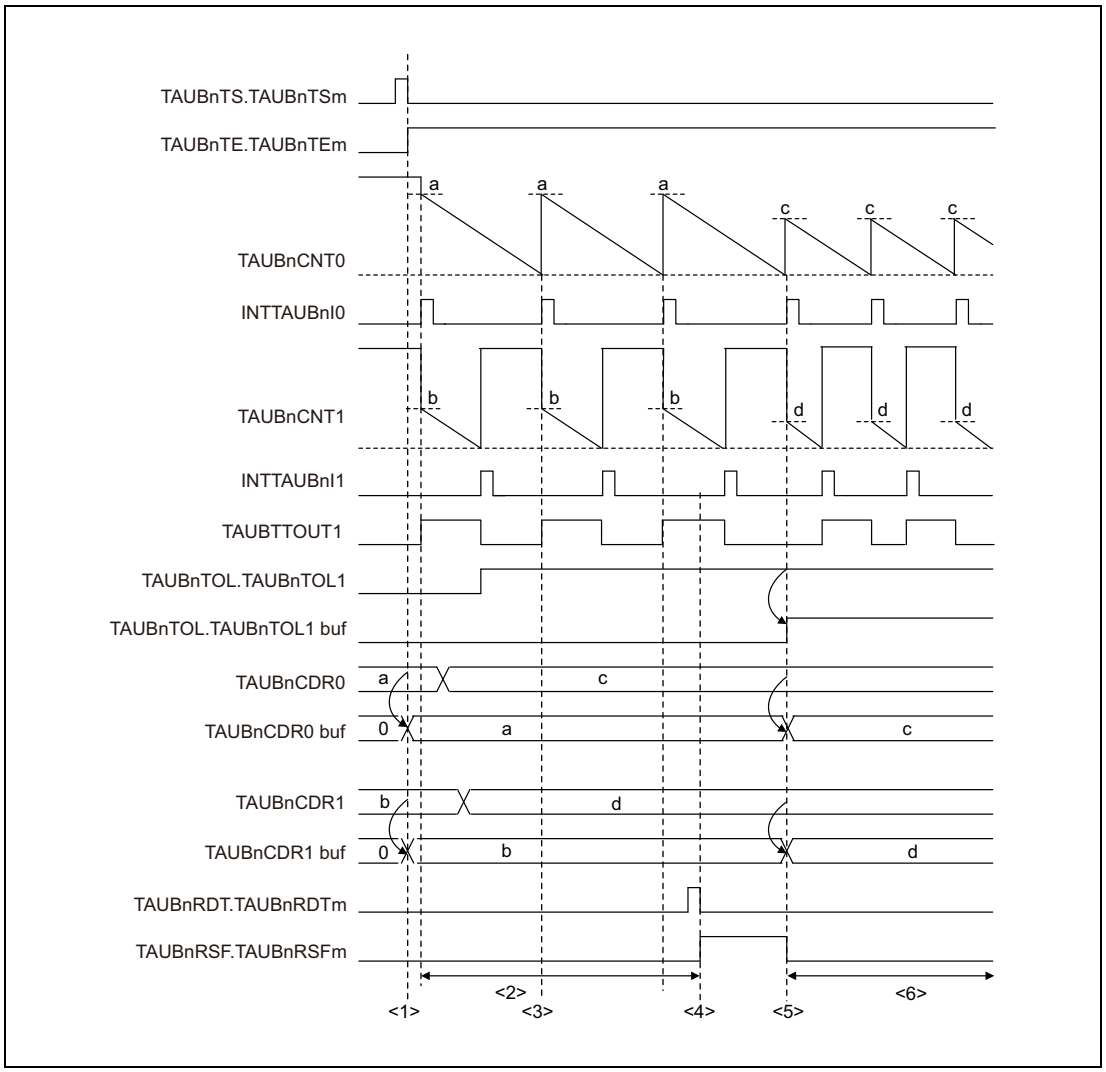


Figure 24.5 Simultaneous Rewrite when the Master Channel (Re)Starts Counting

Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method A is applied.

Description:

(1) When TAUBnTS.TAUBnTSm = 1 is set, the value of TAUBnCDRm is copied to the TAUBnCDRm buffer and the value of TAUBnTOL.TAUBnTOLm is copied to the TAUBnTOL.TAUBnTOLm buffer.

(2) The TAUBnCDRm and TAUBnTOL.TAUBnTOLm registers can be written at any time.

(3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUBnRSF.TAUBnRSFm = 0)

(4) The reload data trigger bit (TAUBnRDT.TAUBnRDTm) is set to 1 which sets the status flag (TAUBnRSF.TAUBnRSFm = 1), enabling simultaneous rewrite.

(5) Because simultaneous rewrite is enabled, it is triggered when CH0 restarts counting. The TAUBnCDRm value is loaded into the TAUBnCDRm buffer, and the TAUBnTOL.TAUBnTOLm value is loaded into the TAUBnTOL.TAUBnTOLm buffer.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUBnCDRm and TAUBnTOL.TAUBnTOLm can be changed again.

24.6.4.2 Simultaneous Rewrite at the Peak of a Triangular Wave of the Slave Channel (Method B)

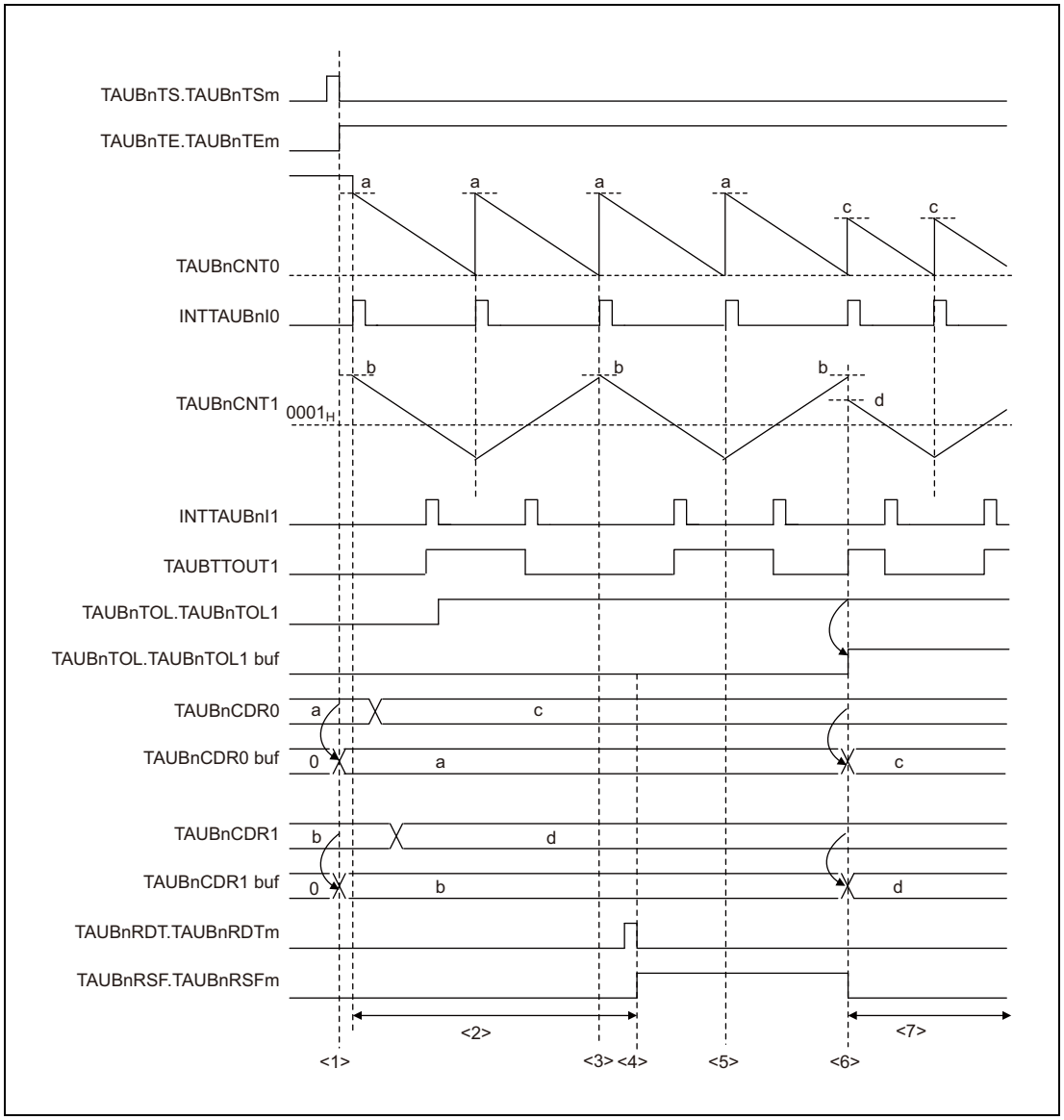


Figure 24.6 Simultaneous Rewrite at the Peak of a Triangular Wave of the Slave Channel

Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method B is applied.

Description:

(1) When TAUBnTS.TAUBnTSm = 1 is set, the value of TAUBnCDRm is copied to the TAUBnCDRm buffer.

(2) The TAUBnCDRm and TAUBnTOL registers can be written at any time.

(3) Simultaneous rewrite does not occur because it is disabled (TAUBnRSF.TAUBnRSFm = 0).

(4) The reload data trigger bit (TAUBnRDT.TAUBnRDTm) is set to 1 which sets the status flag (TAUBnRSF.TAUBnRSFm = 1), enabling simultaneous rewrite.

(5) Simultaneous rewrite does not take place at the bottom of the triangular cycle.

(6) Simultaneous rewrite takes place at the start timing of the top of the triangular cycle. The TAUBnCDRm value is loaded into the TAUBnCDRm buffer, and the TAUBnTOL.TAUBnTOLm value is loaded into the TAUBnTOL.TAUBnTOLm buffer.

(7) The counters count down and await the next simultaneous rewrite trigger. The values of TAUBnCDRm and TAUBnTOL.TAUBnTOLm can be changed again.

24.6.4.3 Simultaneous Rewrite when INTTAUBnIm is Generated on an Upper Channel Specified by TAUBnRDC.TAUBnRDCm (Method C1)

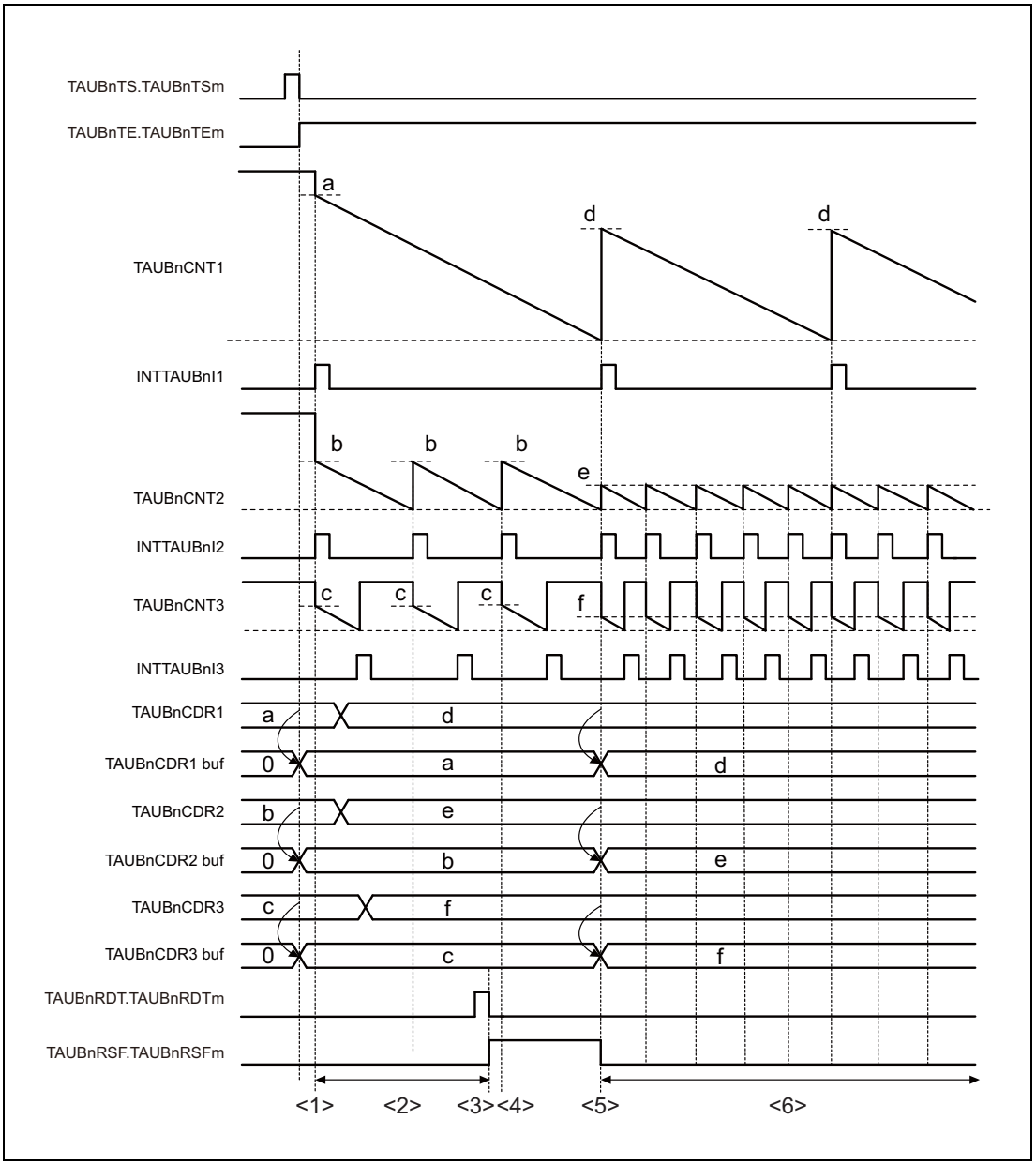


Figure 24.7 Simultaneous Rewrite when INTTAUBnIm is Generated on an Upper Channel Specified by TAUBnRDC.TAUBnRDCm

Setting:

CH1 is the upper channel used for counting down, CH2 is a master channel, and CH3 is the slave channel. The simultaneous rewrite method C1 is applied. The TAUBnRDC register specifies the simultaneous rewrite trigger channel.

Description:

(1) When TAUBnTS.TAUBnTSm is set to 1, the TAUBnCDRm value is copied to the TAUBnCDRm buffer.

(2) The TAUBnCDRm register is always ready to write.

(3) By setting the reload data trigger bit (TAUBnRDT.TAUBnRDTm) to 1, the status flag is set (TAUBnRSF.TAUBnRSFm = 1) to enable simultaneous rewrite.

(4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.

(5) Simultaneous rewrite is triggered by INT1 which is generated when counter 1 reaches 0000_H. The TAUBnCDRm values are loaded into the corresponding TAUBnCDRm buffers.

(6) The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUBnCDRm registers can be rechanged.

24.7 Channel Output Modes

The output of the TAUBTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUBnTOE.TAUBnTOEm = 0)
  When controlled by software, the value written in the output register bit (TAUBnTO.TAUBnTOm) is sent to the output pin (TAUBTTOUTm).
- By TAUB signals (TAUBnTOE.TAUBnTOEm = 1)
  When controlled by TAUB signals, the output level of TAUBTTOUTm is set or reset or toggled by internal signals. The value of TAUBnTO.TAUBnTOm is updated accordingly to reflect the value of TAUBTTOUTm.
  - Independently (TAUBnTOM.TAUBnTOMm = 0)
    In case of independent operation, the output of the TAUBTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUBnTOM.TAUBnTOMm = 0).
  - Synchronously (TAUBnTOM.TAUBnTOMm = 1)
    In case of synchronous operation, the output of the TAUBTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUBnTOM.TAUBnTOMm = 1).

The TAUBnTO.TAUBnTOm bit can always be read to determine the current value of TAUBTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 24.38, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 24.7.2, Channel Output Modes Controlled Independently by TAUBn Signals**
- **Section 24.7.3, Channel Output Modes Controlled Synchronously by TAUBn Signals**

Batch operation of TAUBnTOm bit

Whether a set value is reflected to the TAUBnTOm bit or not is controlled by the TAUBnTOE.TAUBnTOEm bit.

The TAUBnTOm setting is written only to the bit (channel) set with TAUBnTOE.TAUBnTOEm bit = 0 when a write to the TAUBnTO register is attempted. No TAUBnTOm setting is reflected to the bit (channel) set with TAUBnTOE.TAUBnTOEm bit = 1.

NOTE

The TAUBnTO.TAUBnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or negative logic of the output is specified by control bit TAUBnTOL.TAUBnTOLm.

The value of TAUBnTOL.TAUBnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function or triangular wave PWM output function. If TAUBnTOL.TAUBnTOLm is changed after the counter starts, the output of TAUBTTOUTm is undefined.

See **Section 24.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 24.38**.

Table 24.38 Channel Output Modes

Channel Output Mode	TAUBnTOE. TAUBnTOEm	TAUBnTOM. TAUBnTOMm	TAUBnTOC. TAUBnTOCm	TAUBnTDE. TAUBnTDEm
By software				
Independent channel output mode controlled by software	0	x		
By TAUB signals, independently				
Independent channel output mode 1	1	0	0	0
Independent channel output mode 2			1	
By TAUB signals, synchronously				
Synchronous channel output mode 1	1	1	0	0
Synchronous channel output mode 2			1	0
Synchronous channel output mode 2 with dead time output				1

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

NOTE

The following bits cannot be changed during count operation (TAUBnTE.TAUBnTEm = 1):

- TAUBnTOM.TAUBnTOMm
- TAUBnTOC.TAUBnTOCm
- TAUBnTDE.TAUBnTDEm

24.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUBTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUBnTOE.TAUBnTOEm = 0).

(1) Set TAUBnTO.TAUBnTOm to specify the initial level of the TAUBTTOUTm output.

(2) Set channel output mode according to **Table 24.38, Channel Output Modes**, and the output logic using the TAUBnTOL.TAUBnTOLm bit.

(3) Start the counter (TAUBnTS.TAUBnTSm = 1).

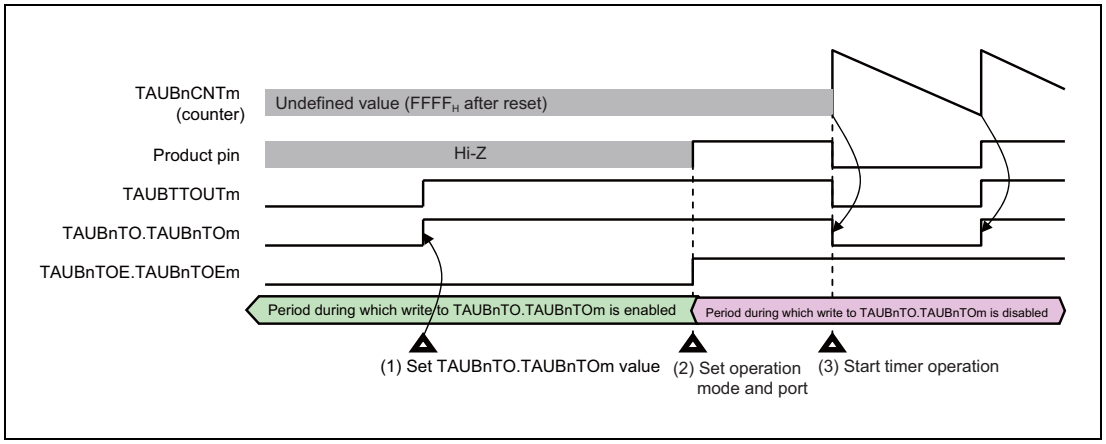


Figure 24.8 General Procedure for Specifying a TAUBTTOUTm Channel Output Mode

24.7.2 Channel Output Modes Controlled Independently by TAUBn Signals

This section lists the channel output modes that are controlled independently by TAUBn signals. The control bits used to specify a mode are listed in **Table 24.38, Channel Output Modes**.

24.7.2.1 Independent Channel Output Mode 1

Set/reset conditions

In this output mode, TAUBTTOUTm toggles when INTTAUBnIm is detected. The value of TAUBnTOL.TAUBnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 24.38, Channel Output Modes**.

24.7.2.2 Independent Channel Output Mode 2

Set/reset conditions

In this output mode, TAUBTTOUTm is set when INTTAUBnIm occurs at the time of count start, and reset when INTTAUBnIm occurs due to a match between TAUBnCNTm and TAUBnCDRm.

Prerequisites

There are no prerequisites other than those shown in **Table 24.38, Channel Output Modes**.

24.7.3 Channel Output Modes Controlled Synchronously by TAUBn Signals

This section lists the channel output modes that are controlled synchronously by TAUBn signals. The control bits used to specify a mode are listed in **Table 24.38, Channel Output Modes**.

24.7.3.1 Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUBnIm of the master channel serves as a set signal and INTTAUBnIm of the slave channel as a reset signal. If INTTAUBnIm of the master channel and INTTAUBnIm of the slave channel are generated at the same time, INTTAUBnIm of the slave channel (reset signal) has priority over INTTAUBnIm (set signal) of the master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 24.38, Channel Output Modes**.

24.7.3.2 Synchronous Channel Output Mode 2

In this output mode, the operating mode should be set to count-up/-down mode. The result is trianglular wave PWM output at TAUBTTOUTm. For details, see **Section 24.14.5, Triangular Wave PWM Output Function**.

Set/reset conditions

TAUBnCNTm of the slave channel counts down and up alternatively. When it passes 0001_H it generates an interrupt, causing TAUBTTOUTm to toggle.

Prerequisites

A set of two channels is required to generate the triangular wave PWM output. TAUBTTOUTm should be set to 0 before the function starts.

24.7.3.3 Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUBTTOUTm. The set/reset conditions are shown in **Figure 24.9**.

Set/reset conditions

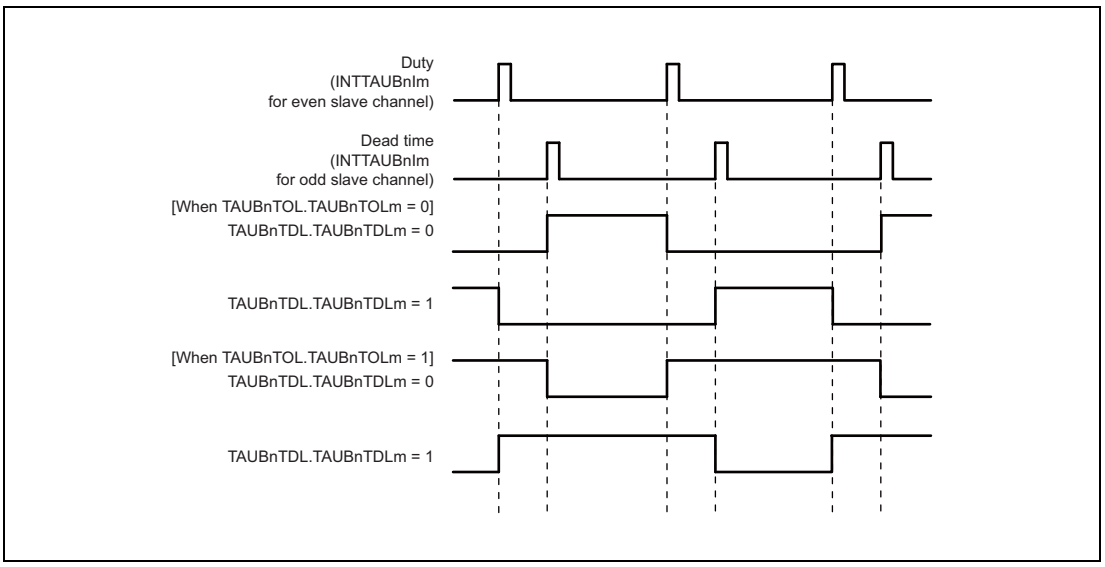


Figure 24.9 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output

With regard to the edge to which dead time is added, set TAUBnTDL.TAUBnTDLm = 0 for rising edges and TAUBnTDL.TAUBnTDLm = 1 for falling edges.

Prerequisites

Dead time control requires a set of three channels, each operating in the following modes:

- One master channel
  The master channel should be set to interval timer mode.
- One even slave channel
  The even slave channel should be set to count-up/-down mode.
- One odd slave channel (even channel + 1)
  The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUBnTOE.TAUBnTOEm
- TAUBnTOM.TAUBnTOMm
- TAUBnTOC.TAUBnTOCm
- TAUBnTDE.TAUBnTDEm

24.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUBnTS.TAUBnTSm is set to 1 in each operating mode.

In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

24.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-Up/-Down Mode, and Count Capture Mode

The counter starts operating at the next count clock after TAUBnTS.TAUBnTSm is set to 1. The value of data register is also loaded when the counter starts.

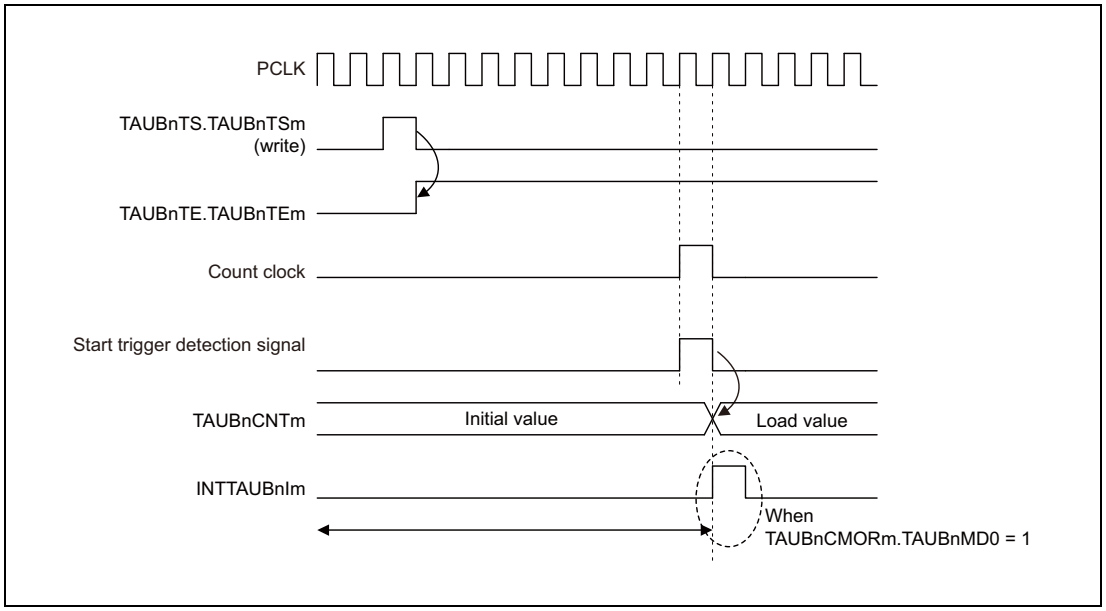


Figure 24.10 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, Up/Down Count Mode, and Count Capture Mode

NOTE

Make sure to set TAUBnCMORm.TAUBnMD0 to 0 when using the count-up/-down mode.

24.8.2 Event Count Mode

The value of data register is loaded as soon as TAUBnTS.TAUBnTSm is set to 1. The counter also starts immediately. The value of data register decrements when the subsequent count clock cycle starts.

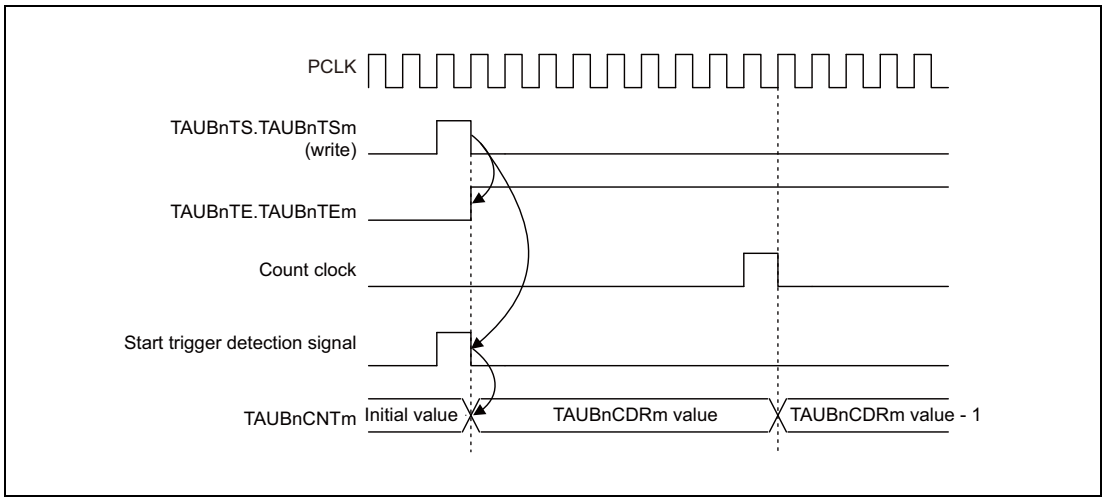


Figure 24.11 Start Timing in Event Count Mode

24.8.3 Other Operating Modes

In other operating modes, the counter operation start timing is triggered only upon detection of a valid edge of TAUBTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which are irrelevant to start of counter operation, determine the frequency with which all operations take place.

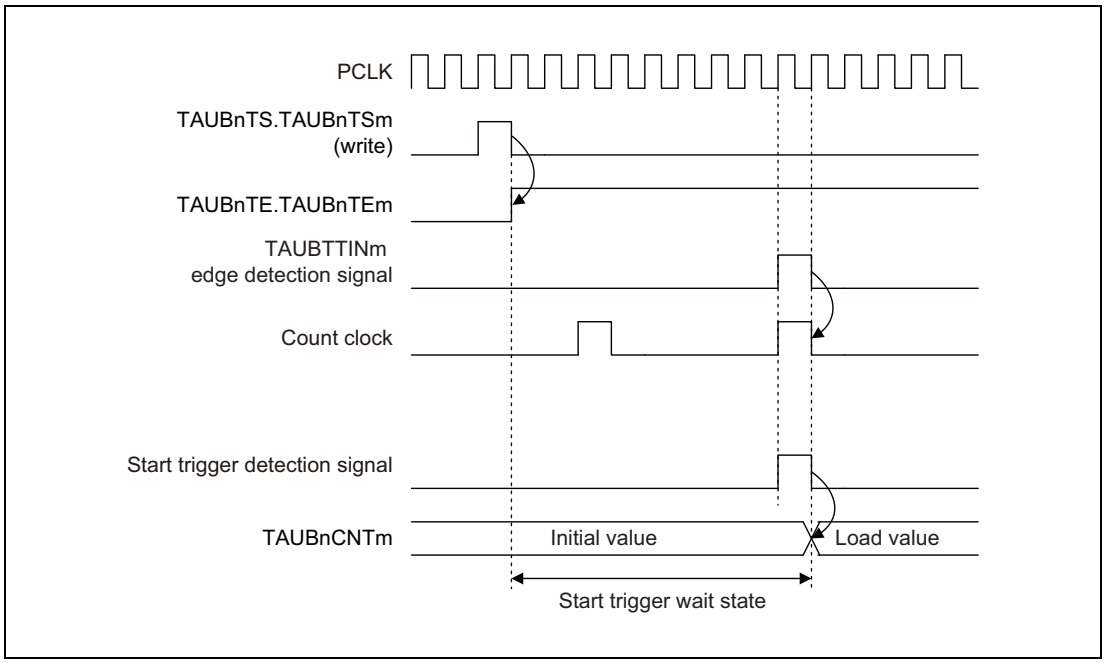


Figure 24.12 Count Start Timing in Other Operating Modes

24.9 TAUBTTOUTm Output and INTTAUBnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUBnIm is generated using the TAUBnCMORm.TAUBnMD0 bit. The generation of INTTAUBnIm when the TAUBnCMORm.TAUBnMD0 bit starts counting and the effect to TAUBTTOUTm depend on the selected function. For details, refer to the description of TAUBnCMORm.TAUBnMD0 of each function.

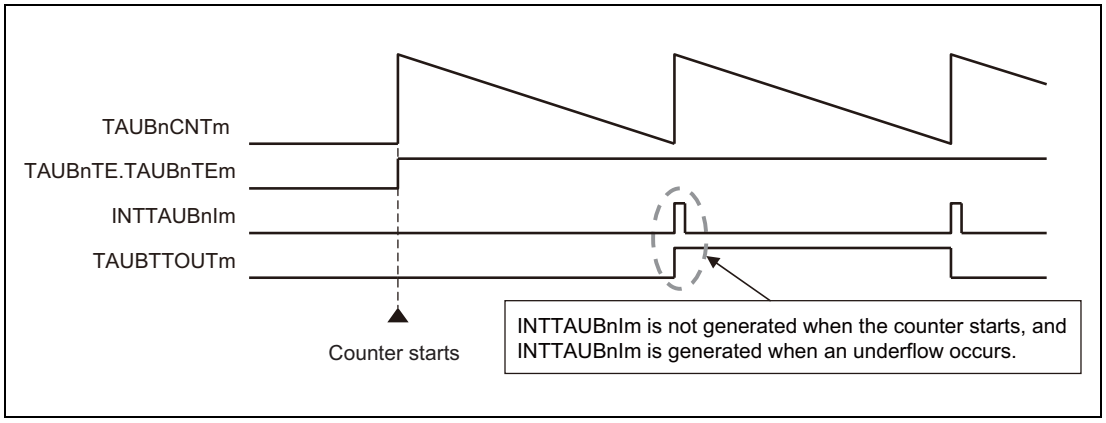


Figure 24.13 INTTAUBnIm Generation Timing (TAUBnCMORm.TAUBnMD0=0)

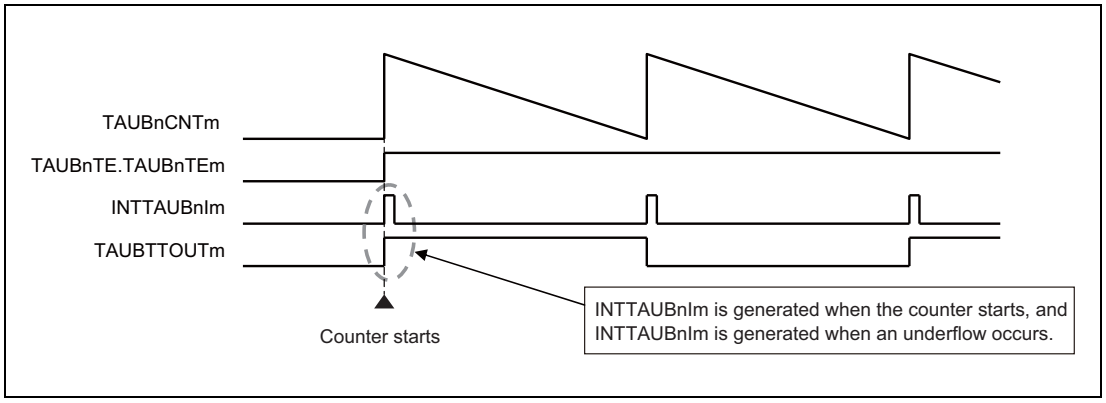


Figure 24.14 INTTAUBnIm Generation Timing (TAUBnCMORm.TAUBnMD0=1)

24.10 Interrupt Generation upon Overflow

For certain independent functions, an interrupt is not generated when the counter value reaches $FFFF_H$ and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channels that are operating in count-up and count-down modes.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000_H at the same time as the first channel overflows (TAUBnCNTm = $FFFF_H$).
- Set TAUBnCDRm of the second channel to $FFFF_H$.
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same TAUBTTINm input.
- The trigger detection settings (TAUBnCMORm.TAUBnSTS[2:0] and TAUBnCMURm.TAUBnTIS[1:0]) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000_H at exactly the same time as the up-counter of the first channel overflows (TAUBnCNTm = $FFFF_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

24.10.1 Example of Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of the TAUBTTINm input interval timer function can detect the overflow when TAUBnCNTm of the TAUBTTINm input pulse interval measurement function exceeds $FFFF_H$.

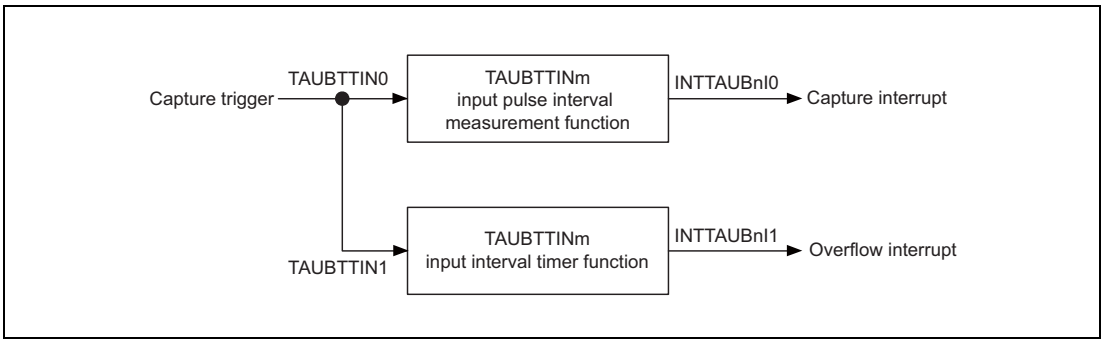


Figure 24.15 Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

Timing diagram

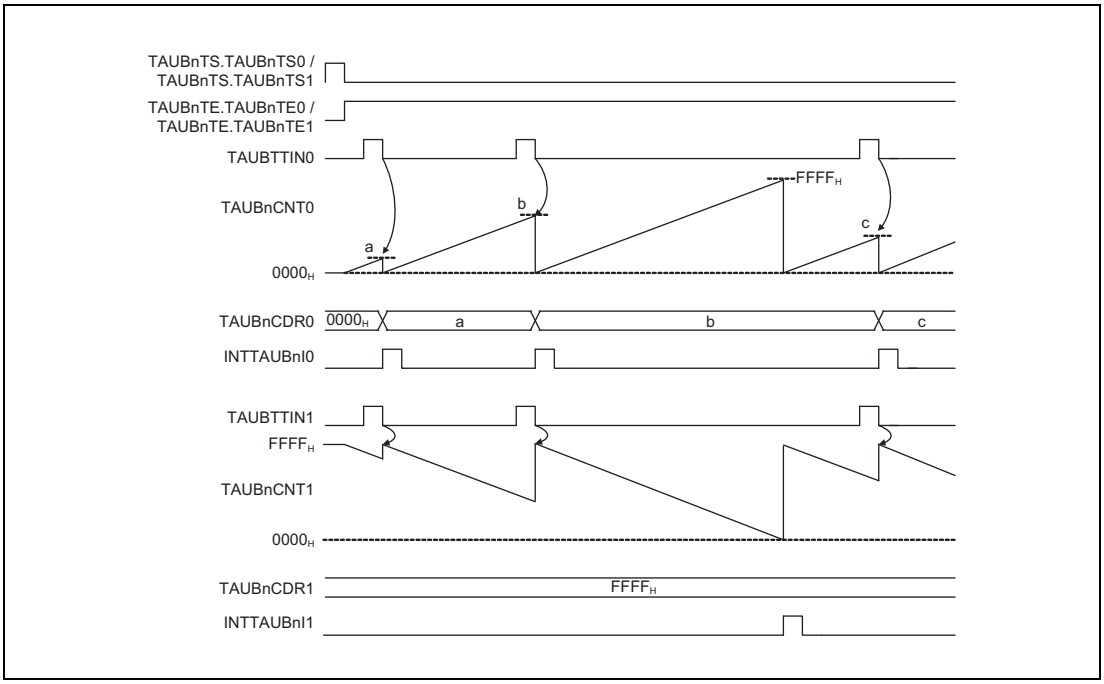


Figure 24.16 Interrupt Generation by Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

24.10.2 Example of Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of the overflow interrupt output function (during TAUBTTINm width measurement) can detect the overflow when TAUBnCNTm of the TAUBTTINm input signal width measurement function exceeds $FFFF_H$.

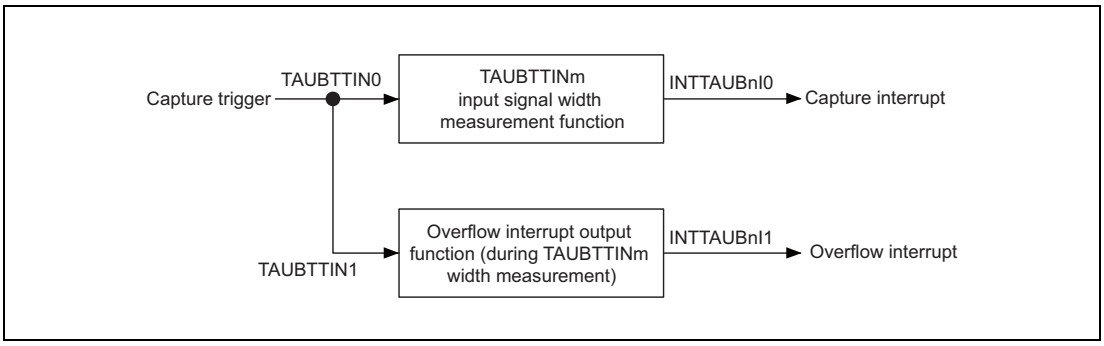


Figure 24.17 Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Timing diagram

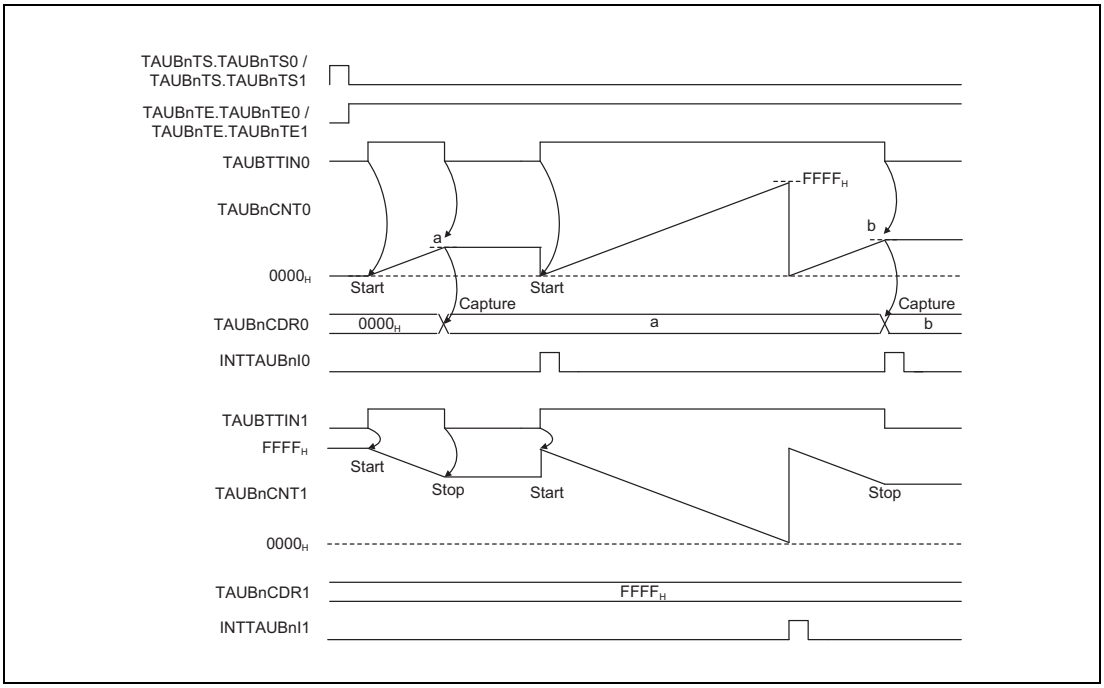


Figure 24.18 Interrupt Generation by Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

24.10.3 Example of Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

When the counters of both channels are enabled simultaneously, INTTAUBnIm of the interval timer function can detect the overflow when TAUBnCNTm of the TAUBTTINm input position detection function exceeds $FFFF_H$.

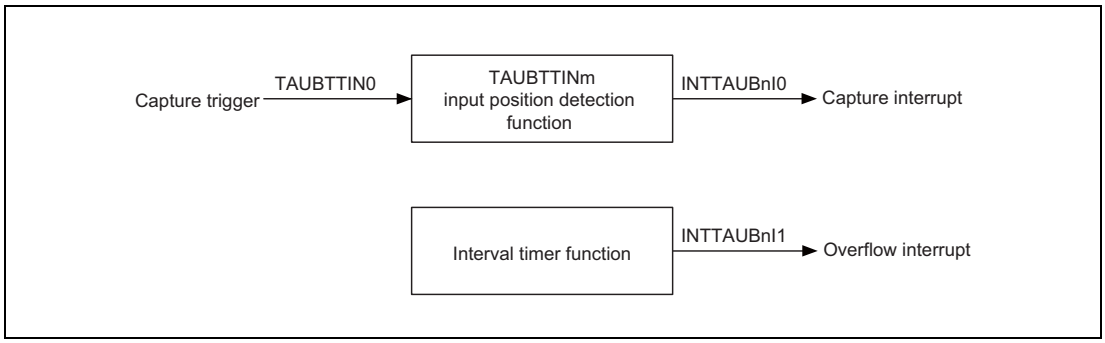


Figure 24.19 Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

Timing diagram

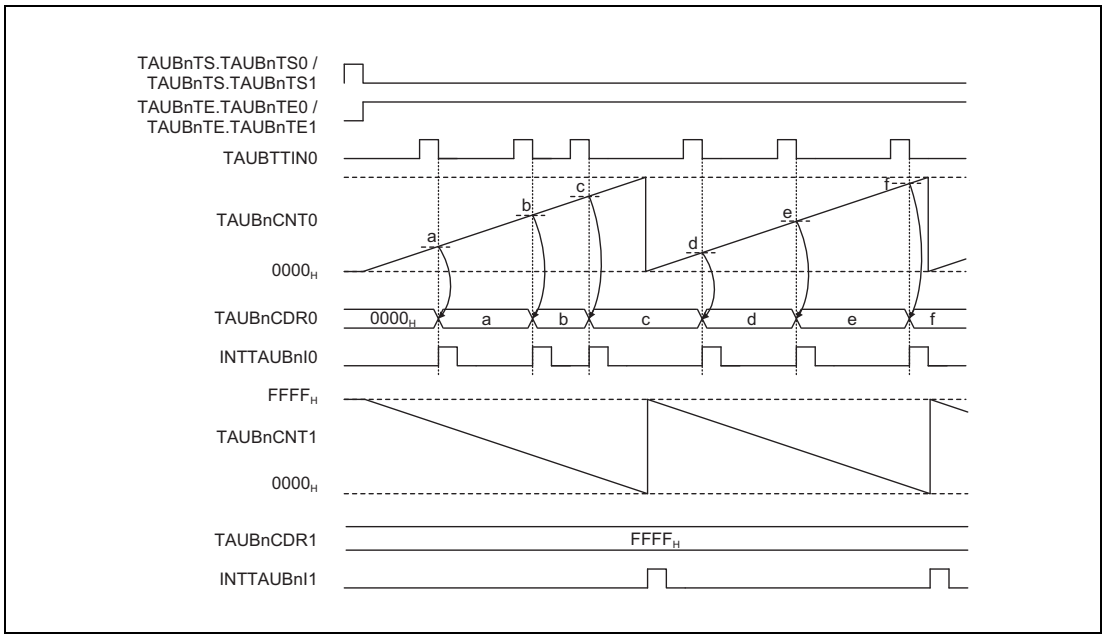


Figure 24.20 Interrupt Generation by Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

24.10.4 Example of Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of the overflow interrupt output function (during TAUBTTINm input period count detection) can detect the overflow when TAUBnCNTm of the TAUBTTINm input period count detection function exceeds $FFFF_H$.

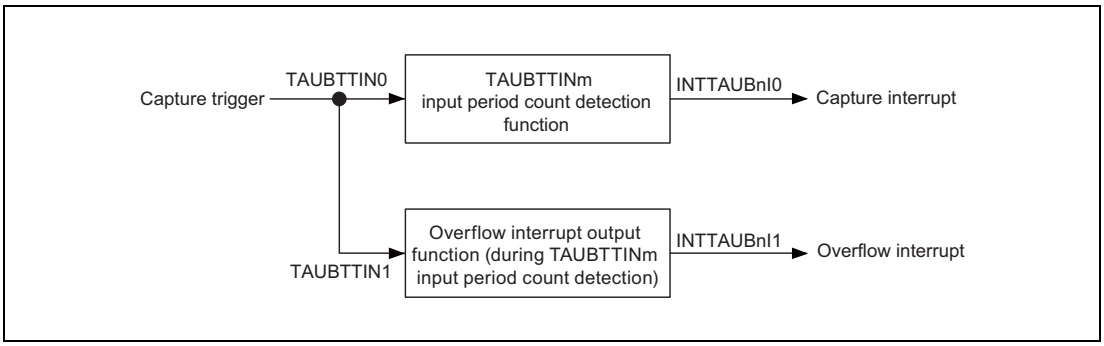


Figure 24.21 Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (TAUBTTINm Input Period Count Detection)

Timing diagram

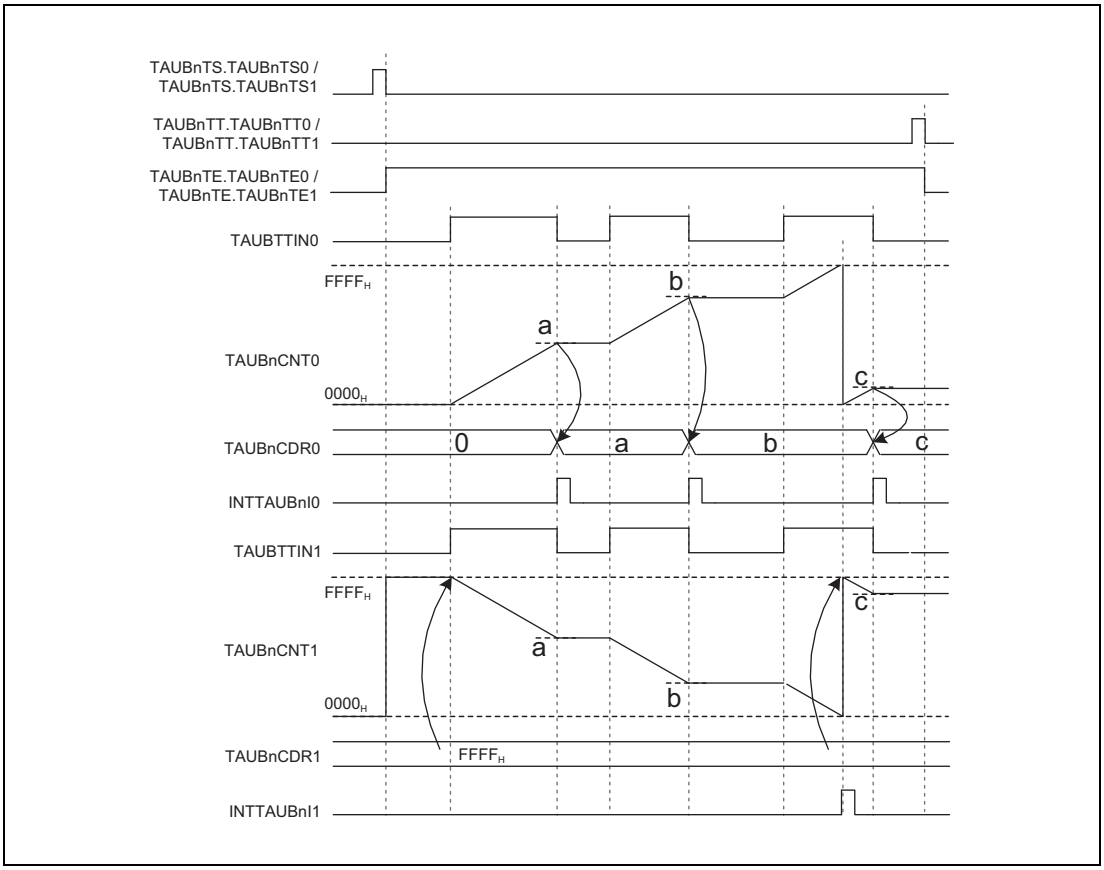


Figure 24.22 Interrupt Generation by Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

24.11 TAUBTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

The following figure shows when edge detection takes place.

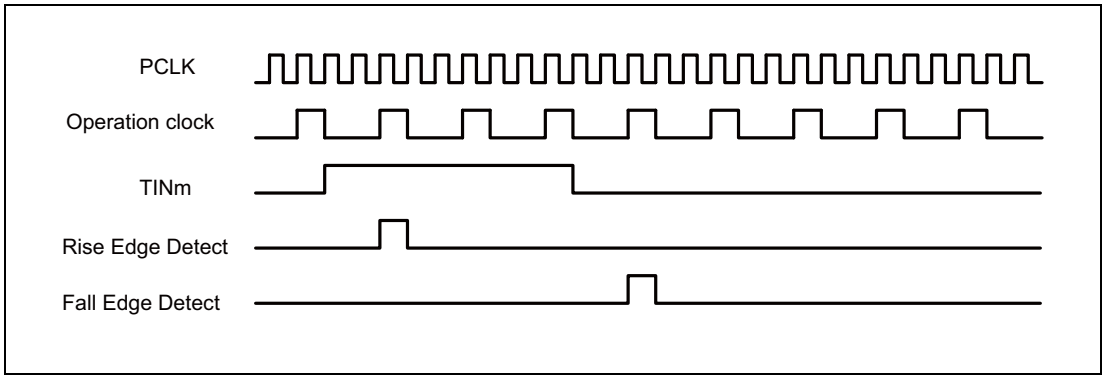


Figure 24.23 Basic Edge Detection Timing

Figure 24.23 is an image of the operation timing. Actually, the delay time caused by the noise filter and the synchronization circuit between the TAUBnIm terminal and TAUBn will be generated.

24.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the TAUB. For a general overview of independent channel operation functions, see **Section 24.2, Overview**.

24.12.1 Interval Timer Function

24.12.1.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUBnIm) at regular intervals. When an interrupt is generated, the TAUBTTOUTm signal toggles, resulting in a square wave.

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is loaded to TAUBnCNTm and the counter starts to count down from this value.

When the counter reaches 0000_H, INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm stop but retain their values. The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.

Conditions

If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUBTTOUTm does not toggle. This results in an inverted TAUBTTOUTm signal output when TAUBnCMORm.TAUBnMD0 is set to 1.

24.12.1.2 Equations

INTTAUBnIm cycle = count clock cycle × (TAUBnCDRm + 1)

TAUBTTOUTm square wave cycle = count clock cycle × (TAUBnCDRm + 1) × 2

24.12.1.3 Block Diagram and General Timing Diagram

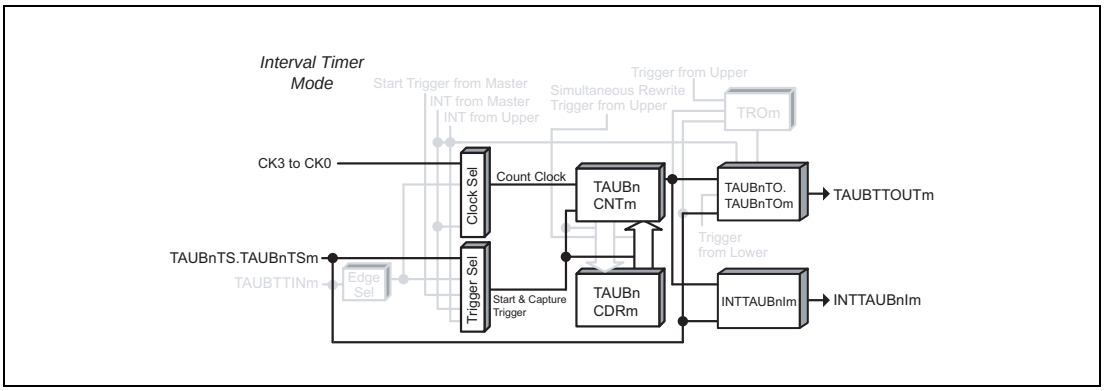


Figure 24.24 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

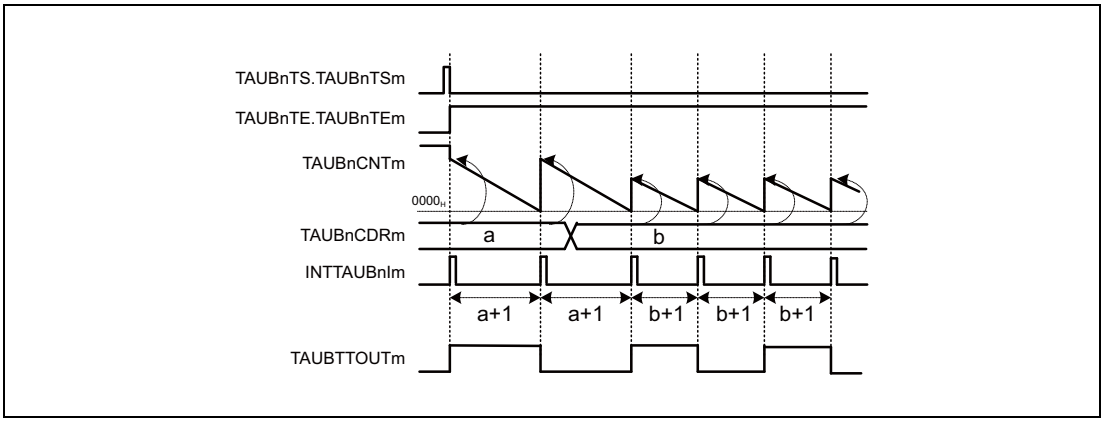


Figure 24.25 General Timing Diagram for Interval Timer Function

24.12.1.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.39 Contents of the TAUBnCMORm Register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm is not generated and TAUBTTOUTm does not toggle at operation start. 1: INTTAUBnIm is generated and TAUBTTOUTm toggles at operation start or restart.

(2) TAUBnCMURm

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

Table 24.40 Contents of the TAUBnCMURm Register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode

Table 24.41 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	Write 0_B.
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUBnTOE.TAUBnTOEm = 0. TAUBTTOUTm can then be controlled independently of the interrupts.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Interval Timer Function. Therefore, these registers must be set to 0.

Table 24.42 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.1.5 Operating Procedure for Interval Timer Function

Table 24.43 Operating Procedure for Interval Timer Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 24.39, Contents of the TAUBnCMORm Register for Interval Timer Function** and **Table 24.40, Contents of the TAUBnCMURm Register for Interval Timer Function** Set the value of the TAUBnCDRm register Set the channel output mode by setting the control bits as described in **Table 24.41, Control Bit Settings for Independent Channel Output Mode 1**	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated and TAUBTTOUTm toggles.
During operation	The TAUBnCDRm register value can be changed at any time. The TAUBnCNTm register can be read at all times.	TAUBnCNTm counts down. When the counter reaches 0000_H: • TAUBnCNTm reloads the TAUBnCDRm value and continues count operation • INTTAUBnIm is generated and TAUBTTOUTm toggles.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

24.12.1.6 Specific Timing Diagrams

(1) TAUBnCDRm = 0000_H, count clock = PCLK/2

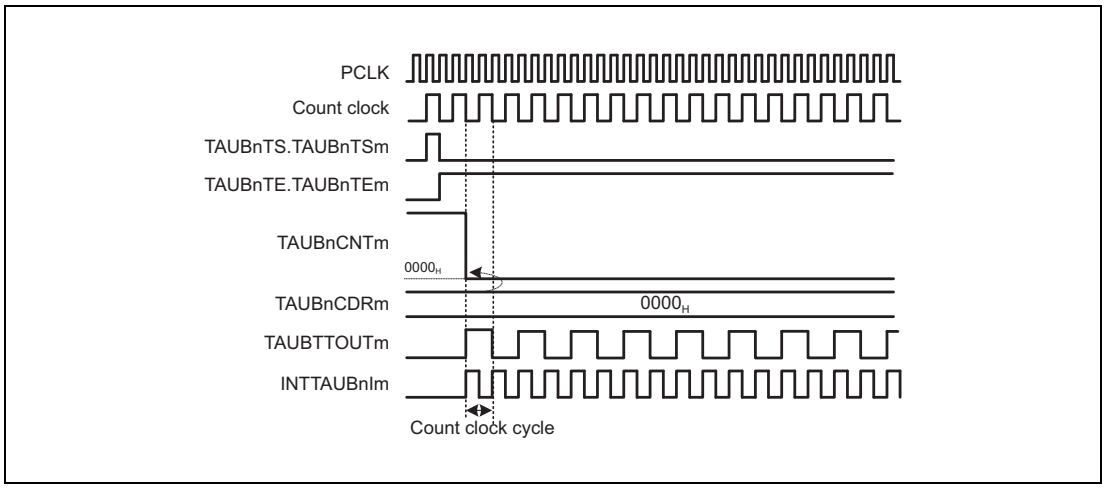


Figure 24.26 TAUBnCDRm = 0000_H, Count Clock = PCLK/2

- TAUBnCDRm = 0000_H, and the count clock = PCLK/2, the TAUBnCDRm value is written to TAUBnCNTm every count clock, meaning that TAUBnCNTm is always 0000_H.
- INTTAUBnIm is generated every count clock, resulting in TAUBTTOUTm toggling every count clock.

(2) TAUBnCDRm = 0000_H, count clock = PCLK

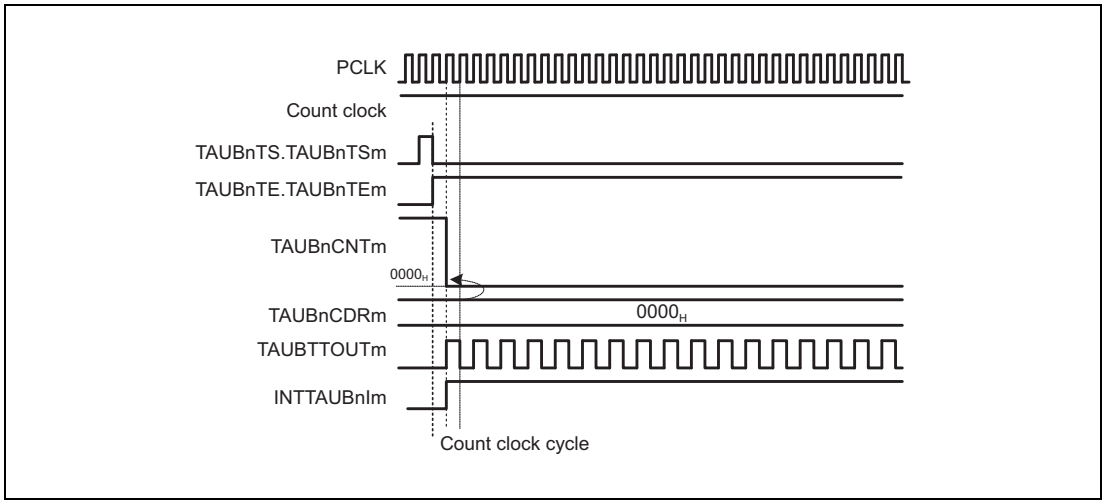


Figure 24.27 TAUBnCDRm = 0000_H, Count Clock = PCLK

- TAUBnCDRm = 0000_H, and the count clock = PCLK, the TAUBnCDRm value is written to TAUBnCNTm every PCLK clock, meaning that TAUBnCNTm is always 0000_H.
- INTTAUBnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated.
  TAUBTTOUTm is toggled every PCLK clock.

(3) Operation stop and restart (TAUBnCMORm TAUBnMDO = 1)

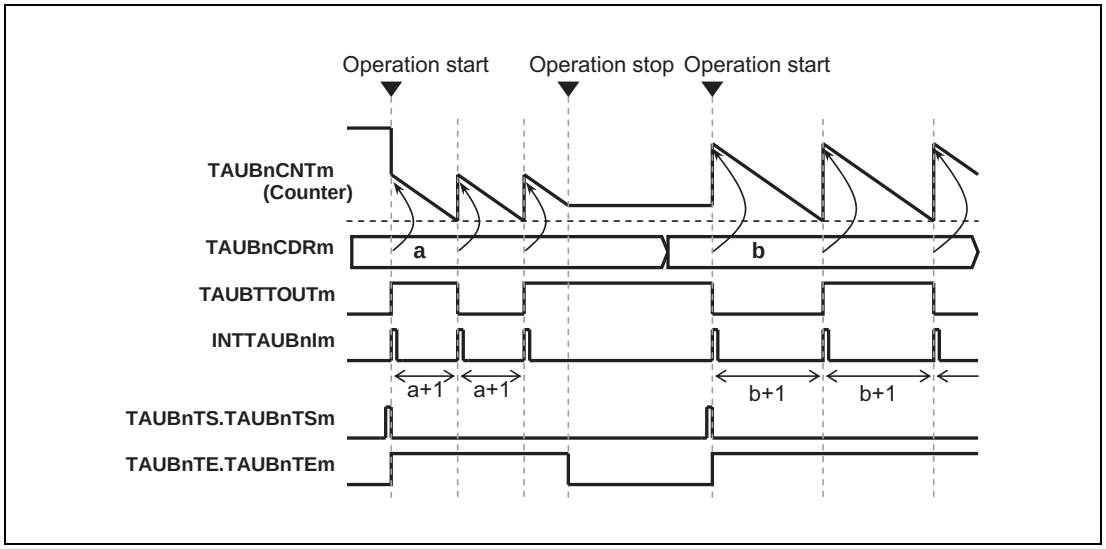


Figure 24.28 Operation Stop and Restart, TAUBnCMORm.TAUBnMD0 = 1

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm and TAUBTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1.

(4) Operation stop and restart (TAUBnCMORm.TAUBnMD0 = 0)

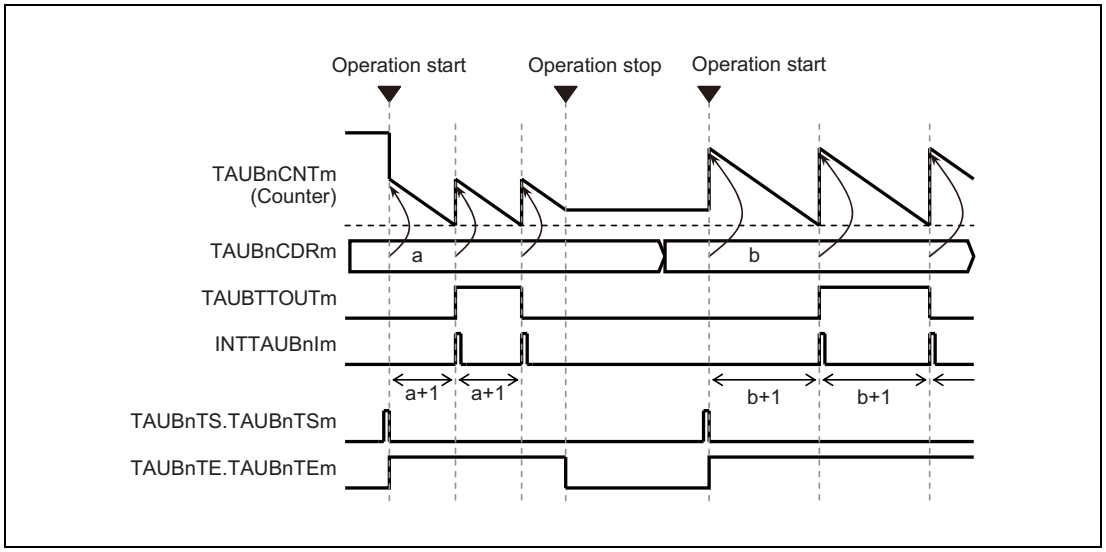


Figure 24.29 Operation Stop and Restart, TAUBnCMORm.TAUBnMD0 = 0

(5) Forced restart (TAUBnCMORm.TAUBnMD0 = 1)

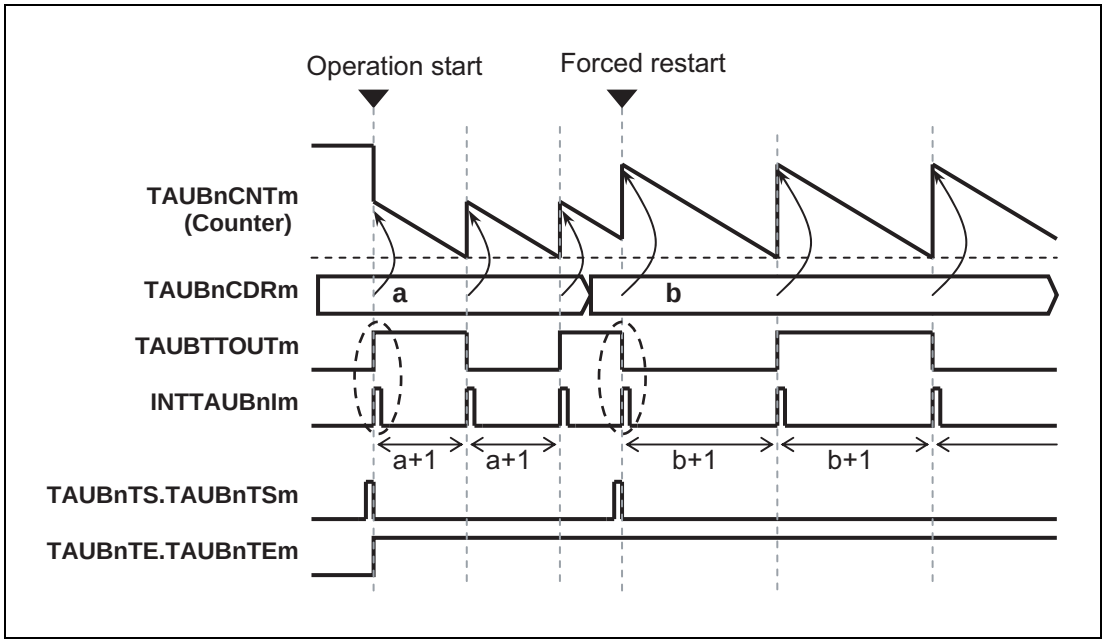


Figure 24.30 Forced Restart Operation, TAUBnCMORm.TAUBnMD0 = 1

- The counter can be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.
- If the TAUBnCMORm.TAUBnMD0 bit is set to 1, an interrupt is generated at start or restart and the output TAUBTTOUTm toggles.

(6) Forced restart (TAUBnCMORm.TAUBnMD0 = 0)

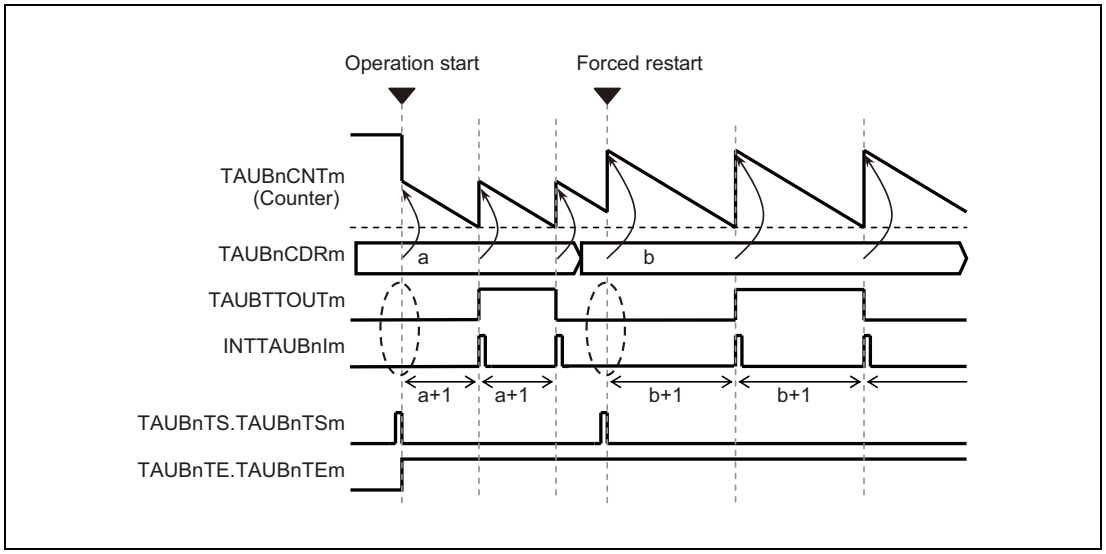


Figure 24.31 Forced Restart Operation (TAUBnCMORm.TAUBnMD0 = 0)

- The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.
- If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUBTTOUTm does not toggle.

24.12.2 TAUBTTINm Input Interval Timer Function

24.12.2.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUBnIm) at regular intervals or when a valid TAUBTTINm input edge is detected.

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The current value of TAUBnCDRm is loaded to TAUBnCNTm and the counter starts to count down from this value.

INTTAUBnIm is generated when the counter reaches 0000_H or by an effective TAUBTTINm input edge. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm stop but retain their values. The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.

The type of edge used as the trigger is specified using the TAUBnCMURm.TAUBnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

24.12.2.2 Block Diagram and General Timing Diagram

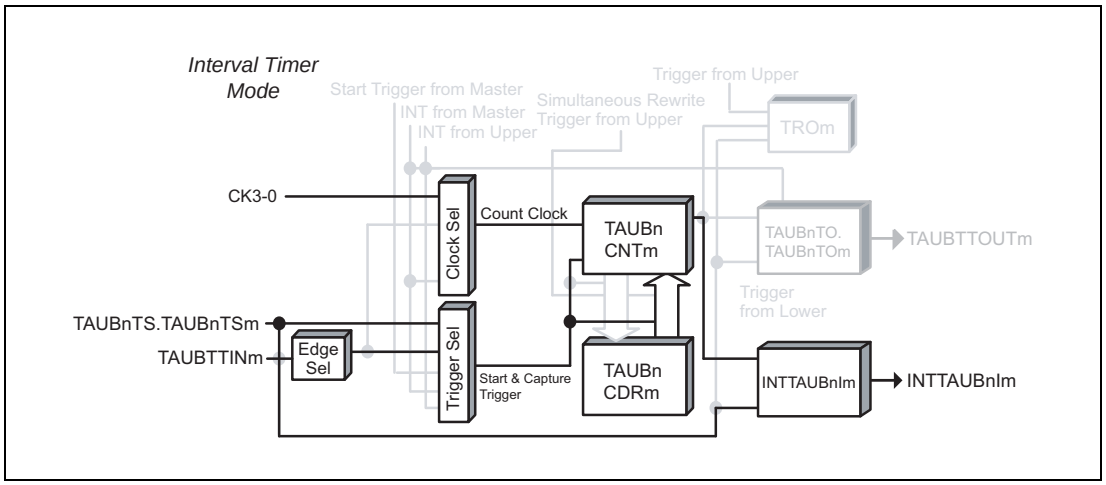


Figure 24.32 Block Diagram for TAUBTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1).
- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

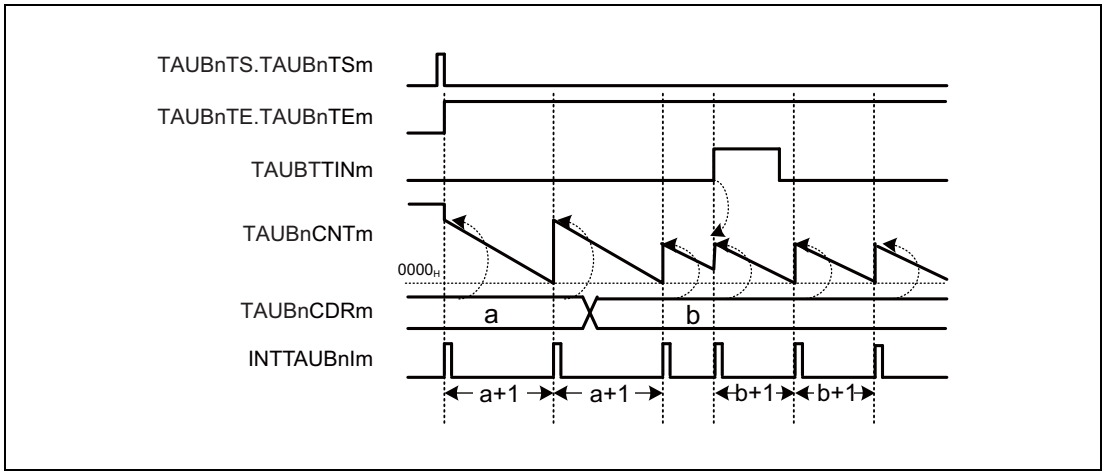


Figure 24.33 General Timing Diagram for TAUBTTINm Input Interval Timer Function

24.12.2.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.44 Contents of the TAUBnCMORm Register for TAUBTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated at operation start 1: Generates INTTAUBnIm at operation start

(2) TAUBnCMURm

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

Table 24.45 Contents of the TAUBnCMURm Register for TAUBTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

This function does not use channel output mode.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Interval Timer Function. Therefore, these registers must be set to 0.

Table 24.46 Simultaneous Rewrite Settings for TAUBTTINm Input Interval Timer Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.2.4 Operating Procedure for TAUBTTINm Input Interval Timer Function

Table 24.47 Operating Procedure for TAUBTTINm Input Interval Timer Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 24.44, Contents of the TAUBnCMORm Register for TAUBTTINm Input Interval Timer Function** and **Table 24.45, Contents of the TAUBnCMURm Register for TAUBTTINm Input Interval Timer Function** Set the value of the TAUBnCDRm register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated.
During operation	The values of the TAUBnCMURm.TAUBnTIS[1:0] and the TAUBnCDRm register can be changed at any time. The TAUBnCNTm register can be read at all times. Detection of TAUBTTINm edge	TAUBnCNTm counts down. When the counter reaches 0000_H: • TAUBnCNTm reloads the TAUBnCDRm value and continues count operation • INTTAUBnIm is generated When a TAUBTTINm input valid edge is detected during count operation, TAUBnCNTm reloads the TAUBnCDRm value and continues count operation. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1 TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

24.12.2.5 Specific Timing Diagrams

The timing diagrams in **Section 24.12.1, Interval Timer Function** apply, and in addition the counter can also be restarted by an effective TAUBTTINm input edge.

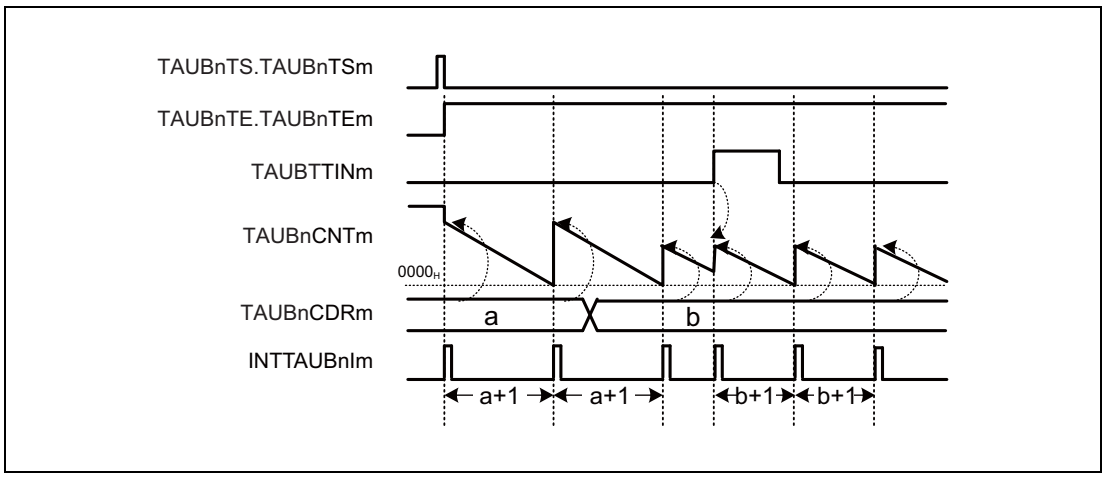


Figure 24.34 Counter Triggered by Rising TAUBTTINm Input Edge (TAUBnCMURm.TAUBnTIS[1:0] = 01_B), TAUBnCMORm.TAUBnMD0 = 1

- If an effective TAUBTTINm input edge is detected, an interrupt INTTAUBnIm is generated. In this example, the effective edge is a rising edge (TAUBnCMURm.TAUBnTIS[1:0] = 01_B).

24.12.3 Clock Divide Function

24.12.3.1 Overview

Summary

This function is used as a frequency divider. The frequency of the input signal TAUBTTINm is divided by a factor related to TAUBnCDRm, and an interrupt INTTAUBnIm is generated.

Prerequisites

- TAUBTTINm must have a fixed frequency
- The operation mode must be set to interval timer mode, see **Table 24.48, Contents of the TAUBnCMORm Register for Clock Divide Function**

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value, using TAUBTTINm as the count clock.

When the counter value reaches 0000_H, INTTAUBnIm is generated. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm = 1, which in turn sets TAUBnTE.TAUBnTEm = 0. TAUBnCNTm stops but retains its value. The function can be restarted by setting TAUBnTS.TAUBnTSm = 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.

Conditions

If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated.

NOTE

The TAUBTTINm input signal is sampled at the frequency of the operation clock, specified by TAUBnCMORm.TAUBnCKS[1:0] bits.

24.12.3.2 Block Diagram and General Timing Diagram

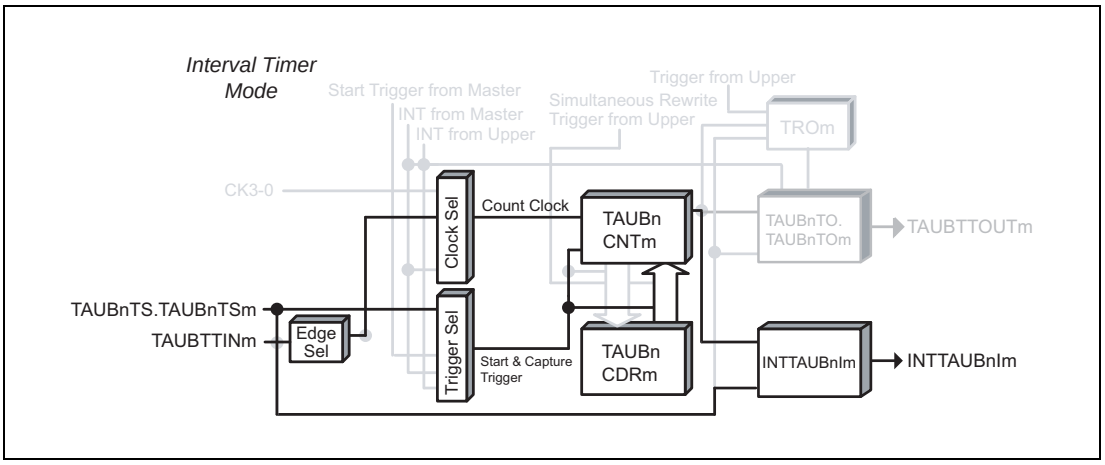


Figure 24.35 Block Diagram for Clock Divide Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

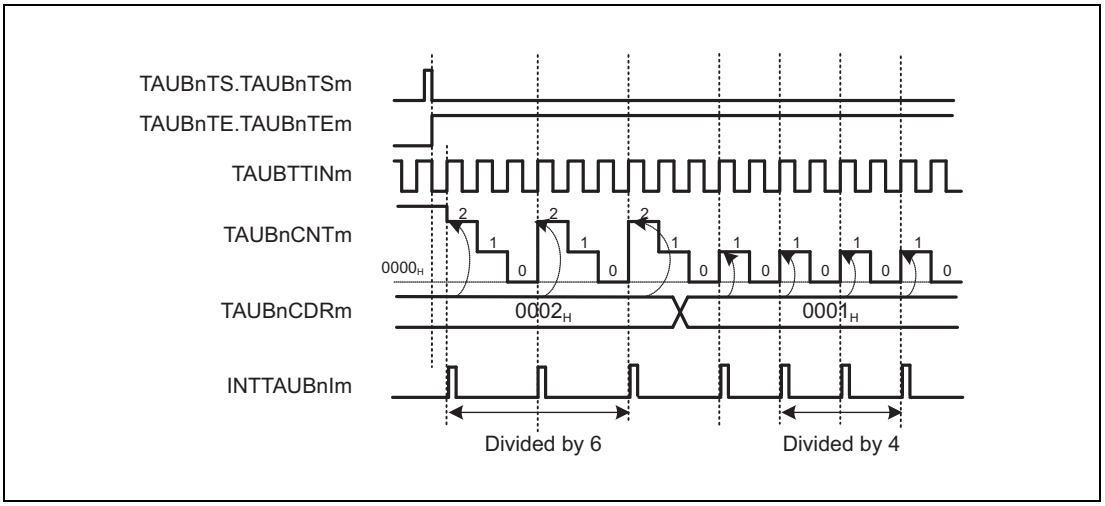


Figure 24.36 General Timing Diagram for Clock Divide Function

24.12.3.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.48 Contents of the TAUBnCMORm Register for Clock Divide Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 1_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated at operation start 1: Generates INTTAUBnIm at operation start

(2) TAUBnCMURm

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

Table 24.49 Contents of the TAUBnCMURm Register for Clock Divide Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

This function does not use channel output mode.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Clock Divide Function. Therefore, these registers must be set to 0.

Table 24.50 Simultaneous Rewrite Settings for Clock Divide Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.3.4 Operating Procedure for Clock Divide Function

Table 24.51 Operating Procedure for Clock Divide Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 24.48, Contents of the TAUBnCMORm Register for Clock Divide Function** and **Table 24.49, Contents of the TAUBnCMURm Register for Clock Divide Function** Set the value of the TAUBnCDRm register. Set the control bits as described in **Table 24.49, Contents of the TAUBnCMURm Register for Clock Divide Function**.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 is set to 1, INTTAUBnIm is generated.
During operation	The value of TAUBnCDRm can be changed at any time. The TAUBnCNTm register can be read at all times.	When a TAUBTTINm input edge is detected, TAUBnCNTm counts down. When the counter reaches 0000_H: • TAUBnCNTm loads the TAUBnCDRm value and continues count operation • INTTAUBnIm is generated
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

24.12.3.5 Specific Timing Diagrams

(1) TAUBnCDRm = 0000_H

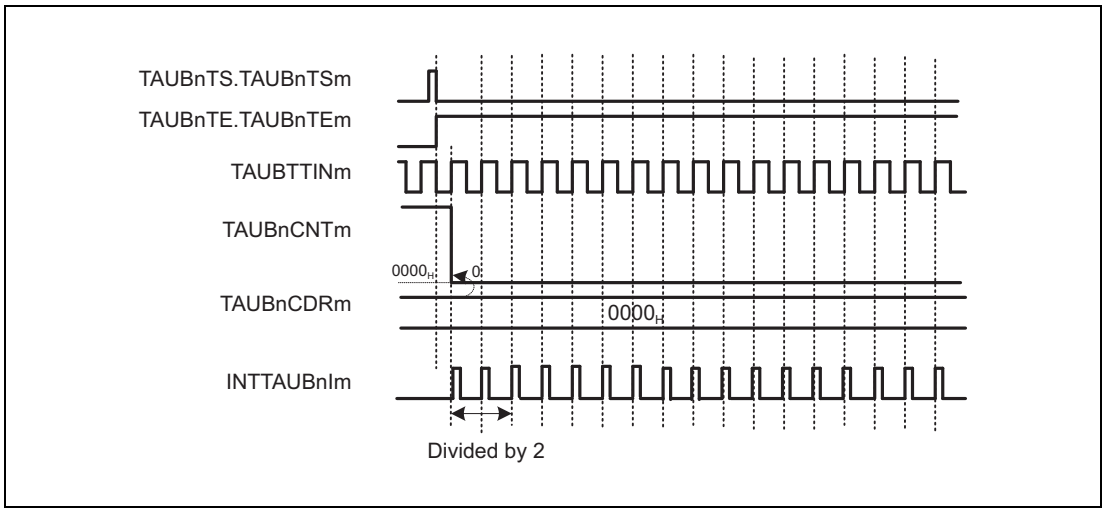


Figure 24.37 TAUBnCDRm = 0000_H, TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01_B

- If TAUBnCDRm is 0000_H, TAUBnCNTm is also always 0000_H.
- INTTAUBnIm is generated every count clock.

Figure 24.37 is an image of the operation timing. Actually, there is a delay time caused by the noise filter and the synchronization circuit between the TAUBnIm terminal and TAUBn.

(2) Restart

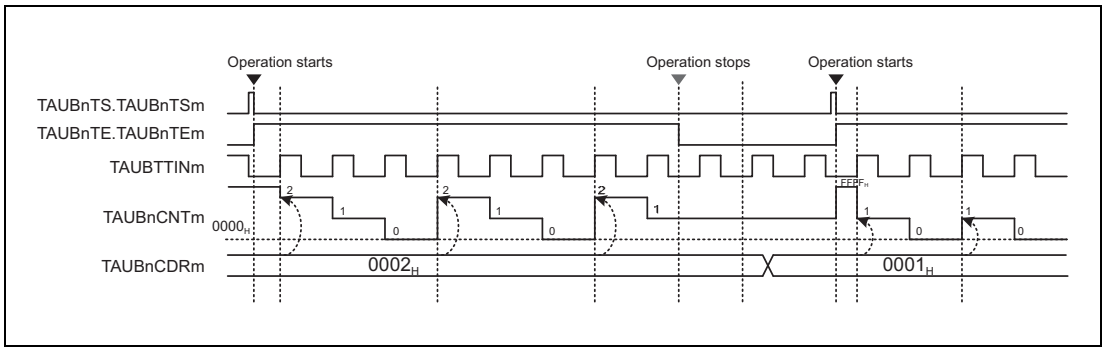


Figure 24.38 Restart (TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

(3) Forced restart

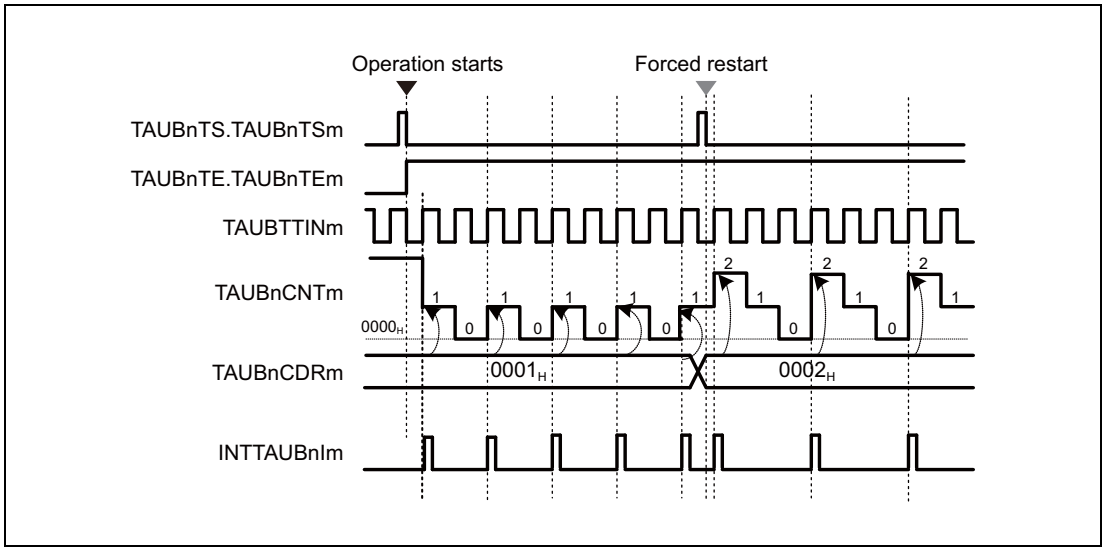


Figure 24.39 Forced Restart (TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

To forcibly restart the counter.

- The counter can be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.
- The value of TAUBnCDRm is written to TAUBnCNTm and the count operation restarts.

24.12.4 External Event Count Function

24.12.4.1 Overview

Summary

This function is used as an event timer. It generates an interrupt (INTTAUBnIm) when a specific number of valid edges of TAUBTTINm input are detected.

Prerequisites

- The operation mode must be set to event count mode, see **Table 24.52, Contents of the TAUBnCMORm Register for External Event Count Function**
- TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. When the counter starts, the current value of TAUBnCDRm is written to TAUBnCNTm.

When an effective TAUBTTINm input edge is detected, the value of TAUBnCNTm reduces. TAUBnCNTm retains this value until a valid TAUBTTINm input edge is detected or the counter is restarted.

When effective edges are detected (TAUBnCDRm + 1) times, INTTAUBnIm is generated. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues to operate.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0. The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUBnTS.TAUBnTSm to 1 during operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as the trigger is specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

24.12.4.2 Equations

Number of valid edges,
detected before INTTAUBnIm is generated = TAUBnCDRm + 1

24.12.4.3 Block Diagram and General Timing Diagram

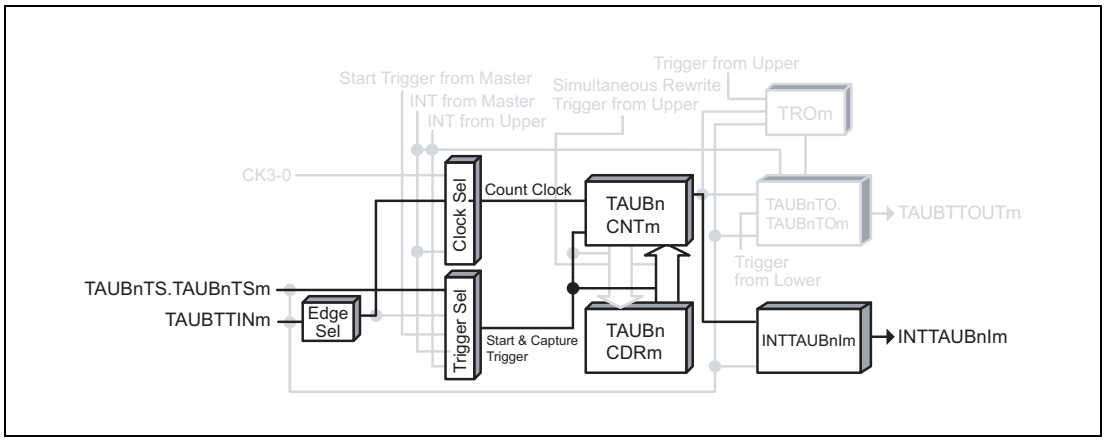


Figure 24.40 Block Diagram for External Event Count Function

The following settings apply to the general timing diagram.

- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

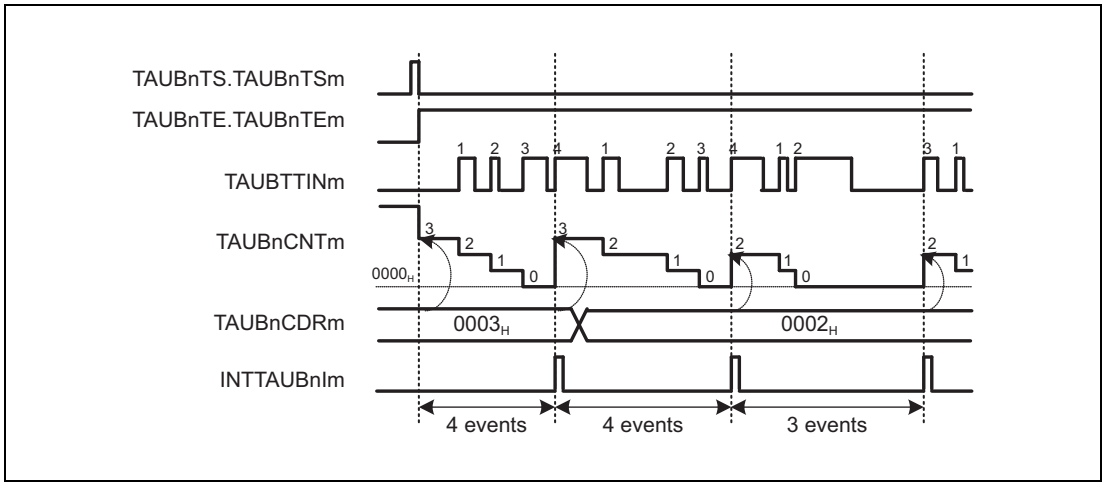


Figure 24.41 General Timing Diagram for External Event Count Function

24.12.4.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.52 Contents of the TAUBnCMORm Register for External Event Count Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 1_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0011_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

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

Table 24.53 Contents of the TAUBnCMURm Register for External Event Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge is detected. 01: Rising edge is detected. 10: Both edges are detected. 11: Setting prohibited

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the External Event Count Function.

Therefore, these registers must be set to 0.

Table 24.54 Simultaneous Rewrite Settings for External Event Count Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.4.5 Operating Procedure for External Event Count Function

Table 24.55 Operating Procedure for External Event Count Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registersas described in **Table 24.52, Contents of the TAUBnCMORm Register for External Event Count Function** and **Table 24.53, Contents of the TAUBnCMURm Register for External Event Count Function** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value and waits for detection of the TAUBTTINm input edge.
During operation	Detection of TAUBTTINm edges. The value of TAUBnCDRm can be changed at any time. The TAUBnCNTm register can be read at any time.	TAUBnCNTm performs count-down operation each time a TAUBTTINm input edge is detected. When effective edges are detected (TAUBnCDRm + 1) times: • TAUBnCNTm loads the TAUBnCDRm value and continues count operation • INTTAUBnIm is generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

24.12.4.6 Specific Timing Diagrams

(1) TAUBnCDRm = 0000_H

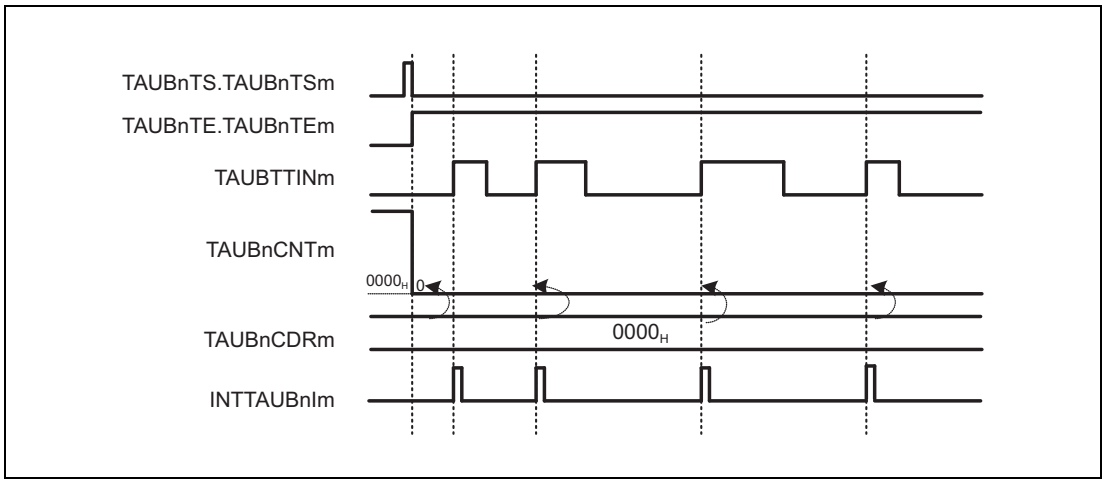


Figure 24.42 TAUBnCDRm = 0000_H, TAUBnCMURm.TAUBnTIS[1:0] = 01_B

- If 0000_H = TAUBnCDRm, 0000_H is loaded to TAUBnCNTm every time a valid TAUBTTINm input edge is detected.

  This means, INTTAUBnIm is generated every time a valid TAUBTTINm input edge is detected.

(2) Operation stop and restart

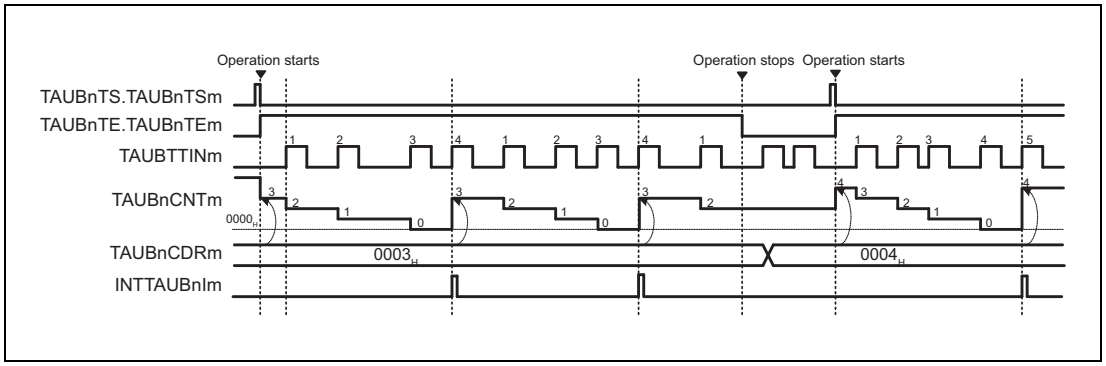


Figure 24.43 Operation Stop and Restart (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm stops and the current value is retained. TAUBTTINm continues and TAUBnCNTm ignores the valid edge.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm loads the TAUBnCDRm value and restarts count operation.

(3) Forced restart

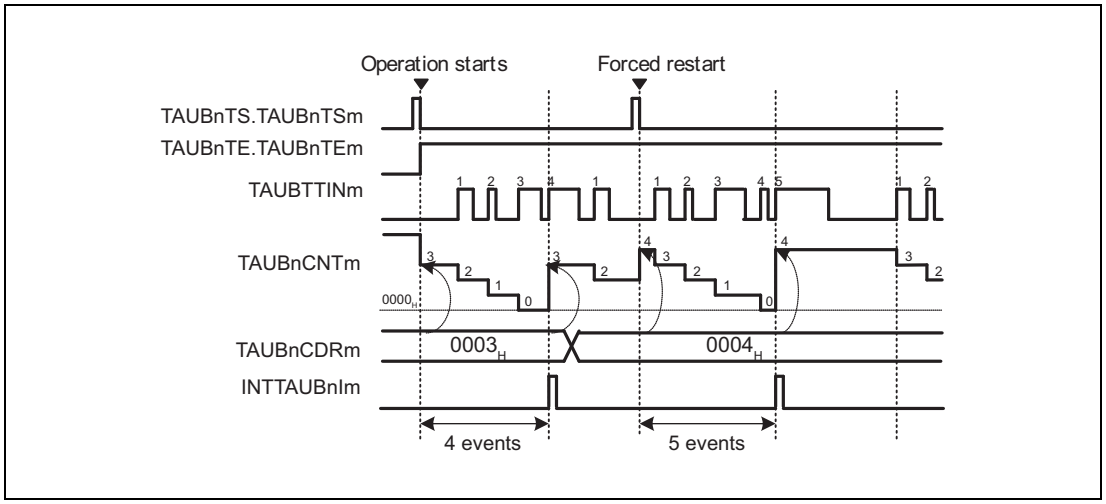


Figure 24.44 Forced Restart (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

A forced restart applies the new TAUBnCDRm value to TAUBnCNTm.

- The counter can be restarted (without stopping it first), by setting TAUBnTS.TAUBnTSm to 1 during operation.
- The value of TAUBnCDRm is loaded to TAUBnCNTm and the counter awaits the next valid TAUBTTINm input edge.

24.12.5 One-Pulse Output Function

24.12.5.1 Overview

Summary

This function generates an interrupt (INTTAUBnIm) when a valid TAUBTTINm input edge is detected and continues to generate interrupts at the specified interval. TAUBTTINm input signal pulses that occur within the defined interval are ignored.

Prerequisites

- The operation mode must be set to pulse one-count mode. (See **Table 24.56, Contents of the TAUBnCMORm Register for One-Pulse Output Function**).
- Trigger detection must be disabled during counting (TAUBnCMORm.TAUBnMD0 = 0).

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The counter starts when a valid TAUBTTINm input edge is detected. The value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from the TAUBnCDRm value, and an interrupt is generated.

When the counter reaches 0001_H an interrupt is generated. The counter stops at 0000_H and awaits the next effective TAUBTTINm input edge.

When the counter is counting down, further TAUBTTINm input signals are ignored, i.e. the counter does not reset.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as the trigger is specified by the TAUBnCMURm.TIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

24.12.5.2 Equations

Interval between TAUBTTINm and INTTAUBnIm = count clock cycle × TAUBnCDRm

24.12.5.3 Block Diagram and General Timing Diagram

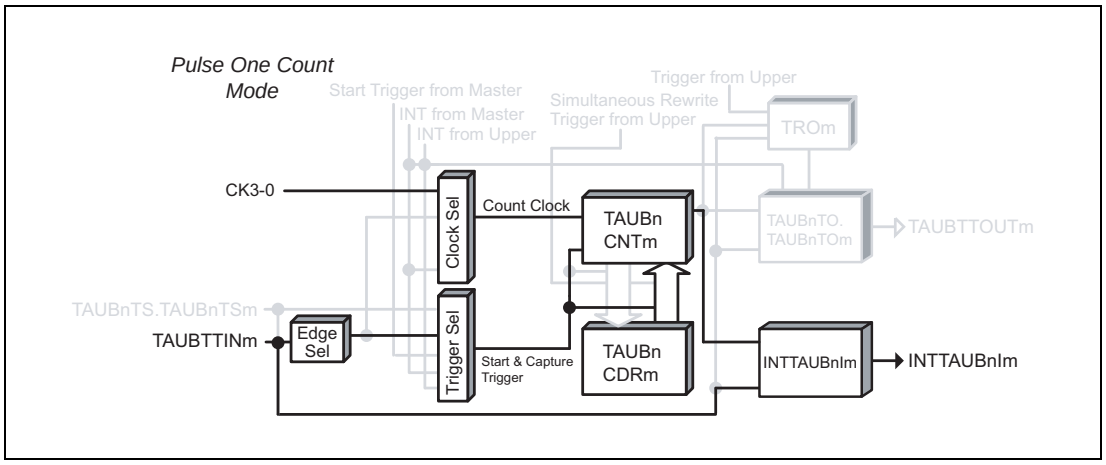


Figure 24.45 Block Diagram for One-Pulse Output Function

The following settings apply to the general timing diagram.

- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

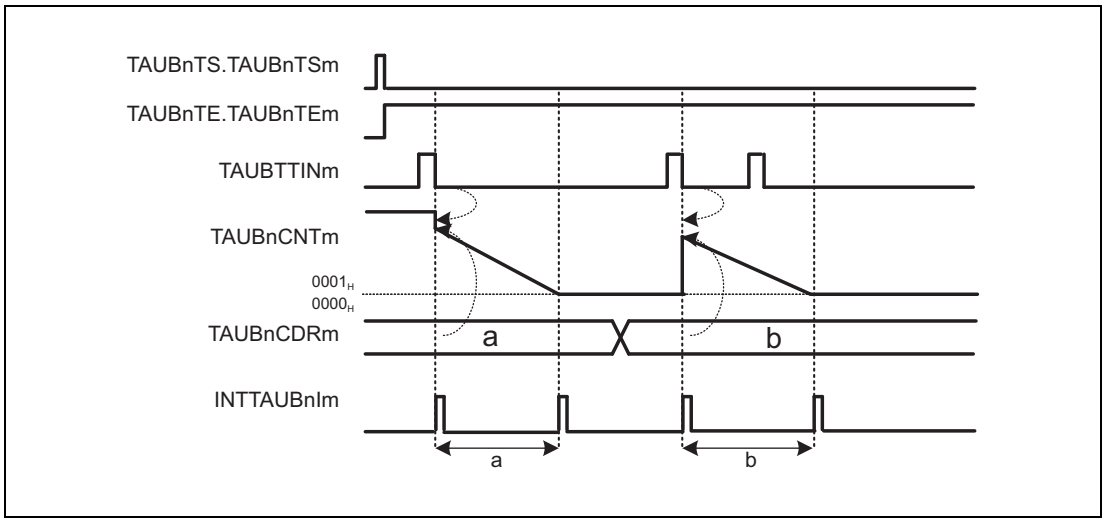


Figure 24.46 General Timing Diagram for One-Pulse Output Function

24.12.5.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.56 Contents of the TAUBnCMORm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

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

Table 24.57 Contents of the TAUBnCMURm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

This function does not use channel output mode.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the One-Pulse Output Function.
Therefore, these registers must be set to 0.

Table 24.58 Simultaneous Rewrite Settings for One-Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.5.5 Operating Procedure for One-Pulse Output Function

Table 24.59 Operating Procedure for One-Pulse Output Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in**Table 24.56, Contents of the TAUBnCMORm Register for One-Pulse Output Function** and **Table 24.57, Contents of the TAUBnCMURm Register for One-Pulse Output Function** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUBTTINm start edge	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value.
During operation	The value of TAUBnCDRm can be changed at any time. The TAUBnCNTm register can be read at all times.	INTTAUBnIm is generated when TAUBnCNTm starts. TAUBnCNTm counts down. When the counter reaches 0001_H, INTTAUBnIm is generated. TAUBnCNTm stops counting and waits for a trigger. If a trigger occurs while TAUBnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

24.12.6 TAUBTTINm Input Pulse Interval Measurement Function

24.12.6.1 Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUBnCSRm.TAUBnOVF to measure the interval of the TAUBTTINm input signal.

Prerequisites

- TAUBTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The counter TAUBnCNTm starts counting up from 0000_H. When a valid TAUBTTINm edge is detected, the value of TAUBnCNTm is captured, transferred to TAUBnCDRm, and an interrupt INTTAUBnIm is generated. The counter resets to 0000_H and subsequently continues operation.

If the counter reaches $FFFF_H$ before a valid TAUBTTINm edge is detected, it overflows to 0000_H. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUBnCDRm and TAUBnCSRm.TAUBnOVF respectively depend on the values of bits TAUBnCMORm.TAUBnCOS[1:0].

Table 24.60 Effects of an Overflow

TAUBnCMORm.COS[1:0]	When Overflow Occurs		When a Valid TAUBTTINm Input is then Detected	
	TAUBnCDRm	TAUBnCSRm.TAUBnOVF	TAUBnCDRm, TAUBnCNTm	TAUBnCSRm.TAUBnOVF
00	Unchanged	0	TAUBnCNTm loaded to TAUBnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUBnCNTm set to 0, TAUBnCDRm unchanged	Unchanged
11		1		

If TAUBnCMORm.TAUBnCOS[0] is 1, the overflow bit TAUBnCSRm.TAUBnOVF can only be cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

The combination of the value of TAUBnCDRm and TAUBnCSRm.TAUBnOVF can be used to deduce the interval of the TAUBTTINm signal. However, if an overflow occurs multiple times before a valid TAUBTTINm input is detected, the overflow bit TAUBnCSRm.TAUBnOVF cannot indicate this.

The function can be stopped by setting TAUBnTT.TAUBnTTm = 1, which in turn sets TAUBnTE.TAUBnTEm = 0. TAUBnCNTm stops but retains its value. While the function is stopped, TAUBTTINm input valid edge detection and TAUBnCNTm capture are not performed.

The counter is reset to 0000_H and subsequently continues operation.

Conditions

If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the interrupt at start or restart is not generated.

NOTE

When TAUBnCMORm.TAUBnCOS[1:0] = 10_B or 11_B, the value of TAUBnCNTm is not written to TAUBnCDRm when the first valid TAUBTTINm input edge occurs after an overflow. However, an interrupt is generated.

24.12.6.2 Equations

TAUBTTINm input pulse interval = count clock cycle ×

[(TAUBnCSRm.TAUBnOVF × ($FFFF_H$ + 1)) + TAUBnCDRm capture value + 1]

24.12.6.3 Block Diagram and General Timing Diagram

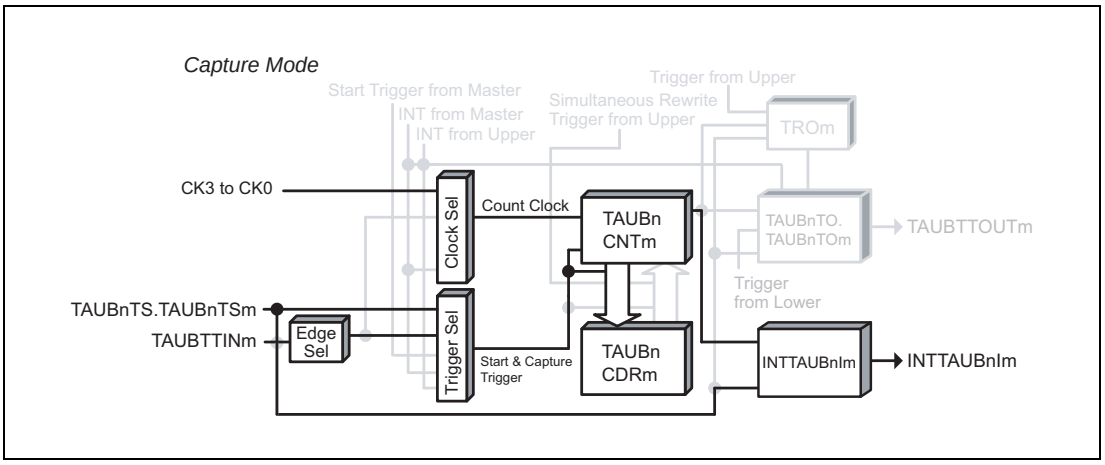


Figure 24.47 Block Diagram for TAUBTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)
- When a valid TAUBTTINm input is detected after an overflow TAUBnCDRm is changed and TAUBnCSRm.TAUBnOVF is set to 1 (TAUBnCMORm.TAUBnCOS[1:0] = 00_B)

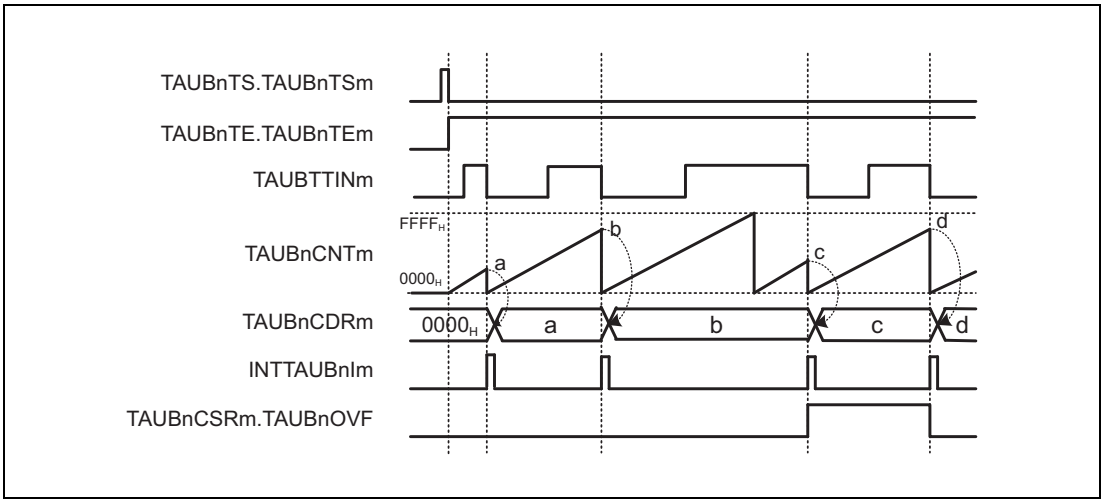


Figure 24.48 General Timing Diagram for TAUBTTINm Input Pulse Interval Measurement Function

24.12.6.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.61 Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	See **Table 24.60, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0010_B.
0	TAUBnMD0	0: INTTAUBnIm not generated at operation start 1: Generates INTTAUBnIm at operation start

(2) TAUBnCMURm

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

Table 24.62 Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Pulse Interval Measurement Function. Therefore, these registers must be set to 0.

Table 24.63 Simultaneous Rewrite Settings for TAUBTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.6.5 Operating Procedure for TAUBTTINm Input Pulse Interval Measurement Function

Table 24.64 Operating Procedure for TAUBTTINm Input Pulse Interval Measurement Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 24.61, Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Measurement Function** and **Table 24.62, Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Measurement Function** The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm is cleared to 0000_H. INTTAUBnIm is generated when TAUBnCMORm.TAUBnMD0 is set to 1.
During operation	Detection of TAUBTTINm edges. The values of TAUBnCMURm.TAUBnTIS[1:0] bits can be changed at any time. The TAUBnCDRm and TAUBnCSRm registers can be read at any time. The TAUBnCSCm.TAUBnCLOV bit can be set to 1. (The TAUBnCSRm.TAUBnOVF bit can be cleared to 0.)	TAUBnCNTm starts to count up from 0000_H. When a TAUBTTINm valid edge is detected: • TAUBnCNTm transfers (captures) its value to TAUBnCDRm, and returns to 0000_H • INTTAUBnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and both it and TAUBnCSRm.TAUBnOVF retain their current values.

Restart operation

24.12.6.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUBnCMORm.TAUBnCOS[1:0] = 00_B

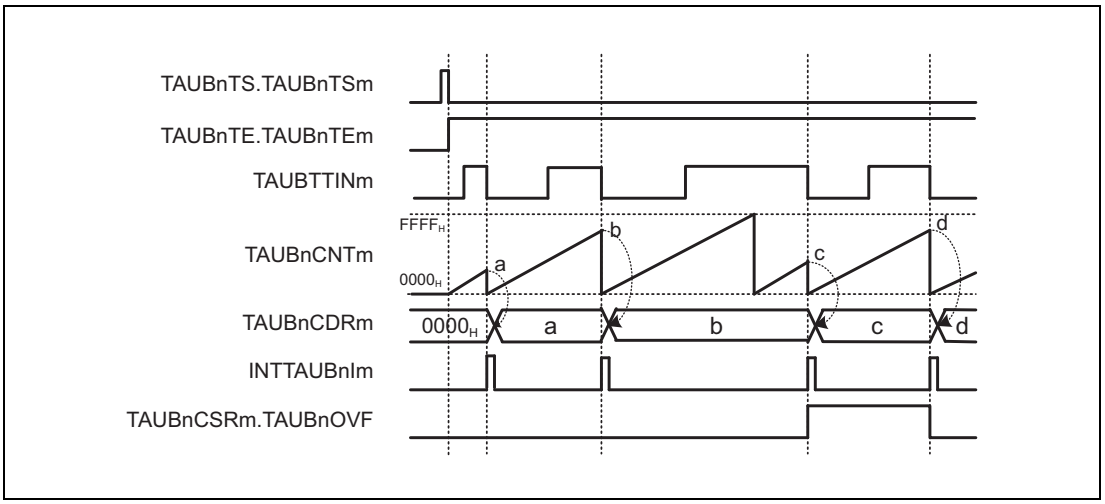


Figure 24.49 TAUBnCMORm.TAUBnCOS[1:0] = 00_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is loaded to TAUBnCDRm and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, while any overflow has not occurred, TAUBnCSRm.TAUBnOVF is cleared to 0.

(2) TAUBnCMORm.TAUBnCOS[1:0] = 01_B

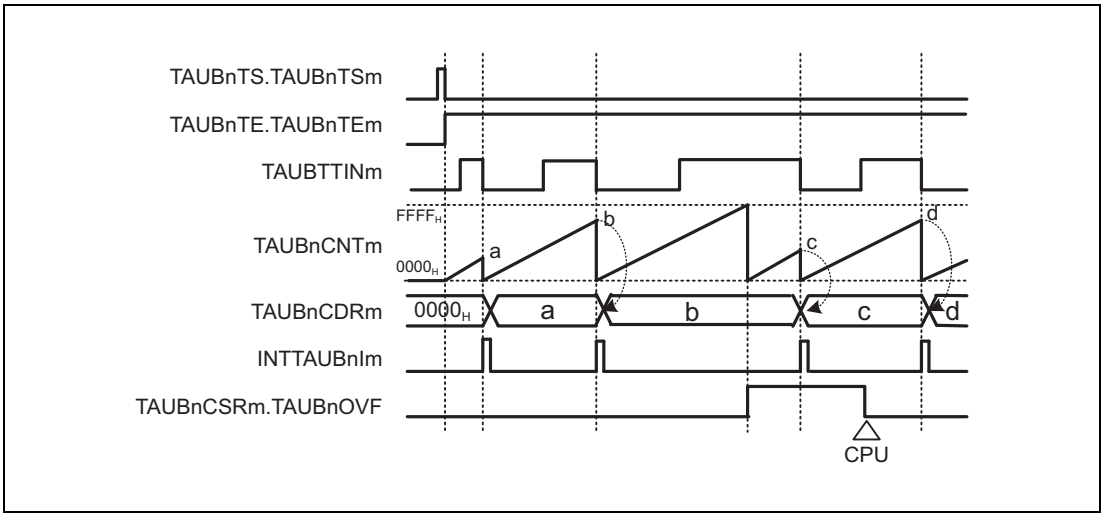


Figure 24.50 TAUBnCMORm.TAUBnCOS[1:0] = 01_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is written to TAUBnCDRm.
- TAUBnCSRm.TAUBnOVF is only cleared by a CPU command. (TAUBnCSCm.TAUBnCLOV bit = 1)

(3) TAUBnCMORm.TAUBnCOS[1:0] = 10_B

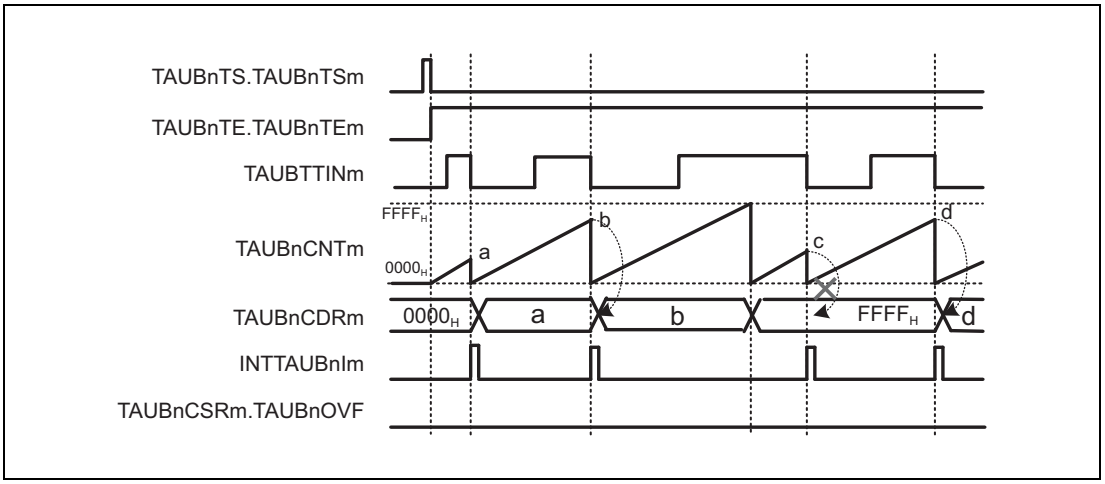


Figure 24.51 TAUBnCMORm.TAUBnCOS[1:0] = 10_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$ and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm is reset to 0, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.

(4) TAUBnCMORm.TAUBnCOS[1:0] = 11_B

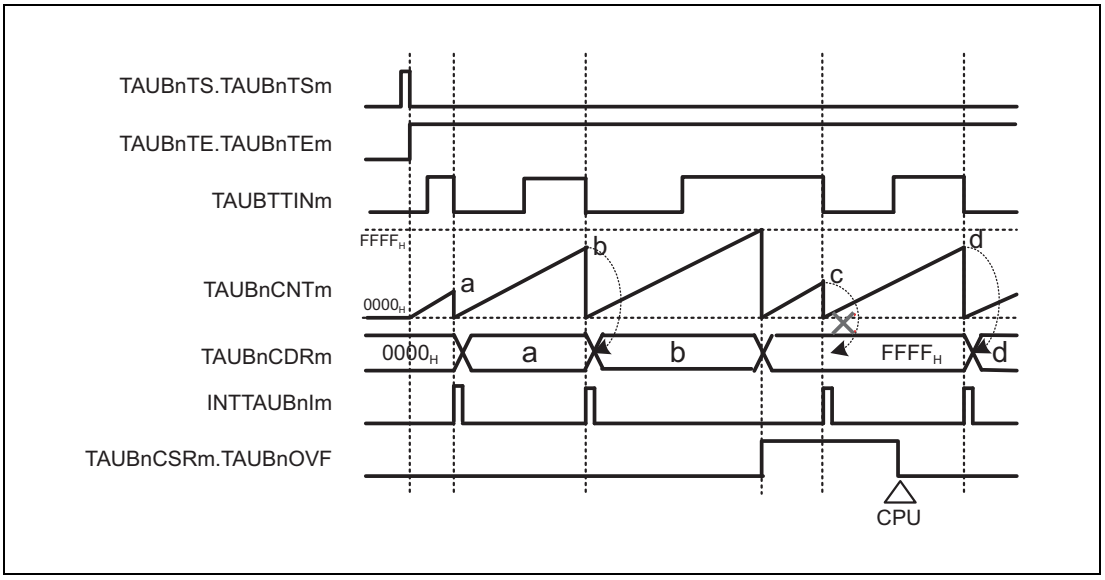


Figure 24.52 TAUBnCMORm.TAUBnCOS[1:0] = 11_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$, and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm is reset to 0, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.
- TAUBnCSRm.TAUBnOVF is cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

(5) When rising and falling edge detection are selected (TAUBnCMORm.TAUBnMD0 = 1)

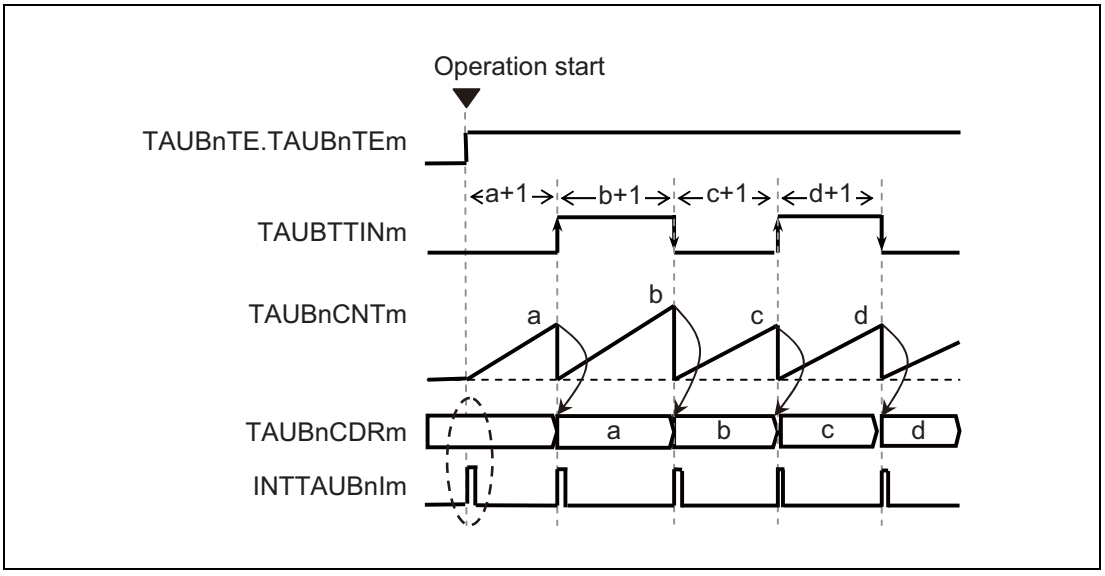


Figure 24.53 TAUBnCMORm.TAUBnMD0 = 1

Setting TAUBnCMURm.TAUBnTIS[1:0] to 10_B (detection of both edges selected) measures the TAUBTTINm rising and falling edge intervals.

(6) Operation stop and operation restart (TAUBnCMORm.TAUBnMD0 = 0)

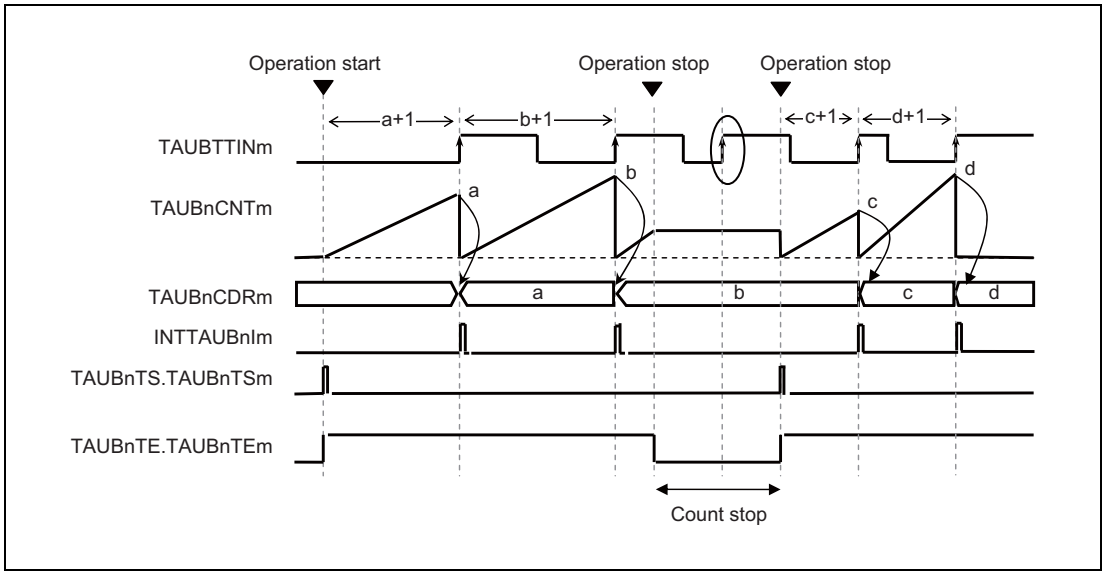


Figure 24.54 Operation Stop and Operation Restart (TAUBnCMORm.TAUBnMD0 = 0)

Setting TAUBnTT.TAUBnTTm to 1 clears TAUBnTE.TAUBnTEm to 0, which stops the count operation. At this time, TAUBnCNTm retains the status and stops.

When TAUBnTE.TAUBnTEm retains 0 (operation stopped), TAUBTTINm input is ignored (edge detection is ignored and capture operation is not performed).

Setting TAUBnTS.TAUBnTSm to 1 clears the counter to 0000_H and restarts count-up operation.

24.12.7 TAUBTTINm Input Signal Width Measurement Function

24.12.7.1 Overview

Summary

This function measures the width of a TAUBTTINm signal by starting counting on one edge of the TAUBTTINm signal and capturing the counter value on the opposite edge.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.

This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. When a valid TAUBTTINm start edge is detected, the counter TAUBnCNTm starts counting up from 0000_H. When a valid TAUBTTINm stop edge is detected, the value of TAUBnCNTm is captured, transferred to TAUBnCDRm, and an interrupt INTTAUBnIm is generated. The counter retains its value (TAUBnCDRm + 1) and awaits the next valid TAUBTTINm input start edge.

If the counter reaches $FFFF_H$ before a valid TAUBTTINm stop edge is detected, it overflows. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUBnCDRm and TAUBnCSRm.TAUBnOVF respectively depend on the values of bits TAUBnCMORm.TAUBnCOS[1:0].

Table 24.65 Effects of an Overflow

TAUBnCMORm.COS[1:0]	When Overflow Occurs		When a Valid TAUBTTINm Input Stop Edge is Detected	
	TAUBnCDRm	TAUBnCSRm.TAUBnOVF	TAUBnCDRm, TAUBnCNTm	TAUBnCSRm.TAUBnOVF
00	Unchanged	0	TAUBnCNTm written to TAUBnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUBnCNTm stops counting TAUBnCDRm unchanged	Unchanged
11		1		

If TAUBnCMORm.TAUBnCOS[0] = 1, the overflow bit TAUBnCSRm.TAUBnOVF can only be cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

The combination of the value of TAUBnCDRm and TAUBnCSRm.TAUBnOVF can be used to deduce the width of the TAUBTTINm signal. However, if an overflow occurs multiple times before a valid TAUBTTINm input is detected, the overflow bit TAUBnCSRm.TAUBnOVF cannot indicate this.

This function cannot be forcibly restarted.

NOTE

When TAUBnCMORm.TAUBnCOS[1] = 1, the value of TAUBnCNTm is not written to TAUBnCDRm when the first valid TAUBTTINm input edge occurs after an overflow. However, an interrupt is generated.

24.12.7.2 Equations

TAUBTTINm input signal width = count clock cycle ×

[(TAUBnCSRm.OVF × ($FFFF_H$ + 1)) + TAUBnCDRm capture value + 1]

24.12.7.3 Block Diagram and General Timing Diagram

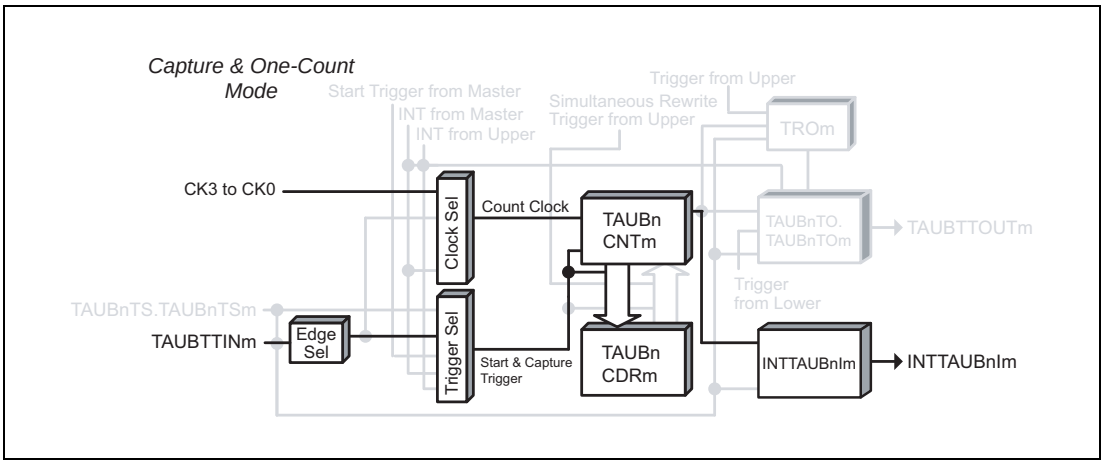


Figure 24.55 Block Diagram for TAUBTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)
- When a valid TAUBTTINm input is detected after an overflow, TAUBnCDRm is changed and TAUBnCSRm.TAUBnOVF is set to 1 (TAUBnCMORm.TAUBnCOS[1:0] = 00_B)

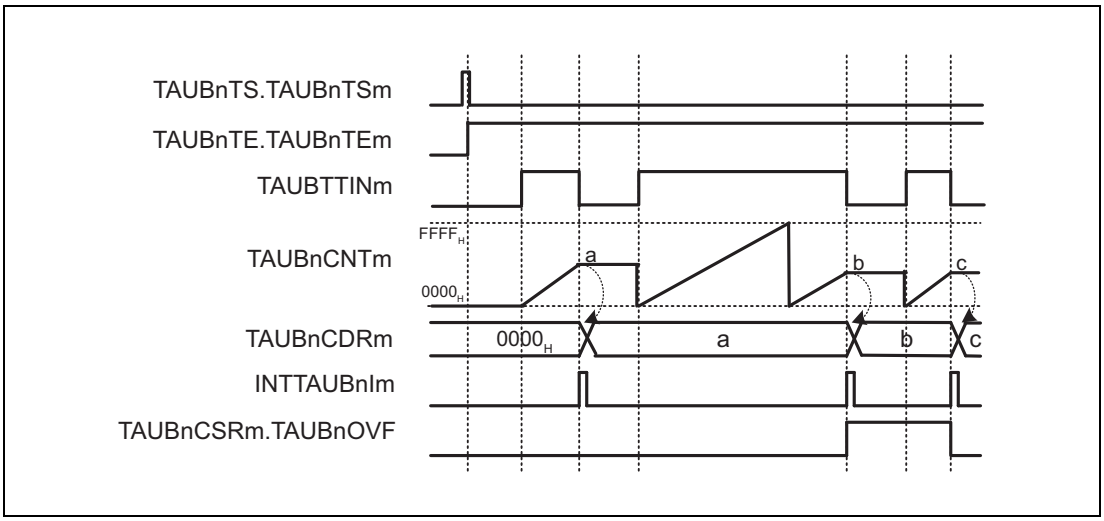


Figure 24.56 General Timing Diagram for TAUBTTINm Input Signal Width Measurement Function

24.12.7.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.66 Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	See **Table 24.65, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0110_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

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

Table 24.67 Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Signal Width Measurement Function. Therefore, these registers must be set to 0.

Table 24.68 Simultaneous Rewrite Settings for TAUBTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.7.5 Operating Procedure for TAUBTTINm Input Signal Width Measurement Function

Table 24.69 Operating Procedure for TAUBTTINm Input Signal Width Measurement Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 24.66, Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Measurement Function** and **Table 24.67, Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Measurement Function** The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When a TAUBTTINm start edge is detected, TAUBnCNTm start edge to count up.
During operation	The TAUBnCDRm, TAUBnCNTm, and TAUBnCSRm registers can be read at any time. TAUBnCSCm.TAUBnCLOV bit can be set to 1.	TAUBnCNTm starts to count up from 0000_H. When a TAUBTTINm valid edge is detected: • TAUBnCNTm transfers (captures) its value to TAUBnCDRm, and retains its value • INTTAUBnIm is then generated. • The count stops at the value transferred to TAUBnCDRm + 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TEm is cleared to 0 and the counter stops. TAUBnCNTm stops and both it and TAUBnCSRm.TAUBnOVF retain their current values.

Restart operation

24.12.7.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUBnCMORm.TAUBnCOS[1:0] = 00_B

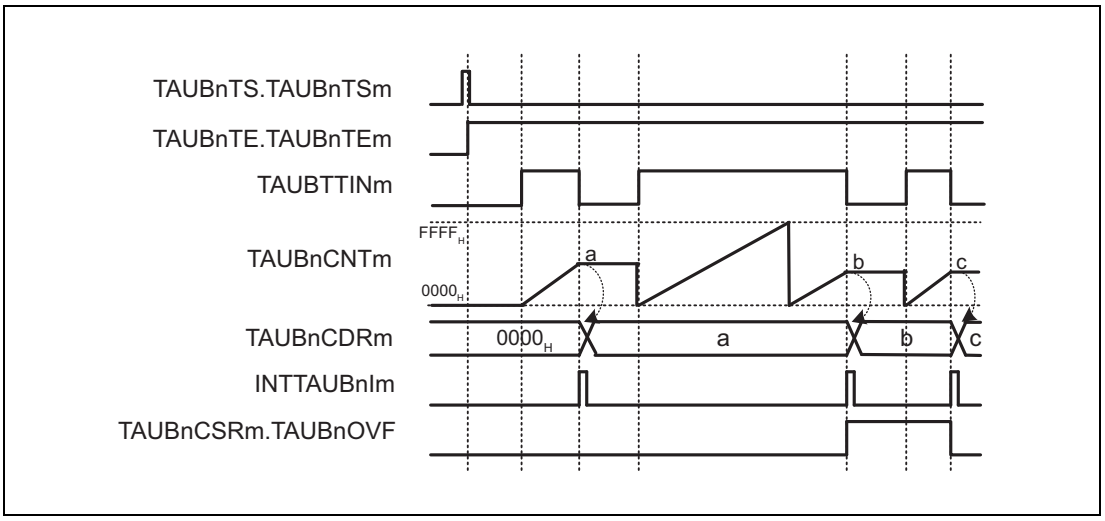


Figure 24.57 TAUBnCMORm.TAUBnCOS[1:0] = 00_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is loaded to TAUBnCDRm and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, while any overflow has not occurred, TAUBnCSRm.TAUBnOVF is cleared to 0.

(2) TAUBnCMORm.TAUBnCOS[1:0] = 01_B

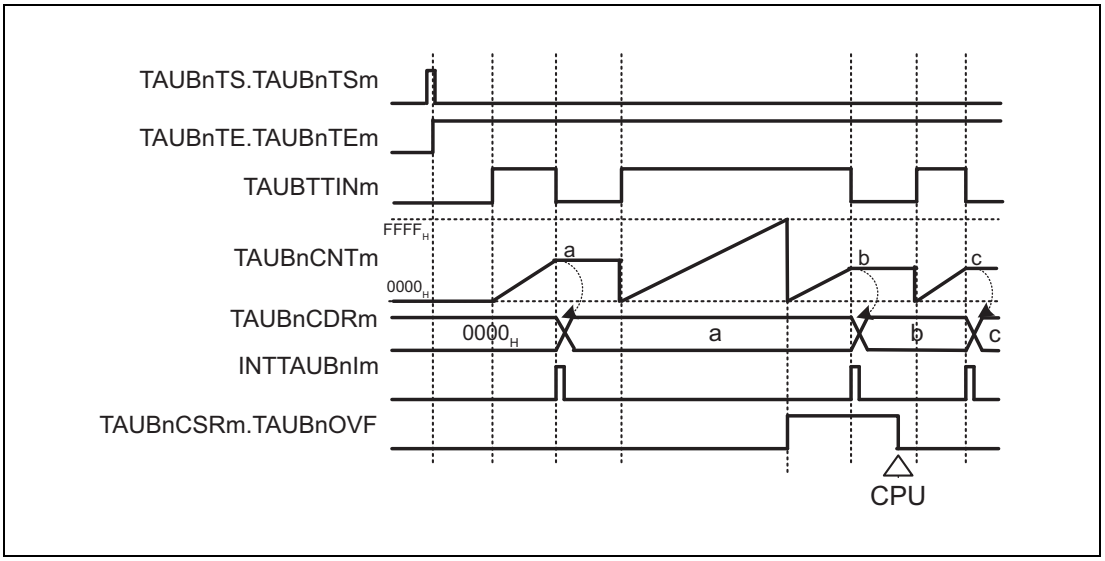


Figure 24.58 TAUBnCMORm.TAUBnCOS[1:0] = 01_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is written to TAUBnCDRm.
- TAUBnCSRm.TAUBnOVF is only cleared by a CPU command (The TAUBnCSCm.TAUBnCLOV bit = 1).

(3) TAUBnCMORm.TAUBnCOS[1:0] = 10_B

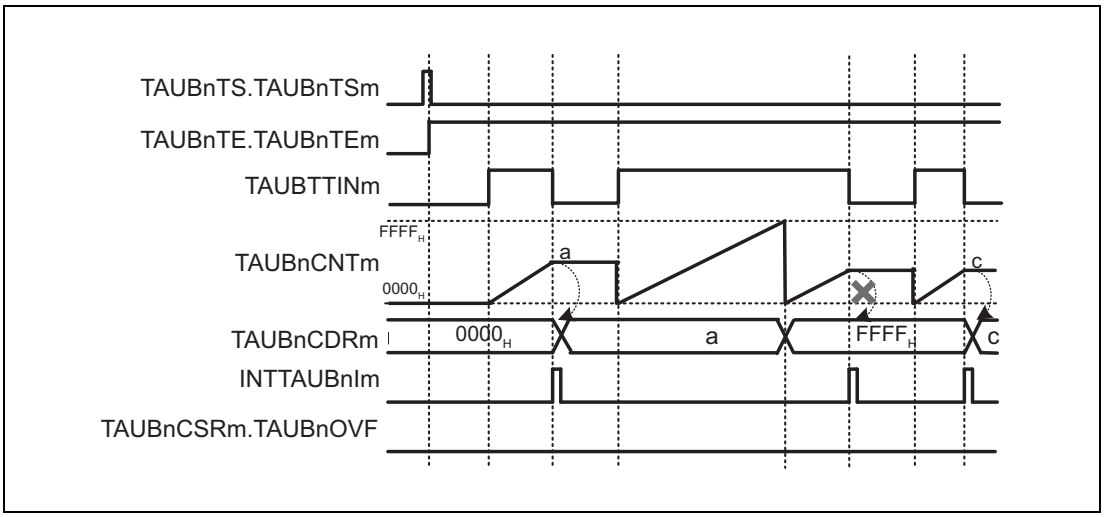


Figure 24.59 TAUBnCMORm.TAUBnCOS[1:0] = 10_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$ and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm stops counting, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.

(4) TAUBnCMORm.TAUBnCOS[1:0] = 11_B

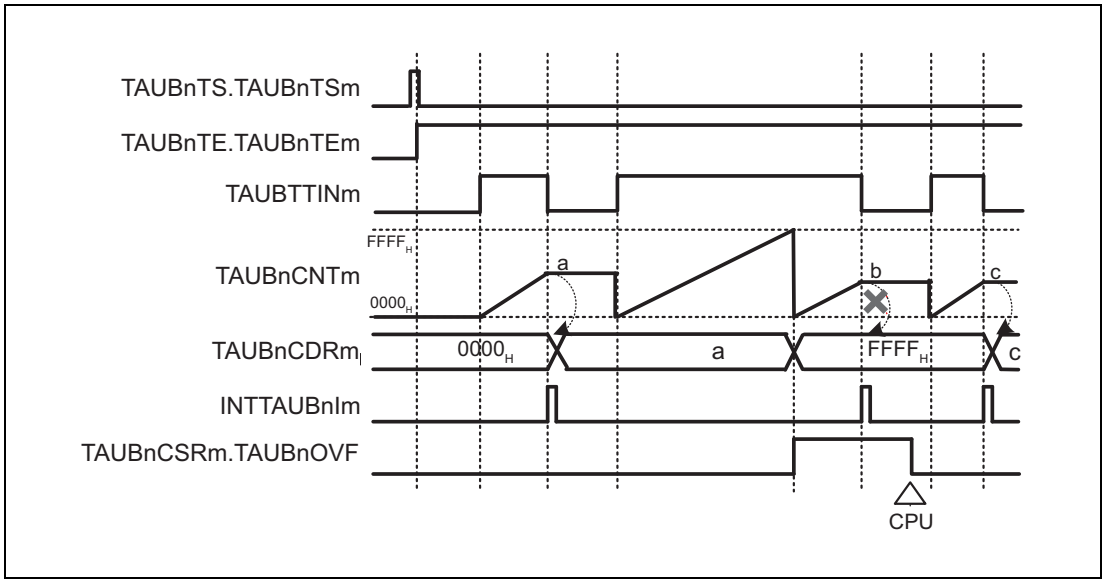


Figure 24.60 TAUBnCMORm.TAUBnCOS[1:0] = 11_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$, and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm stops counting, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.
- TAUBnCSRm.TAUBnOVF is cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

(5) When an overflow occurs (high width measurement)

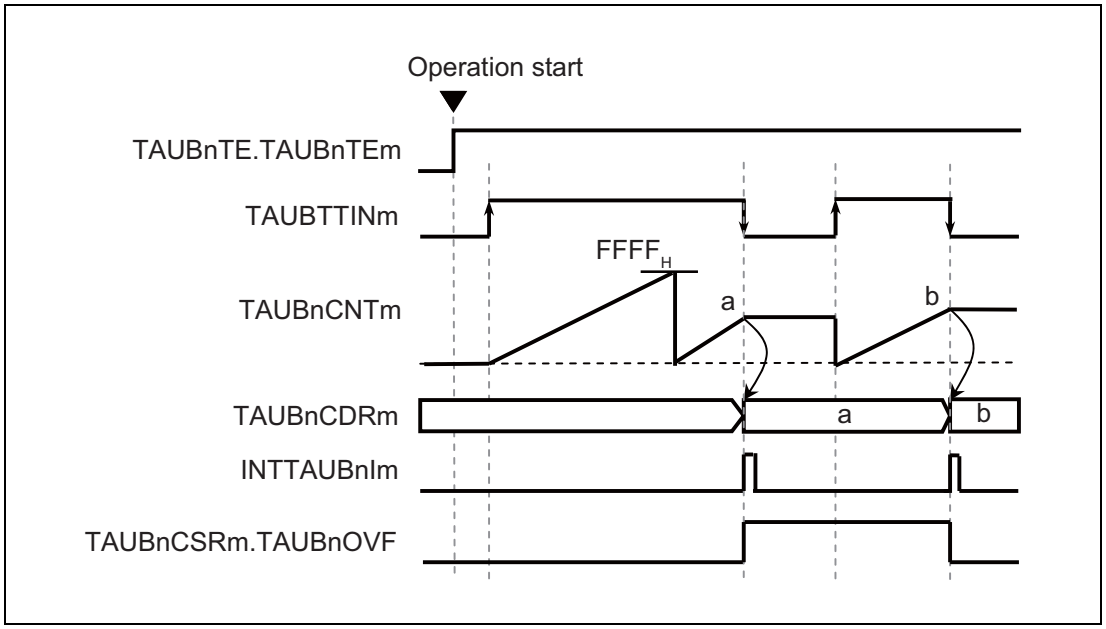


Figure 24.61 When an Overflow Occurs

When a capture trigger is input after the counter value has overflowed, the counter value is transferred to TAUBnCDRm and at the same time TAUBnCSRm.TAUBnOVF is set to 1.

TAUBnCSRm.TAUBnOVF is kept at 1 until the next capture trigger occurs.

If the next capture trigger is not accompanied by an overflow, TAUBnCSRm.TAUBnOVF is cleared to 0.

TAUBTTINm input signal width (example when TAUBnCSRm.TAUBnOVF is 1 and TAUBnCDRm is a)
= count clock cycle × ((10000_H × TAUBnCSRm.TAUBnOVF) + (TAUBnCDRm capture value + 1))
= count clock cycle × ((10000_H × 1) + (a+1))
= count clock cycle × (10000_H + a+1)

24.12.8 TAUBTTINm Input Position Detection Function

24.12.8.1 Overview

Summary

This function measures the interval of input signals by capturing the counter value on a valid edge of the TAUBTTINm signal.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling the count operation. The counter starts to count from 0000_H. When a valid TAUBTTINm input stop edge is detected, the current TAUBnCNTm value is loaded to TAUBnCDRm and an interrupt (INTTAUBnIm) is generated. The count operation continues.

When the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTE

The TAUBTTINm input signal is sampled input signal is sampled at the frequency of the operation clock, specified by the TAUBnCMORm.TAUBnCKS[1:0] bits. As a result, the output cycle of TAUBTTOUTm has an error of ± 1 operation clock cycle.

Conditions

If the TAUBnCMORm.MD0 bit is set to 0, the first interrupt after a start or restart is not generated.

24.12.8.2 Equations

Function duration at a TAUBTTINm input pulse =

count clock cycle × (TAUBnCDRm capture value + 1)

24.12.8.3 Block Diagram and General Timing Diagram

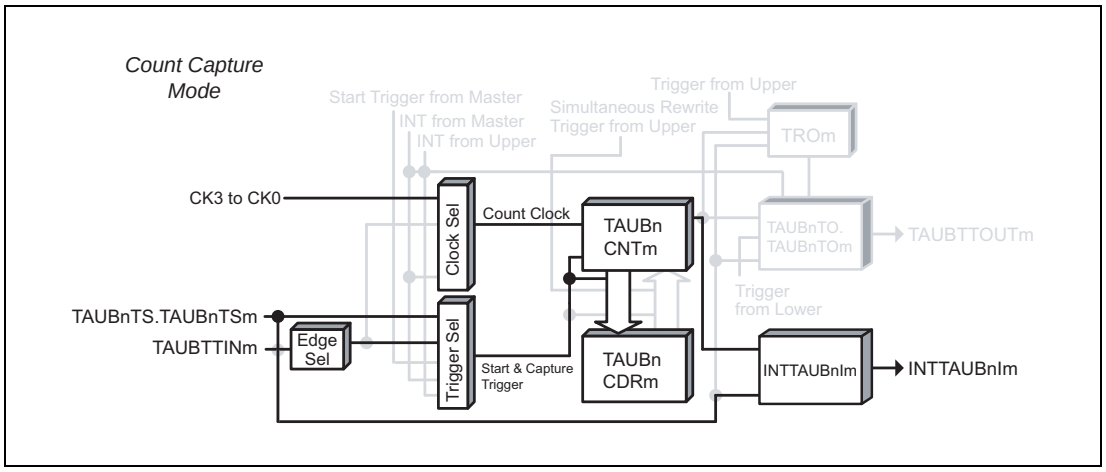


Figure 24.62 Block Diagram for TAUBTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

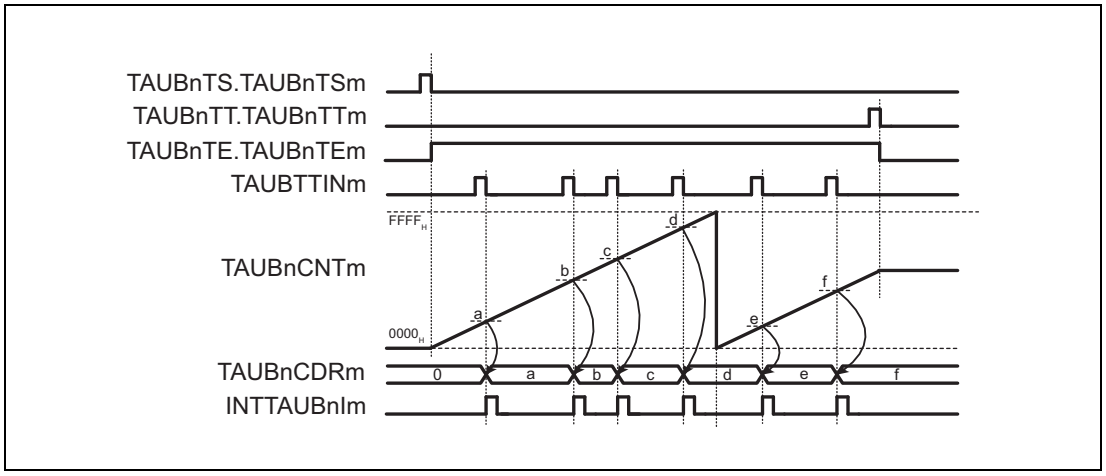


Figure 24.63 General Timing Diagram for TAUBTTINm Input Position Detection Function

24.12.8.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.70 Contents of the TAUBnCMORm Register for TAUBTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1011_B.
0	TAUBnMD0	0: INTTAUBnIm not generated at operation start 1: Generates INTTAUBnIm at operation start

(2) TAUBnCMURm

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

Table 24.71 Contents of the TAUBnCMURm Register for TAUBTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Position Detection Function. Therefore, these registers must be set to 0.

Table 24.72 Simultaneous Rewrite Settings for TAUBTTINm Input Position Detection Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.8.5 Operating Procedure for TAUBTTINm Input Position Detection Function

Table 24.73 Operating Procedure for TAUBTTINm Input Position Detection Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 24.70, Contents of the TAUBnCMORm Register for TAUBTTINm Input Position Detection Function** and **Table 24.71, Contents of the TAUBnCMURm Register for TAUBTTINm Input Position Detection Function** The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. INTTAUBnIm is generated when TAUBnCMORm.TAUBnMD0 is set to 1.
During operation	The values of TAUBnCMURm.TAUBnTIS[1:0] bits can be changed at any time. The TAUBnCDRm and TAUBnCSRm registers can be read at any time.	TAUBnCNTm starts to count up from 0000_H. When a TAUBTTINm valid edge is detected: • TAUBnCNTm transfers (captures) its value to TAUBnCDRm • INTTAUBnIm is output. • The counter value is not cleared to 0000_H and TAUBnCNTm continues count operation. Afterwards, this procedure is repeated. If TAUBnCNTm reaches $FFFF_H$, the counter restarts from 0000_H.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

24.12.8.6 Specific Timing Diagrams

(1) Operation stop and restart

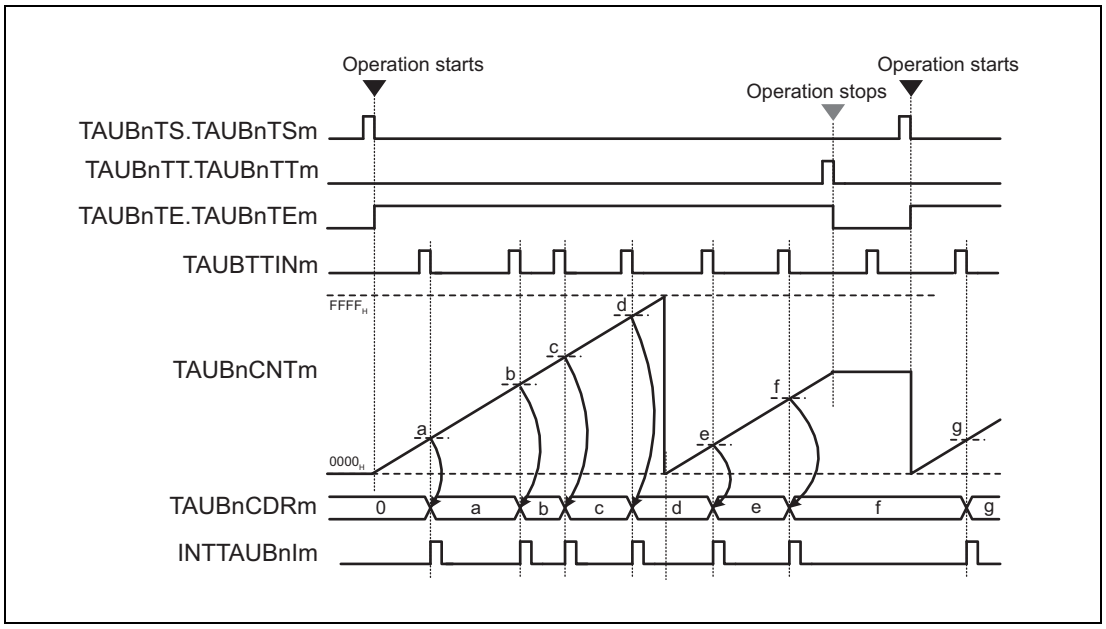


Figure 24.64 Operation Stop and Restart (TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TEm to 0.
- TAUBnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUBTTINm input edges are ignored.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm restarts to count from 0000_H.

24.12.9 TAUBTTINm Input Period Count Detection Function

24.12.9.1 Overview

Summary

This function measures the cumulative width of a TAUBTTINm input signal.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.

This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The counter awaits a valid TAUBTTINm input edge.

When a valid TAUBTTINm input start edge is detected, the counter starts to count from 0000_H.

When a valid TAUBTTINm input stop edge is detected, the current TAUBnCNTm value is loaded to TAUBnCDRm and an interrupt (INTTAUBnIm) is generated. The counter stops and retains its value (TAUBnCDRm + 1) until the next valid TAUBTTINm input start edge is detected.

When a next valid TAUBTTINm input start edge is detected, the counter restarts from the value retained while stopping.

If the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTES

1. The TAUBTTINm input signal is sampled at the frequency of the operation clock, specified by the TAUBnCMORm.TAUBnCKS[1:0] bits.
2. As this function is to measure the TAUBTTINm input signal width, setting TAUBnTS.TAUBnTSm to 1 is prohibited while TAUBnTE.TAUBnTEm = 1.

Conditions

The valid start and stop edges are specified by the TAUBnCMURm.TIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, the TAUBTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] = 11_B, the TAUBTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

24.12.9.2 Equations

Cumulative TAUBTTINm input width =

count clock cycle × (TAUBnCDRm capture value + 1)

24.12.9.3 Block Diagram and General Timing Diagram

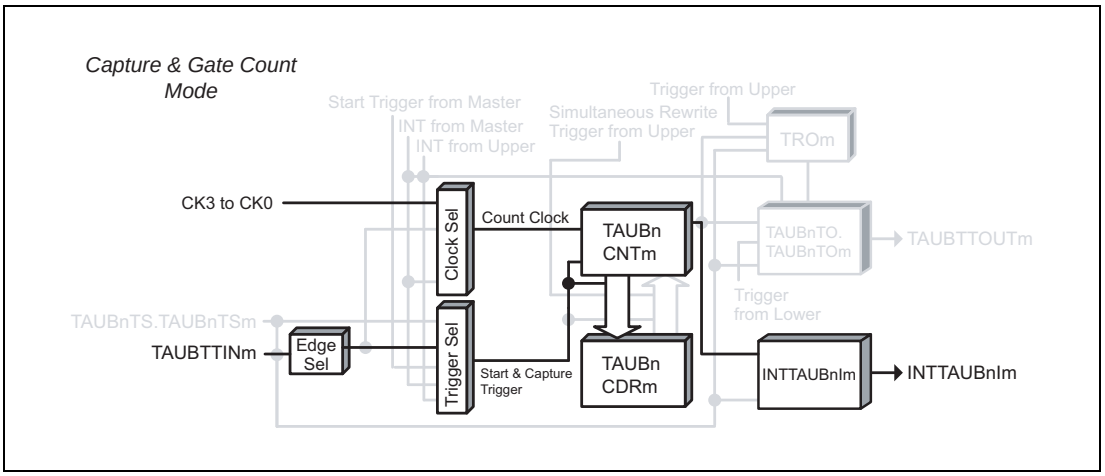


Figure 24.65 Block Diagram for TAUBTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

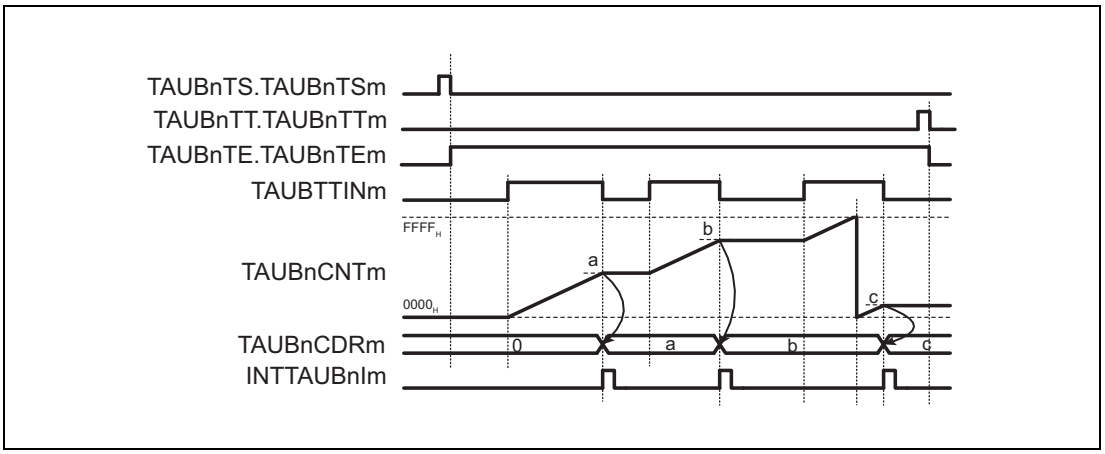


Figure 24.66 General Timing Diagram for TAUBTTINm Input Period Count Detection Function

24.12.9.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.74 Contents of the TAUBnCMORm Register for TAUBTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1101_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

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

Table 24.75 Contents of the TAUBnCMURm Register for the TAUBTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Period Count Detection Function. Therefore, these registers must be set to 0.

Table 24.76 Simultaneous Rewrite Settings for TAUBTTINm Input Period Count Detection Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.9.5 Operating Procedure for TAUBTTINm Input Period Count Detection Function

Table 24.77 Operating Procedure for TAUBTTINm Input Period Count Detection Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 24.74, Contents of the TAUBnCMORm Register for TAUBTTINm Input Period Count Detection Function** and **Table 24.75, Contents of the TAUBnCMURm Register for the TAUBTTINm Input Period Count Detection Function** The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge.
During operation	Detection of TAUBTTINm edges. The TAUBnCDRm, TAUBnCNTm, and TAUBnCSRm registers can be read at any time.	When a TAUBTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUBnCNTm starts to count up from the stop value. When TAUBnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUBnCDRm and INTTAUBnIm is generated. Counting stops at the “value transferred to TAUBnCDRm + 1” value and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When TAUBnCNTm reaches $FFFF_H$, the counter restarts from 0000_H. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

24.12.9.6 Specific Timing Diagrams

(1) Operation stop and restart

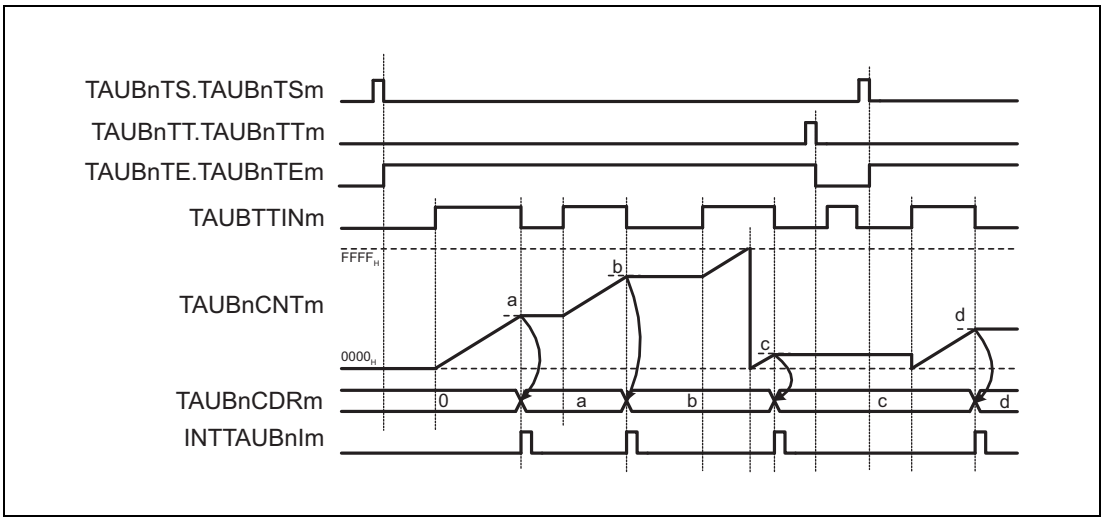


Figure 24.67 Operation Stop and Restart (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUBTTINm input edges are ignored.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm restarts to count from 0000_H.

24.12.10 TAUBTTINm Input Pulse Interval Judgment Function

24.12.10.1 Overview

Summary

This function outputs the result of a comparison between the count value (TAUBnCNTm) and the value in the channel data register (TAUBnCDRm) when a TAUBTTINm input pulse occurs. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is loaded to TAUBnCNTm and the counter starts to count down from this value.

When a TAUBTTINm valid edge is detected or TAUBnTS.TAUBnTSm is set to 1, the function compares the current values of TAUBnCNTm and TAUBnCDRm. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true. TAUBnCNTm reloads the value of TAUBnCDRm and subsequently continues operation, regardless of the result of the comparison.

If the counter reaches 0000_H before a TAUBTTINm valid edge is detected, TAUBnCNTm overflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

Conditions

The TAUBnCMORm.TAUBnMD0 bit specifies the type of comparison:

- If TAUBnCMORm.TAUBnMD0 = 0, INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm.
- If TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm.

24.12.10.2 Block Diagram and General Timing Diagram

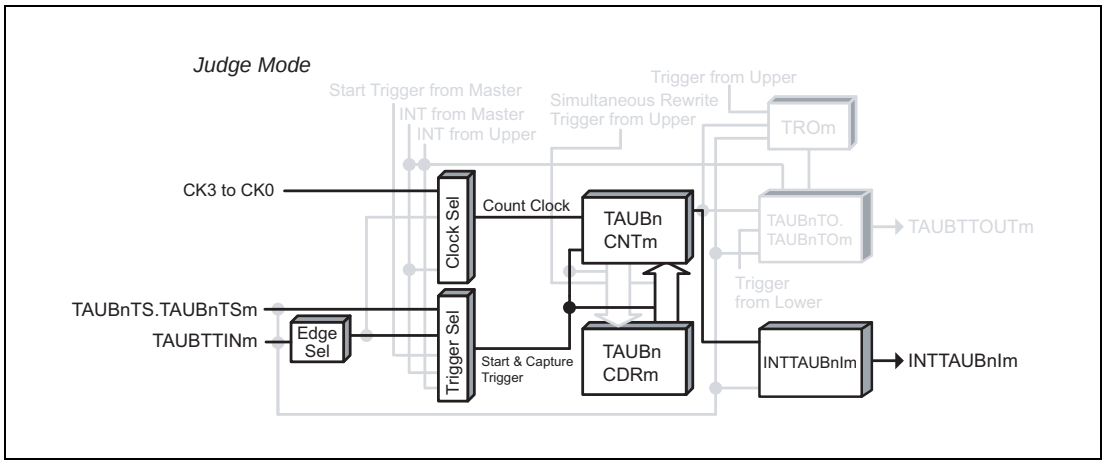


Figure 24.68 Block Diagram for TAUBTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

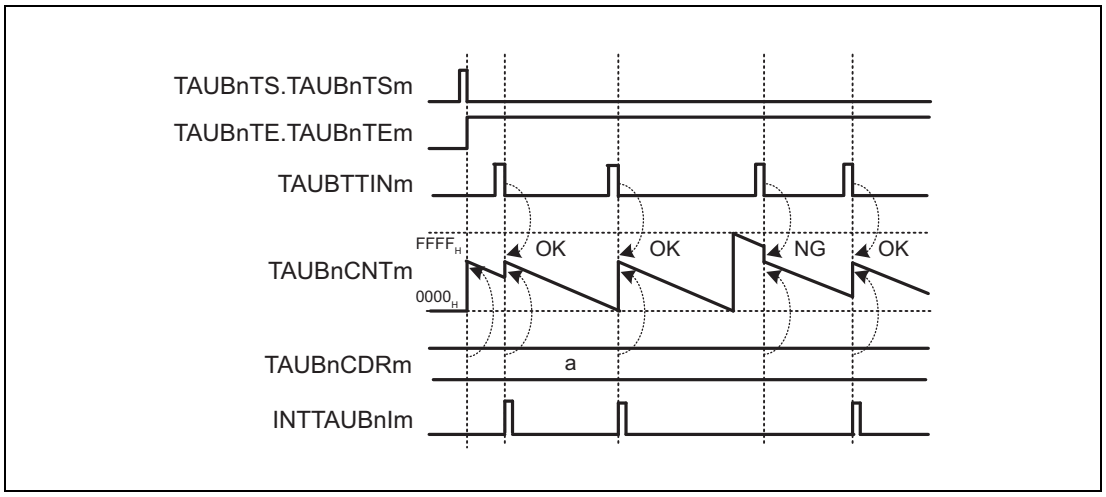


Figure 24.69 General Timing Diagram for TAUBTTINm Input Pulse Interval Judgment Function

24.12.10.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.78 Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0001_B.
0	TAUBnMD0	0: INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm 1: INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm

(2) TAUBnCMURm

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

Table 24.79 Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Pulse Interval Judgment Function. Therefore, these registers must be set to 0.

Table 24.80 Simultaneous Rewrite Settings for TAUBTTINm Input Pulse Interval Judgment Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), these bits are set to 0.
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.10.4 Operating Procedure for TAUBTTINm Input Pulse Interval Judgment Function

Table 24.81 Operating Procedure for TAUBTTINm Input Pulse Interval Judgment Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 24.78, Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Judgment Function** and **Table 24.79, Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Judgment Function** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value.
During operation	The following register can be changed at any time: • TAUBnCDRm register	When TAUBnCMORm.TAUBnMD0 = 0 If TAUBnCNTm ≤ TAUBnCDRm when a TAUBTTINm input edge is detected, INTTAUBnIm is generated. When TAUBnCMORm.TAUBnMD0 = 1 If TAUBnCNTm > TAUBnCDRm when a TAUBTTINm input edge is detected, INTTAUBnIm is generated. If a TAUBTTINm input edge is detected, then TAUBnCNTm starts to count down from the value of TAUBnCDRm. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

24.12.11 TAUBTTINm Input Signal Width Judgment Function

24.12.11.1 Overview

Summary

This function compares the count value (TAUBnCNTm) for the high or low level width of a TAUBTTINm input signal and the TAUBnCDRm value, and outputs the judgment result from the interrupt request signal INTTAUBnIm.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is started by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. When a valid TAUBTTINm input start edge is detected, the current value of TAUBnCDRm is loaded to TAUBnCNTm and the counter starts to count down from this value.

When a TAUBTTINm valid stop edge is detected, the function compares the current values of TAUBnCNTm and TAUBnCDRm. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true. The counter TAUBnCNTm retains its value until the next TAUBTTINm valid start edge is detected, regardless of the result of the comparison.

If the counter reaches 0000_H before a valid TAUBTTINm stop edge is detected, TAUBnCNTm overflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

Conditions

- The TAUBnCMORm.TAUBnMD0 bit specifies the type of comparison:
  - If TAUBnCMORm.TAUBnMD0 = 0, INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm.
  - If TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm.
- The TAUBnCMURm.TAUBnTIS[1:0] bits specify the type of width measurement:
  - For high width measurement, (TAUBnCMURm.TAUBnTIS[1:0] = 11_B) the start edge is a rising TAUBTTINm edge and the stop edge is a falling TAUBTTINm edge.
  - For low width measurement, (TAUBnCMURm.TAUBnTIS[1:0] = 10_B) the start edge is a falling TAUBTTINm edge and the stop edge is a rising TAUBTTINm edge.
- Forced restart is not possible for this function.

24.12.11.2 Block Diagram and General Timing Diagram

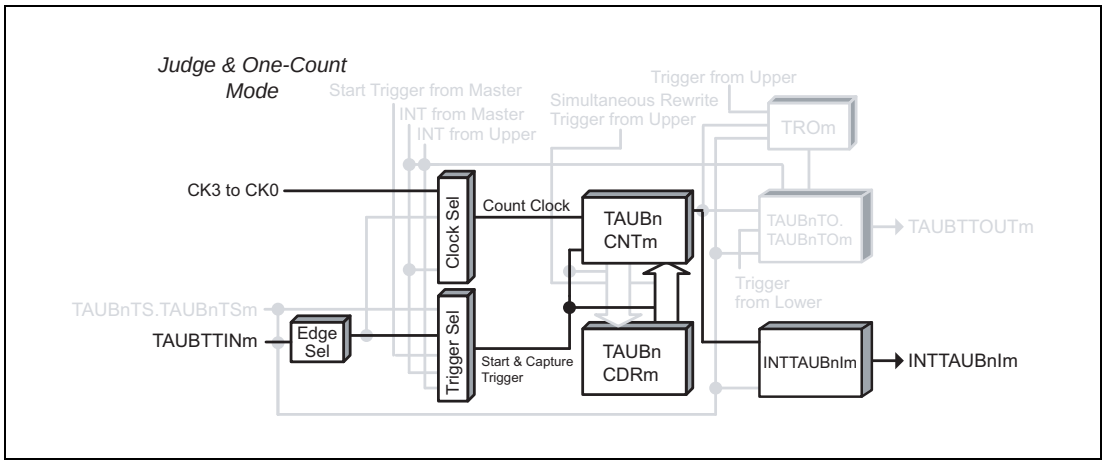


Figure 24.70 Block Diagram for TAUBTTINm Input Signal Width Judgment Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm (TAUBnCMORm.TAUBnMD0 = 0)
- TAUBTTINm valid start edge = rising edge, TAUBTTINm valid stop edge = falling edge (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

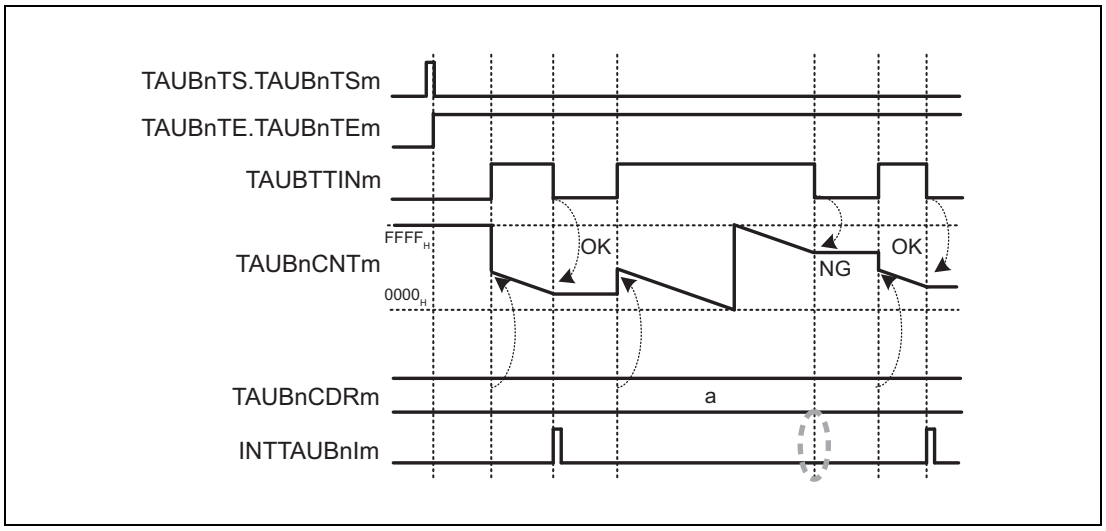


Figure 24.71 General Timing Diagram for TAUBTTINm Input Signal Width Judgment Function

24.12.11.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.82 Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0111_B.
0	TAUBnMD0	0: INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm 1: INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm

(2) TAUBnCMURm

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

Table 24.83 Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Signal Width Judgment Function. Therefore, these registers must be set to 0.

Table 24.84 Simultaneous Rewrite Settings for TAUBTTINm Input Signal Width Judgment Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), these bits are set to 0.
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.11.4 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function

Table 24.85 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 24.82, Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Judgment Function** and **Table 24.83, Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Judgment Function** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge.
During operation	The following register can be changed at any time: • TAUBnCDRm register	If a TAUBTTINm start edge is detected, then TAUBnCNTm starts to count down from the value of TAUBnCDRm. When TAUBnCMORm.TAUBnMD0 = 0 If TAUBnCNTm ≤ TAUBnCDRm when a TAUBTTINm input stop edge is detected, INTTAUBnIm is generated. When TAUBnCMORm.TAUBnMD0 = 1 If TAUBnCNTm > TAUBnCDRm when a TAUBTTINm input stop edge is detected, INTTAUBnIm is generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

24.12.12 Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

24.12.12.1 Overview

Summary

This function measures the width of an individual TAUBTTINm input signal. An interrupt is generated if the TAUBTTINm input width is longer than $FFFF_H + 1$.

Prerequisites

- TAUBTTOUTm is not used for this function
- The value of TAUBnCDRm must be set to $FFFF_H$.

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The counter starts when a valid TAUBTTINm input start edge is detected. $FFFF_H$ is loaded to TAUBnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUBTTINm input start edge is detected, TAUBnCNTm loads $FFFF_H$ and starts to count down.

If the counter reaches 0000_H before a stop edge is detected, an interrupt is generated.

Conditions

The valid start and stop edges are specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, the TAUBTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] = 11_B, the TAUBTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

24.12.12.2 Block Diagram and General Timing Diagram

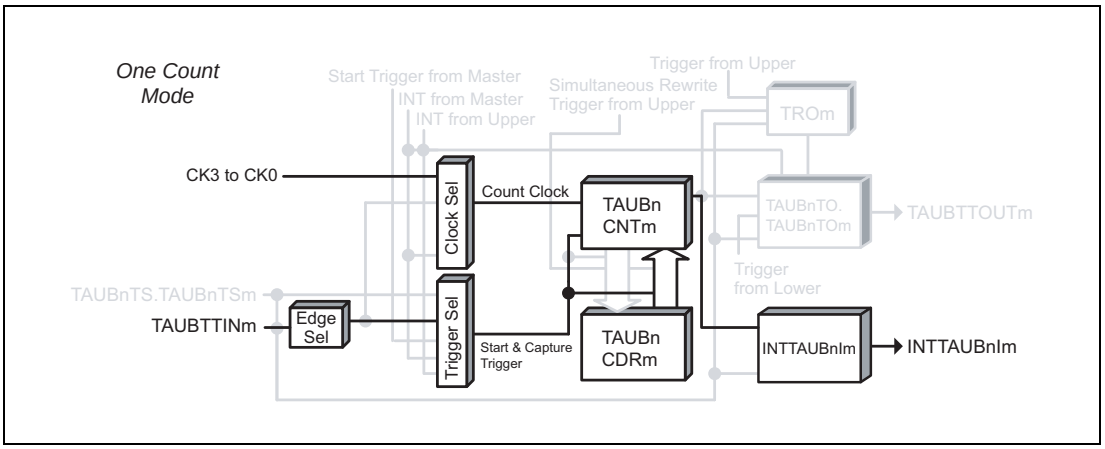


Figure 24.72 Block Diagram for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

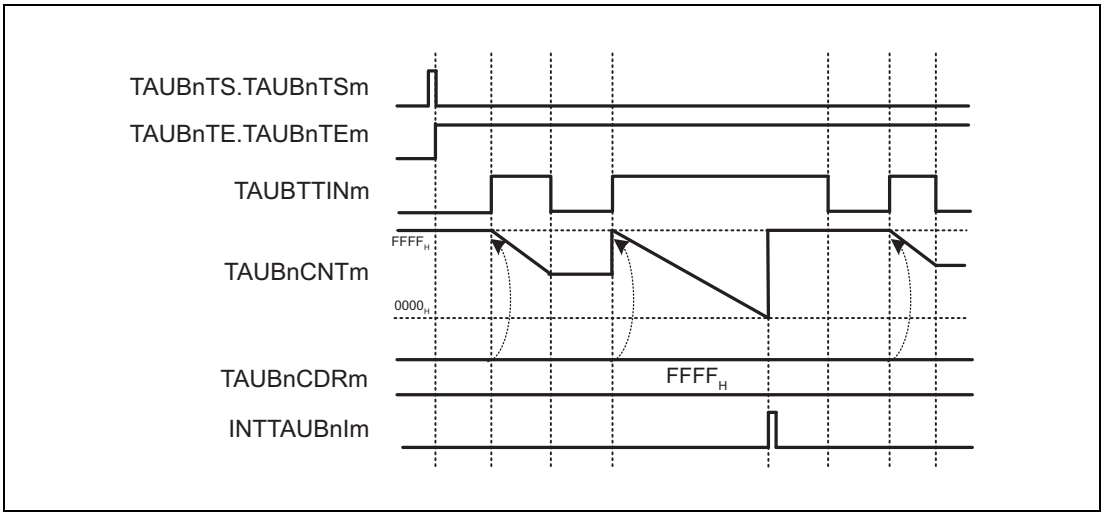


Figure 24.73 General Timing Diagram for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

24.12.12.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.86 Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 0_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

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

Table 24.87 Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUBTTINm Width Measurement). Therefore, these registers must be set to 0.

Table 24.88 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.12.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Table 24.89 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 24.86, Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)** and **Table 24.87, Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)** Set the value of the TAUBnCDRm register to $FFFF_H$.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUBTTINm start edge	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the start edge. When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value ($FFFF_H$).
During operation	The TAUBnCNTm register can be read at any time.	TAUBnCNTm counts down. When the counter reaches 0000_H: • INTTAUBnIm is generated When a TAUBTTINm input stop edge is detected during count operation: • TAUBnCNTm stops and retains its current value. When a TAUBTTINm input start edge is detected while the counter is stopped: • The TAUBnCDRm value ($FFFF_H$) is loaded to TAUBnCNTm again and the counter starts to count down. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

24.12.13 Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

24.12.13.1 Overview

Summary

This function measures the cumulative width of a TAUBTTINm input signal. An overflow interrupt can be specified to be generated if the cumulative TAUBTTINm input width is longer than $FFFF_H$.

Prerequisites

- TAUBTTOUTm is not used for this function
- The value of TAUBnCDRm must be set to $FFFF_H$

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The counter starts when a valid TAUBTTINm input start edge is detected. $FFFF_H$ is loaded to TAUBnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUBTTINm input start edge and then continues to count down from the current value.

When the counter reaches 0000_H an interrupt is generated. $FFFF_H$ is loaded to TAUBnCNTm and the counter continues to count down until a TAUBTTINm input stop edge is detected.

Conditions

The valid start and stop edges are specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, the TAUBTTINm input low width is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] = 11_B, the TAUBTTINm input high width is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

24.12.13.2 Block Diagram and General Timing Diagram

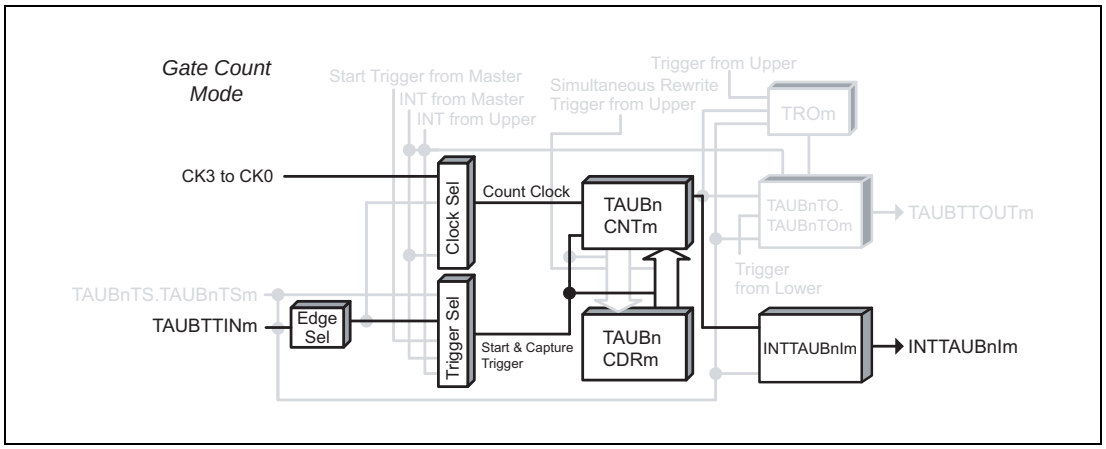


Figure 24.74 Block Diagram for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

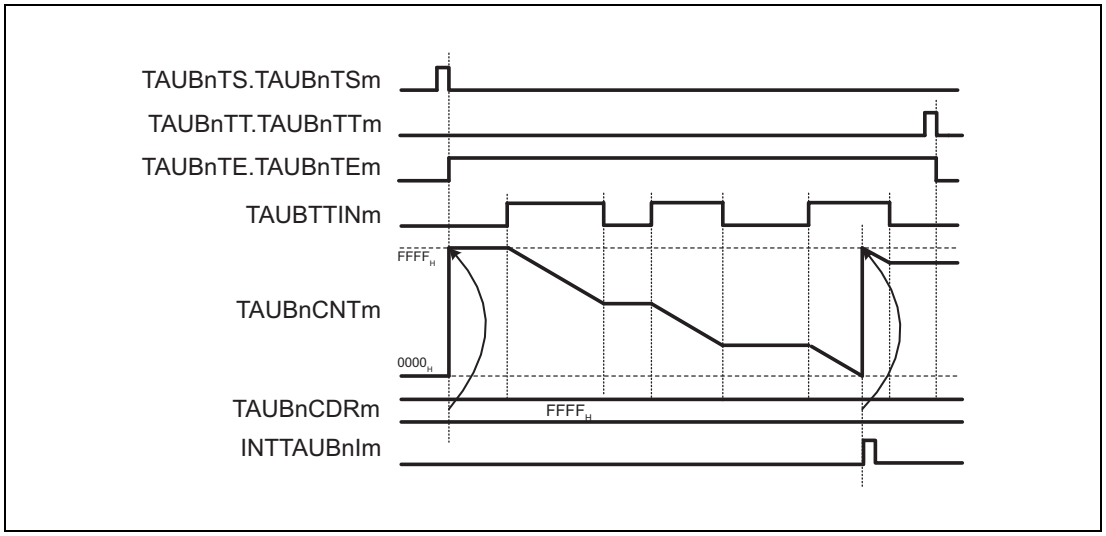


Figure 24.75 General Timing Diagram for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

24.12.13.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.90 Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1100_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

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

Table 24.91 Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 24.92 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

24.12.13.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Table 24.93 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers as described in **Table 24.90, Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)** and **Table 24.91, Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)** Set the value of the TAUBnCDRm register to $FFFF_H$.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1 TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUBTTINm start edge	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the start edge. When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value ($FFFF_H$).
During operation	The TAUBnCNTm register can be read at all times.	TAUBnCNTm counts down. When the counter reaches 0000_H: • INTTAUBnIm is generated • TAUBnCNTm loads the TAUBnCDRm value ($FFFF_H$) and continues count operation When a TAUBTTINm input stop edge is detected during count operation: • TAUBnCNTm stops and retains its current value. When a TAUBTTINm input start edge is detected while the counter is stopped: • TAUBnCNTm starts to count down from the stop value. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

24.13 Independent Channel Simultaneous Rewrite Functions

The following describes functions that carry out simultaneous rewrite:

24.13.1 Simultaneous Rewrite Trigger Generation Function Type 1

24.13.1.1 Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is generated at regular intervals. The upper channel is for generating the simultaneous rewrite trigger (TAUBnRDC.TAUBnRDCm = 1), and the lower channels are for conducting simultaneous rewrite when triggered from the upper channel (TAUBnRDC.TAUBnRDCm = 0).

Prerequisites

- Two (or more) channels that are lower than the channel used as the upper channel, each with simultaneous rewrite enabled (TAUBnRDE.TAUBnRDEm = 1)
- The operation mode of the upper channel must be set to interval timer mode, see **Table 24.94, Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1**
- For the operation modes that can be set to the lower channels, see **Table 24.37, Channel Functions and the Methods They Use for Simultaneous Rewrite**
- In this function, TAUBTTOUTm is not used for all the channels.

Description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) of the upper and lower channel(s) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of the data register buffer of the upper channel (TAUBnCDRm buf) is loaded to the counter (TAUBnCNTm) and the counter starts to count down from this value.
The counter(s) of the lower channel(s) start to count as specified by their selected operating modes.

When a counter reaches 0000_H, an interrupt is generated from the channel.
The corresponding TAUBnCNTm then loads the current TAUBnCDRm buffer value and subsequently continues operation.

If the channel where the interrupt occurs is specified as the trigger channel for simultaneous rewrite (TAUBnRDC.TAUBnRDCm = 1) and is an upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUBnRSF.TAUBnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers. Each time a counter starts to count down, it reads the value in the data register buffer and counts down from this value.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Condition

- The channel which is monitored for INTTAUBnIm is specified by setting TAUBnRDC.TAUBnRDCm = 1 for the corresponding channel. The TAUBnRDC.TAUBnRDCm bit must be 0 for all other channels in which simultaneous rewrite should take place.

24.13.1.2 Equations

Simultaneous rewrite trigger generation cycle = count clock cycle × (TAUBnCDRm + 1)

To control simultaneous rewrite, the following condition must be satisfied:

[For PWM]

TAUBnCDRm = [(value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1) × number of interrupts] – 1

[For triangle PWM]

TAUBnCDRm = [(value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1) × 2 × number of interrupts] – 1

That is, the ratio of TAUBnCDRm + 1 and value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1 must be an integer. This integer corresponds to the number of interrupts.

Note that the cycle for the triangular wave PWM is twice the cycle for the PWM

24.13.1.3 Block Diagram and General Timing Diagram

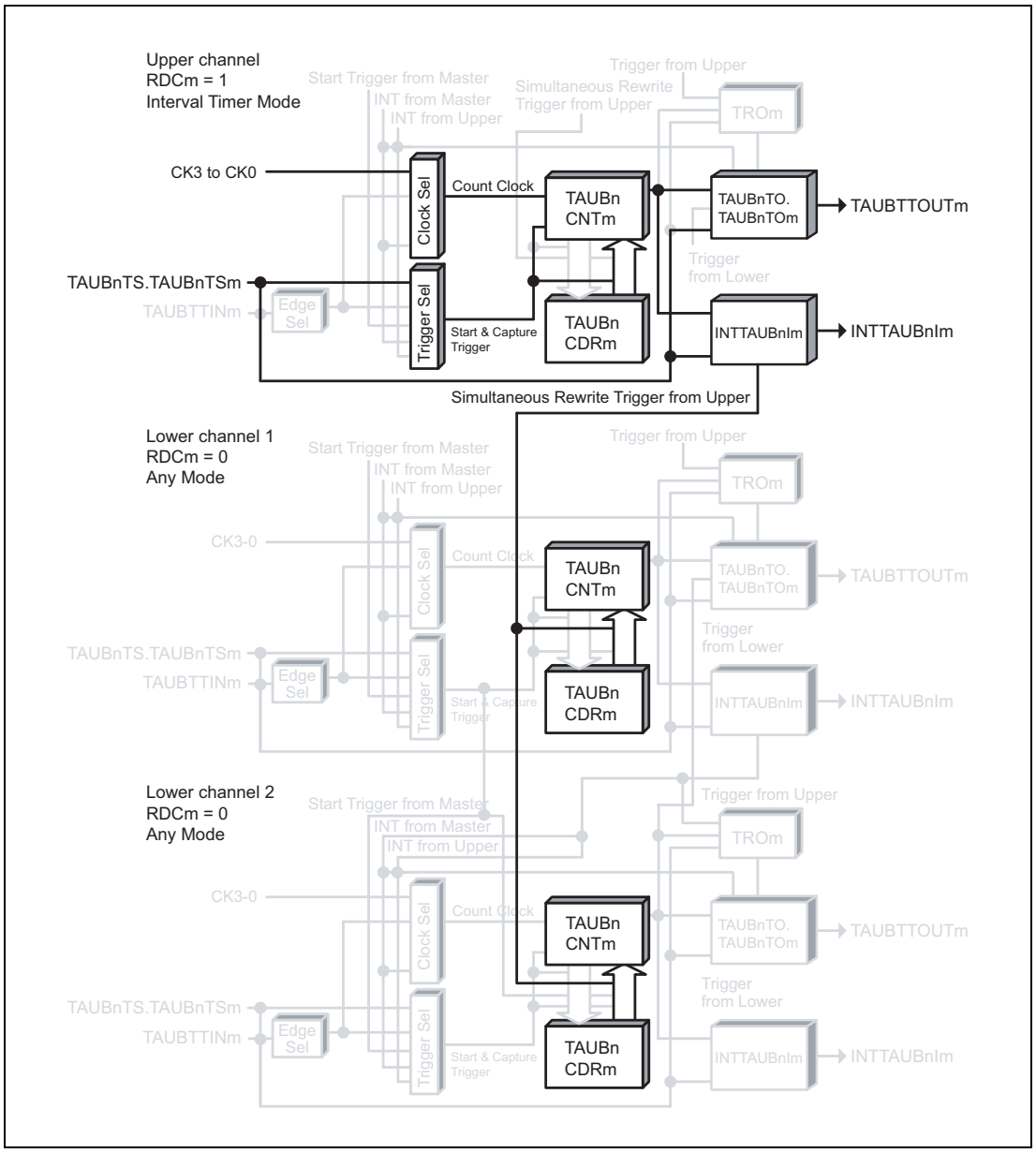


Figure 24.76 Block Diagram for Simultaneous Rewrite Trigger Generation Function Type 1

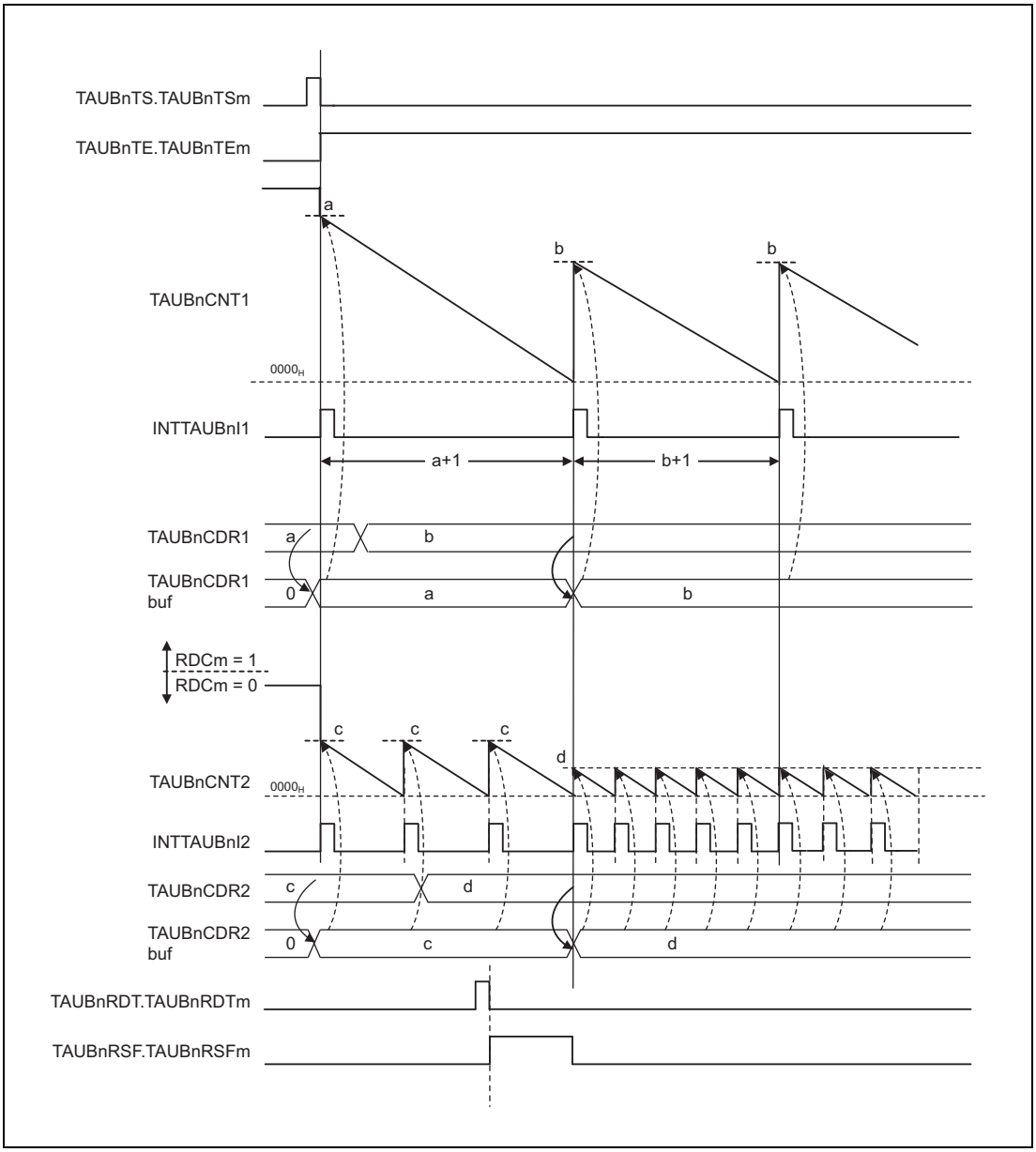


Figure 24.77 General Timing Diagram for Simultaneous Rewrite Trigger Generation Function Type 1

24.13.1.4 Register Settings for The Upper Channel

(1) TAUBnCMORm for the upper channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.94 Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the upper channel

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

Table 24.95 Contents of the TAUBnCMURm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the upper channel

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function. However, it can be used in independent channel output mode controlled by software.

(4) Simultaneous rewrite for the upper channel

Table 24.96 Simultaneous Rewrite Settings for Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUBnRDC.TAUBnRDCm	1: Channel is monitored for an INTTAUBnIm signal that is used as the simultaneous rewrite trigger

24.13.1.5 Register Settings for the Lower Channel(s)

(1) TAUBnCMORm for the lower channel(s)

For the TAUBnCMORm register of the lower channels, follow the TAUBnCMORm register settings for the operation mode that can be set. (See **Table 24.37, Channel Functions and the Methods They Use for Simultaneous Rewrite**)

(2) TAUBnCMURm for the lower channel(s)

For the TAUBnCMURm register of the lower channels, follow the TAUBnCMURm register settings for the operation mode that can be set. (See **Table 24.37, Channel Functions and the Methods They Use for Simultaneous Rewrite**)

(3) Channel output mode for the lower channel(s)

Output can be made according to the setting for lower channels (master/slave). As for the available function for simultaneous rewrite trigger generation function type 1, see **Table 24.37, Channel Functions and the Methods They Use for Simultaneous Rewrite**.

(4) Simultaneous rewrite for the lower channel(s)

Table 24.97 Simultaneous Rewrite Settings for the Lower Channel in Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

24.13.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Table 24.98 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm and TAUBnCMURm registers for the upper channel as described in **Table 24.94, Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1** and **Table 24.95, Contents of the TAUBnCMURm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1** Set the TAUBnCMORm and TAUBnCMURm registers for the lower channel as described in **Section 24.13.1.5, Register Settings for the Lower Channel(s)** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated.
During operation	TAUBnRDT.TAUBnRDTm, TAUBnCDR.TAUBnCDRm can be changed. TAUBnRSF.TAUBnRSFm can be read at all times.	TAUBnCNTm counts down. When the counter reaches 0000_H: • TAUBnCNTm reloads the TAUBnCDRm value and continues count operation • INTTAUBnIm is generated Simultaneous rewrite is controlled when INTTAUBnIm is generated from the channel where TAUBnRDC.TAUBnRDCm is set to 1. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

24.14 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the Timer Array Unit B. For a general overview of synchronous channel operation, see **Section 24.2, Overview**.

24.14.1 PWM Output Function

24.14.1.1 Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the pulse width (duration) of the TAUBTTOUTm to be set. The pulse cycle is set in the master channel. The pulse width is set in the slave channel.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode, see **Table 24.99, Contents of the TAUBnCMORm Register for the Master Channel of the PWM Output Function**
- The operation mode of the slave channel(s) must be set to one-count mode, see **Table 24.102, Contents of the TAUBnCMORm Register for the Slave Channel of the PWM Output Function**
- TAUBTTOUTm is not used for the master channel of this function
- The channel output mode of the slave channel(s) must be set to synchronous channel output mode 1.

Description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is loaded to TAUBnCNTm and the counters start to count down from these values. INTTAUBnIm is generated on the master channel and TAUBTTOUTm (slave) is set and reset, which realizes a PWM output.

- Master channel:

  When the counter of the master channel reaches 0000_H, indicating that the pulse cycle time has elapsed, INTTAUBnIm is generated. The counter loads the TAUBnCDRm value to TAUBnCNTm and counts down.

- Slave channel:

  The INTTAUBnIm of the master channel triggers the counter of the slave channel(s). The current value of TAUBnCDRm (slave) is loaded to TAUBnCNTm (slave) and the counter starts to count down from this value. The TAUBTTOUTm signal is set to the active level.

  When the counter reaches 0000_H, i.e. duty time has elapsed, INTTAUBnIm is generated and the TAUBTTOUTm signal is reset to the inactive level. The counter returns to $FFFF_H$ and awaits the next INTTAUBnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. Please see **Section 24.6, Simultaneous Rewrite**.

24.14.1.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUBnCDRm (slave) / (TAUBnCDRm (master) + 1)) × 100

– Duty cycle = 0%
  TAUBnCDRm (slave) = 0000_H
– Duty cycle = 100%
  TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1

24.14.1.3 Block Diagram and General Timing Diagram

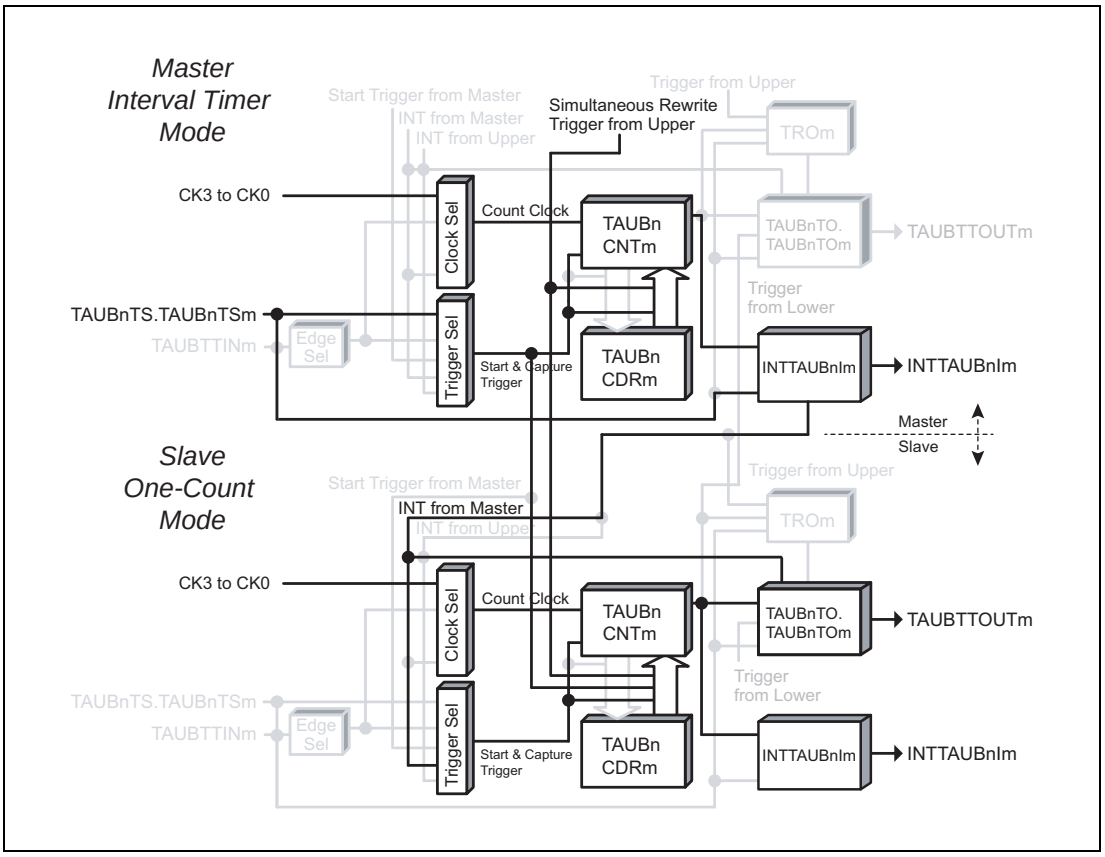


Figure 24.78 Block Diagram for PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUBnTOL.TAUBnTOLm = 0)

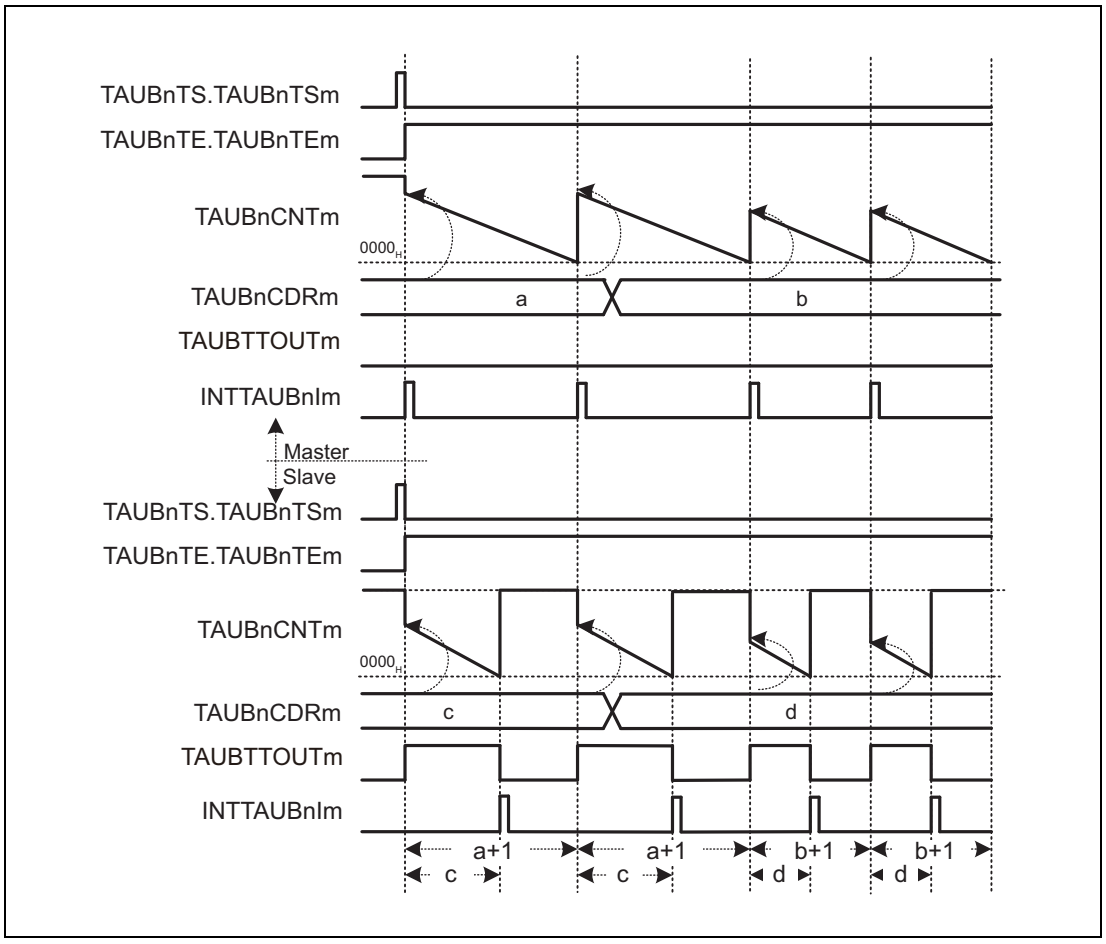


Figure 24.79 General Timing Diagram for PWM Output Function

NOTE

The interval between the starting to count and an interrupt being generated is the value of corresponding TAUBnCDRm + 1.

24.14.1.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.99 Contents of the TAUBnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the master channel

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

Table 24.100 Contents of the TAUBnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

The channel output mode is not used by this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.101 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

NOTE

When used in TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel operating in **Section 24.13.1, Simultaneous Rewrite Trigger Generation Function Type 1**.

Configure the operation following the conditions below.

- The channel set to Simultaneous Rewrite Trigger Output Function Type 1: TAUBnRDCm = 1, TAUBnRDSm = 1
  The setting value of TAUBnCDRm to this channel is as follows.
  = ((setting value of TAUBnCDRm of the master channel subject to simultaneous rewrite + 1) × number of interrupts) – 1
- Master channel: TAUBnRDCm = 0, TAUBnRDSm = 1
- Slave channel: TAUBnRDCm = 0, TAUBnRDSm = 1

Although the value of duty exceeds 100% when the setting value of TAUBnCDRm (slave) > the setting value of TAUBnCDRm (master) + 1, the output will be aggregated to 100%.

24.14.1.5 Register Settings for the Slave Channel(s)

(1) TAUBnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.102 Contents of the TAUBnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 100_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the slave channel(s)

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

Table 24.103 Contents of the TAUBnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the slave channel(s)

Table 24.104 Control Bit Settings for Synchronous Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.105 Simultaneous Rewrite Settings for the Slave Channel of the PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

24.14.1.6 Operating Procedure for PWM Output Function

Table 24.106 Operating Procedure for PWM Output Function

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.1.4, Register Settings for the Master Channel**. Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.1.5, Register Settings for the Slave Channel(s)**. Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnIm is generated on the master channel and TAUBTTOUTm (slave) is set.
During operation	TAUBnCDRm can be changed at any time. TAUBnTOL.TAUBnTOLm can be changed. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master channel loads TAUBnCDRm and counts down. When the counter reaches 0000_H: • INTTAUBnIm (master) is generated • TAUBnCNTm (master) loads the TAUBnCDRm value and continues count operation • TAUBnCNTm (slave) loads the TAUBnCDRm value and counts down • TAUBTTOUTm (slave) is is set to the active level When TAUBnCNTm (slave) reaches 0000_H: • The counter of TAUBnCNTm (slave) stops. • INTTAUBnIm (slave) is generated • TAUBTTOUTm (slave) is is set to the inactive level
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

24.14.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

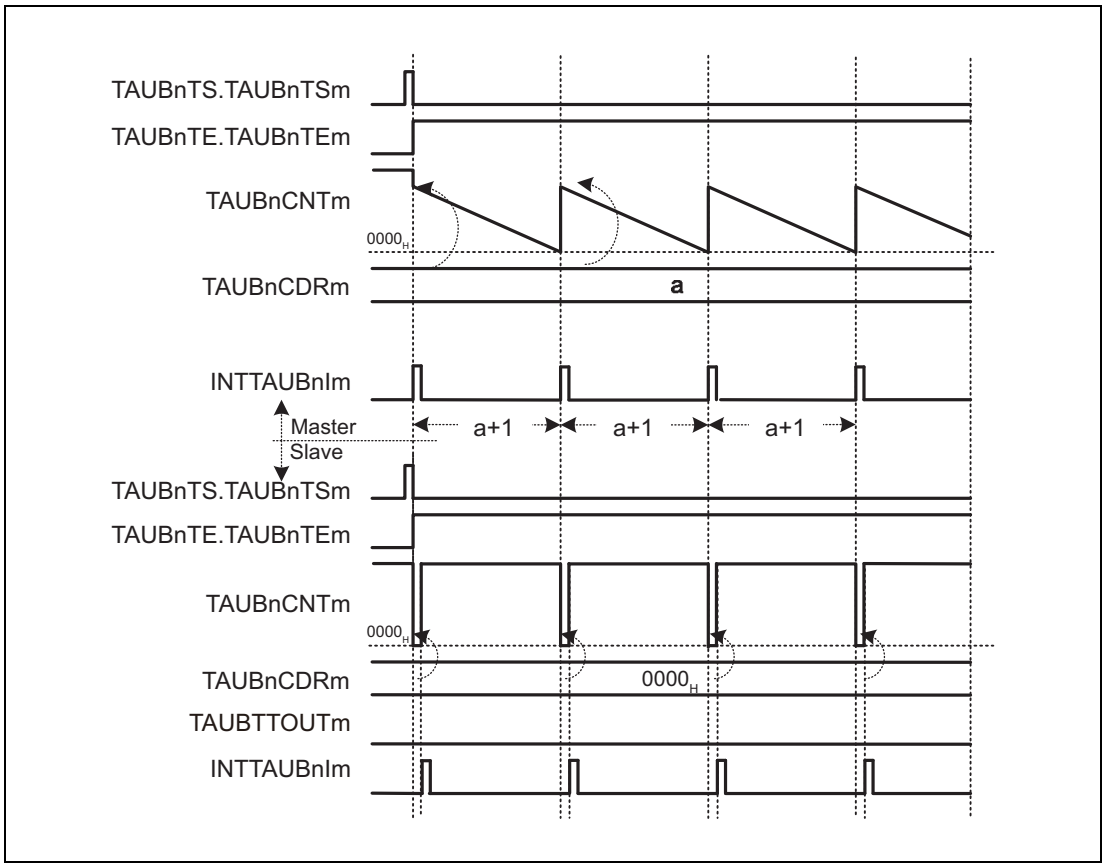


Figure 24.80 TAUBnCDRm (slave) = 0000_H, Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

- Every time the master channel generates an interrupt (INTTAUBnIm), 0000_H is loaded to TAUBnCNTm (slave). As a result, a slave channel interrupt (INTTAUBnIm) is generated at the same time and TAUBTTOUTm remains inactive.
- TAUBnCNTm (slave) generates an interrupt every time the value of TAUBnCDRm is loaded.

(2) Duty cycle = 100%

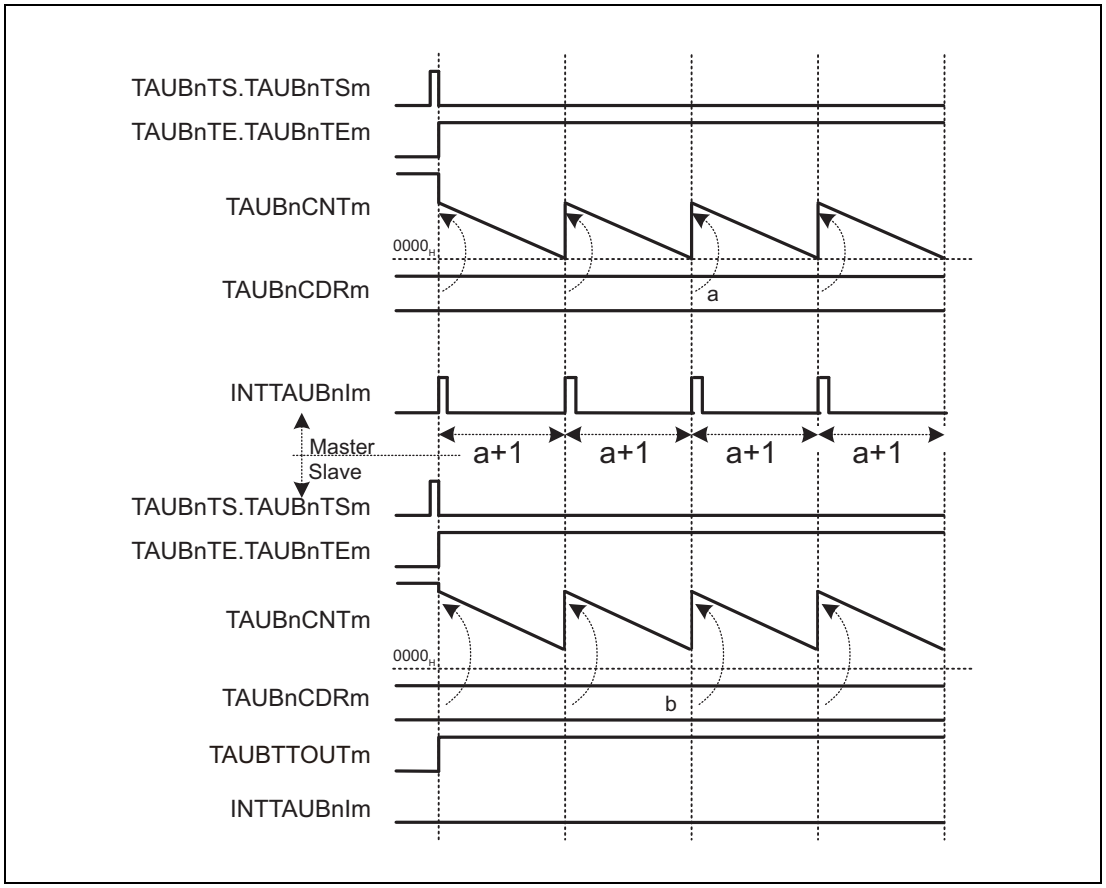


Figure 24.81 TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1, Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

If the value TAUBnCDRm (slave) is higher than the value TAUBnCDRm (master), the counter of the slave channel cannot reach 0000_H and cannot generate interrupts. The TAUBTTOUTm remains at active state.

(3) Operation Stop and Restart

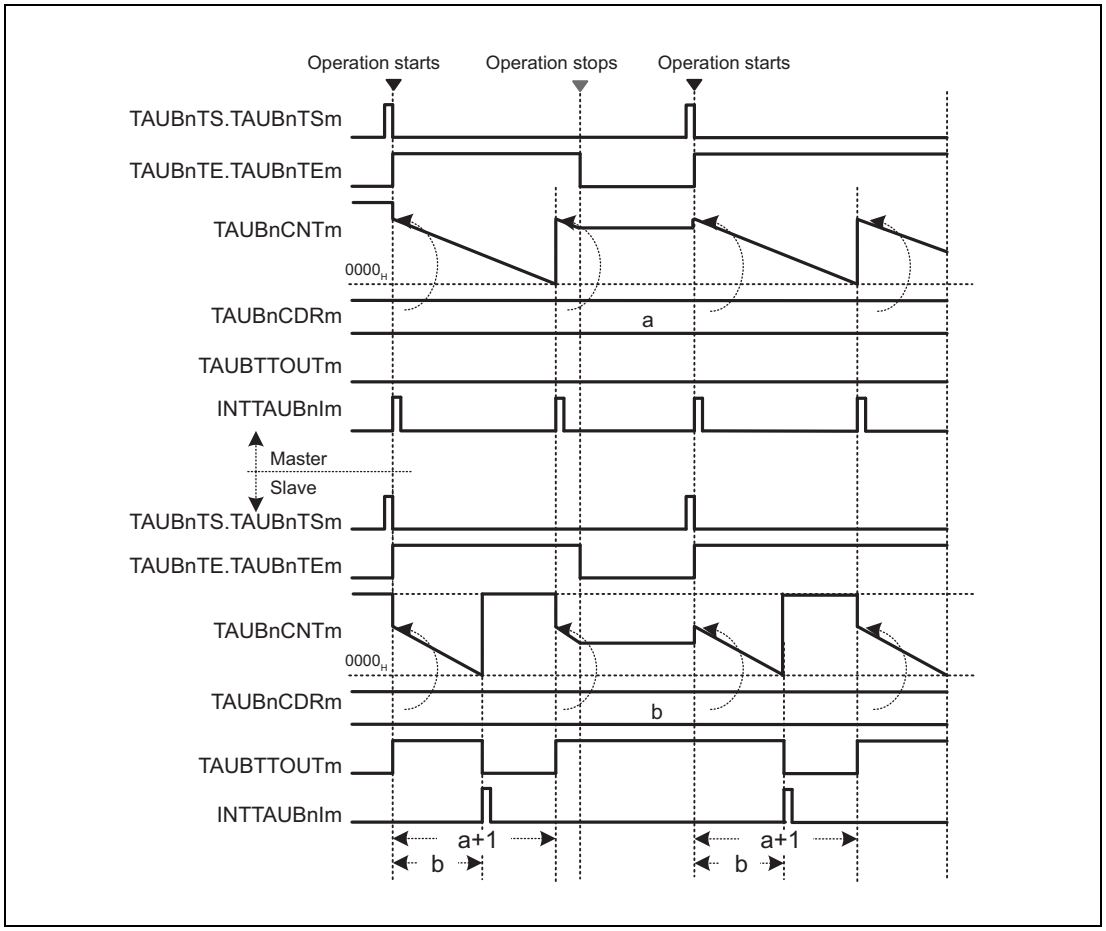


Figure 24.82 Operation Stop and Restart, Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm of the master and slave channel(s) to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm and TAUBTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm of master and slave channel(s) to 1. TAUBnCNTm of master and slave channel reload the current values of TAUBnCDRm and start to count down from these values.

(4) Operation Stop and Restart (Slave output, Initialization)

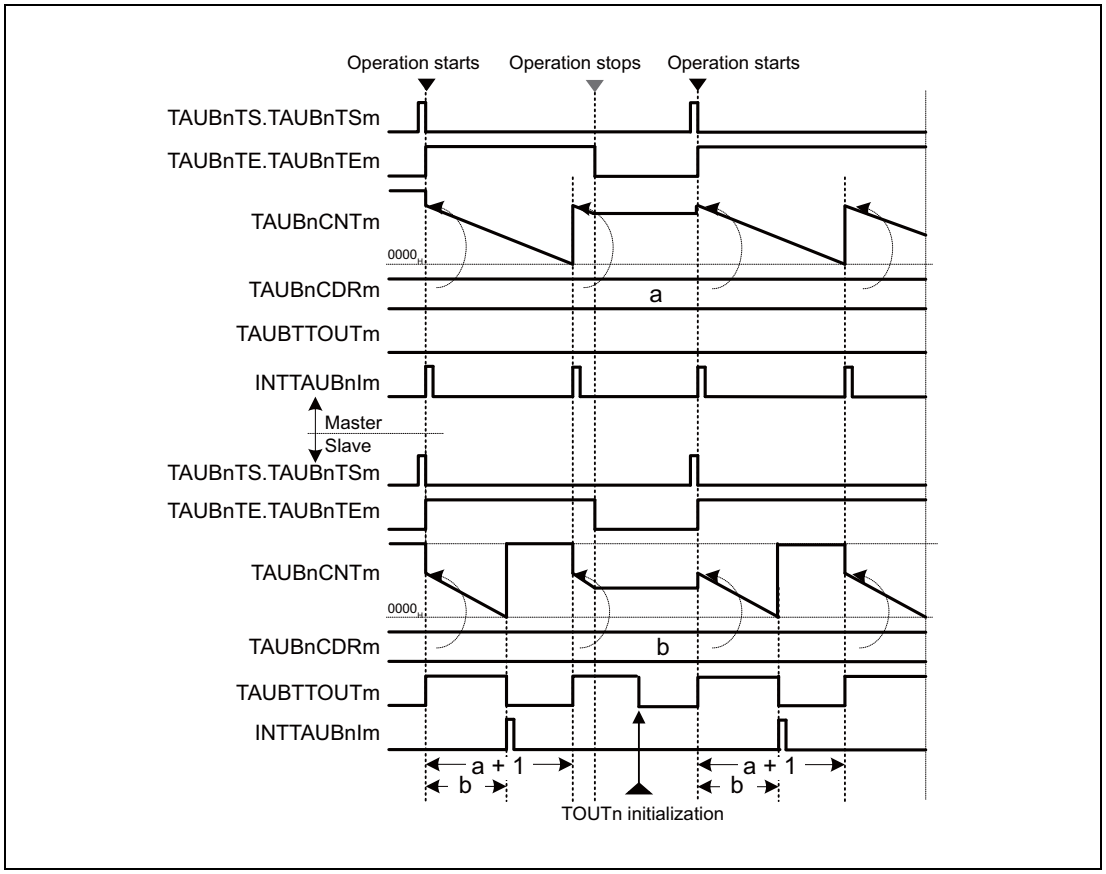


Figure 24.83 Operation Stop and Restart (Slave Output, Initialization)

When TAUBnTOE.TAUBnTOEm of the slave channel is set to 0 while TAUBnTE.TAUBnTEm = 0 and the inactive level of TAUBTTOUTm is written in the TAUBnTO.TAUBnTOm, the output level of TAUBTTOUTm (slave channel) becomes active when INTTAUBnIm is issued after the count operation is started.

24.14.2 One-Shot Pulse Output Function

24.14.2.1 Overview

Summary

This function outputs a signal pulse with a defined pulse width and a specific delay time compared to an external input signal pulse by using a master and a slave channel. The delay time is specified using the master channel. The pulse width is specified using the slave channel.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to one-count mode, see **Table 24.107, Contents of the TAUBnCMORm Register for the Master Channel of the One-Shot Pulse Output Function**
- The operation mode of the slave channel must be set to pulse one-count mode, see **Table 24.110, Contents of the TAUBnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function**
- TAUBTTOUTm is not used for the master channel of this function
- The channel output mode of the slave channel must be set to independent channel output mode 2.
- TAUBTTINm (master) has to be detected while TAUBnCNTm (master) and TAUBnCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUBTTINm (slave).

Description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) for the master and slave channels to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

- Master channel:
  When the next valid TAUBTTINm input edge is detected, the current value of TAUBnCDRm is loaded to TAUBnCNTm. The counter starts to count down from this value. If TAUBnCMORm.TAUBnMD0 = 0, a trigger (TAUBTTINm) which is detected within the delay time is ignored.
  When the counter of the master channel reaches 0000_H, INTTAUBnIm is generated. The counter is reset to $FFFF_H$ and awaits the next valid TAUBTTINm input edge.
- Slave channel:
  The INTTAUBnIm of the master channel triggers the counter of the slave channel. The current value of TAUBnCDRm (slave) is loaded to TAUBnCNTm (slave) and the counter starts to count down from this value.
  An interrupt is generated and the TAUBTTOUTm signal is set.
  When the counter reaches 0001_H, INTTAUBnIm is generated and the TAUBTTOUTm signal is reset. The counter stops at 0000_H and awaits the next INTTAUBnIm of the master channel.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channels, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

The counter of the master channel can be restarted without stopping it first (forced restart) by setting TAUBnTS.TAUBnTSm to 1 during operation.

NOTES

1. If a forced restart of the counter is executed during operation, the width of the output signal does not correspond to the value of TAUBnCDRm (slave).
2. The TAUBTTINm input signal is sampled at the frequency of the operating clock, specified by TAUBnCMORm.TAUBnCKS[1:0] bits. As a result, the output cycle of TAUBTTOUTm has an error of ± 1 operation clock cycle.

Conditions

- If TAUBnCMORm.TAUBnMD0 of the master channel is set to 0, during counting detected TAUBTTINm input edges are ignored.
- Simultaneous rewrite can be used with this function. Please see **Section 24.6, Simultaneous Rewrite**.

24.14.2.2 Equations

Delay from trigger input to pulse output
= (TAUBnCDRm (master) + 1) × count clock cycle

Pulse width = (TAUBnCDRm (slave)) × count clock cycle

24.14.2.3 Block Diagram and General Timing Diagram

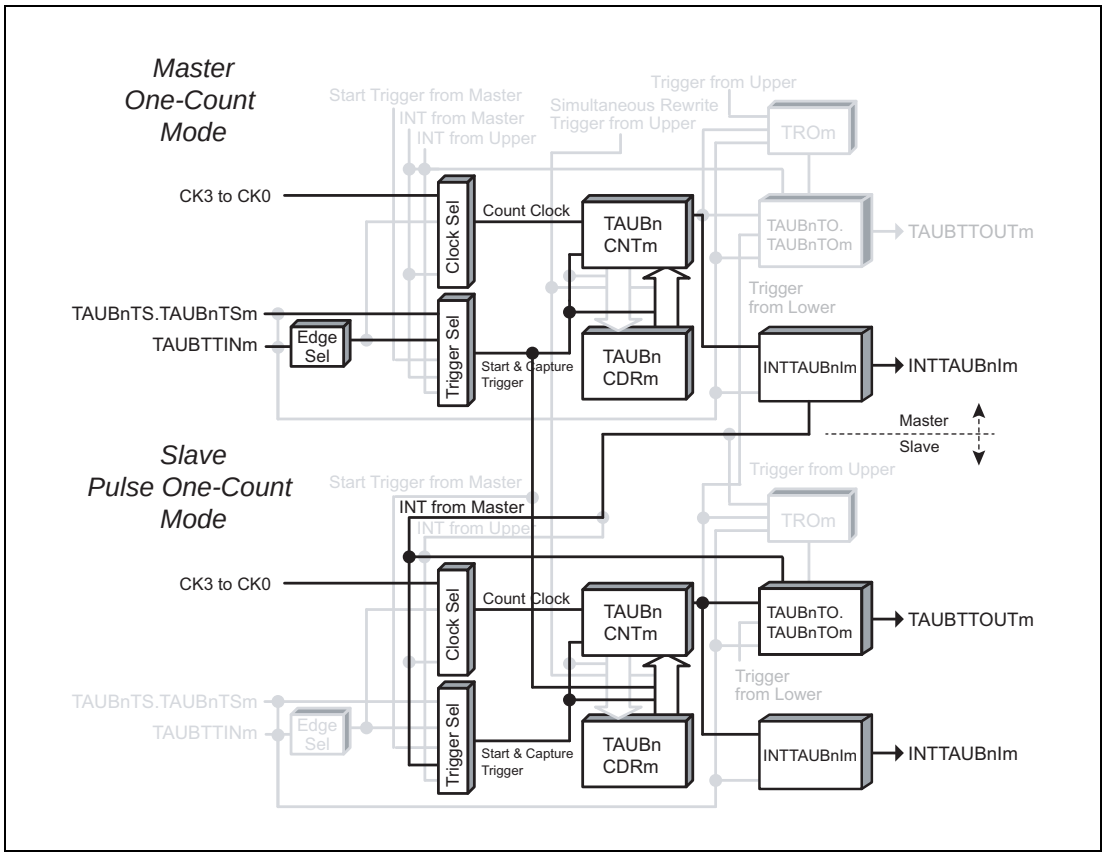


Figure 24.84 Block Diagram for One-Shot Pulse Output Function

The following settings apply to the general basic diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

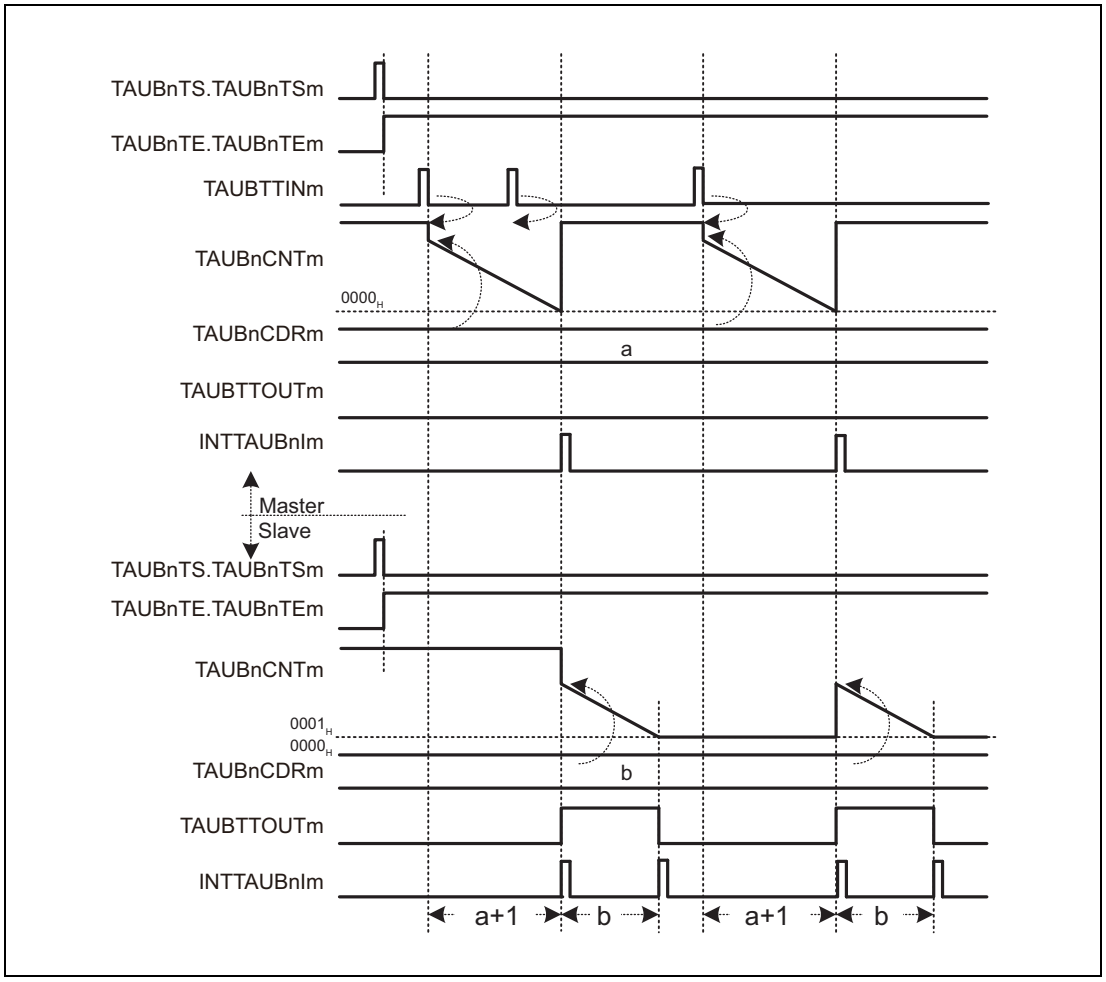


Figure 24.85 General Timing Diagram for One-Shot Pulse Output Function

24.14.2.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.107 Contents of the TAUBnCMORm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	0: Disables start trigger detection during counting 1: Enables start trigger detection during counting

(2) TAUBnCMURm for the master channel

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

Table 24.108 Contents of the TAUBnCMURm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode for the master channel

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.109 Simultaneous Rewrite Settings for the Master Channel of the One-Shot Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

24.14.2.5 Register Settings for the Slave Channel

(1) TAUBnCMORm for the slave channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.110 Contents of the TAUBnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 100_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010_B.
0	TAUBnMD0	0: Disables start trigger detection during counting 1: Enables start trigger detection during counting The value of the MD0 bit of the master and slave channels must be identical.

(2) TAUBnCMURm for the slave channel

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

Table 24.111 Contents of the TAUBnCMURm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the slave channel

Table 24.112 Control Bit Settings for Independent Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	1: Enables independent channel output mode
TAUBnTOM.TAUBnTOMm	0: Independent channel output
TAUBnTOC.TAUBnTOCm	1: Operating mode 2
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	0: Disables dead time operation
TAUBnTDL.TAUBnTDLm	0: When dead time operation is disabled (TAUBnTDE.TAUBnTDEm = 0), set these bits to 0

(4) Simultaneous rewrite for the slave channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.113 Simultaneous Rewrite Settings for the Slave Channel of the One-Shot Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

24.14.2.6 Operating Procedure for One-Shot Pulse Output Function

Table 24.114 Operating Procedure for One-Shot Pulse Output Function

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.2.4, Register Settings for the Master Channel**. Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.2.5, Register Settings for the Slave Channel**. Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the master channel awaits a TAUBTTINm input.
During operation	TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	When a valid TAUBTTINm input edge is detected, TAUBnCNTm of the master channel loads the TAUBnCDRm value and counts down. When the counter reaches 0000_H: • INTTAUBnIm (master) is generated • TAUBnCNTm (master) is reset to $FFFF_H$ and waits for the next valid TAUBTTINm input edge. • TAUBnCNTm (slave) loads the TAUBnCDRm value and starts to count down • INTTAUBnIm (slave) is generated • TAUBTTOUTm (slave) is set to the active level. When TAUBnCNTm (slave) reaches 0001_H: • The counter of TAUBnCNTm (slave) stops. • INTTAUBnIm (slave) is generated • TAUBTTOUTm (slave) is set to an inactive level.
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

24.14.2.7 Specific Timing Diagrams

(1) TAUBnCDRm (master) = 0000_H

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

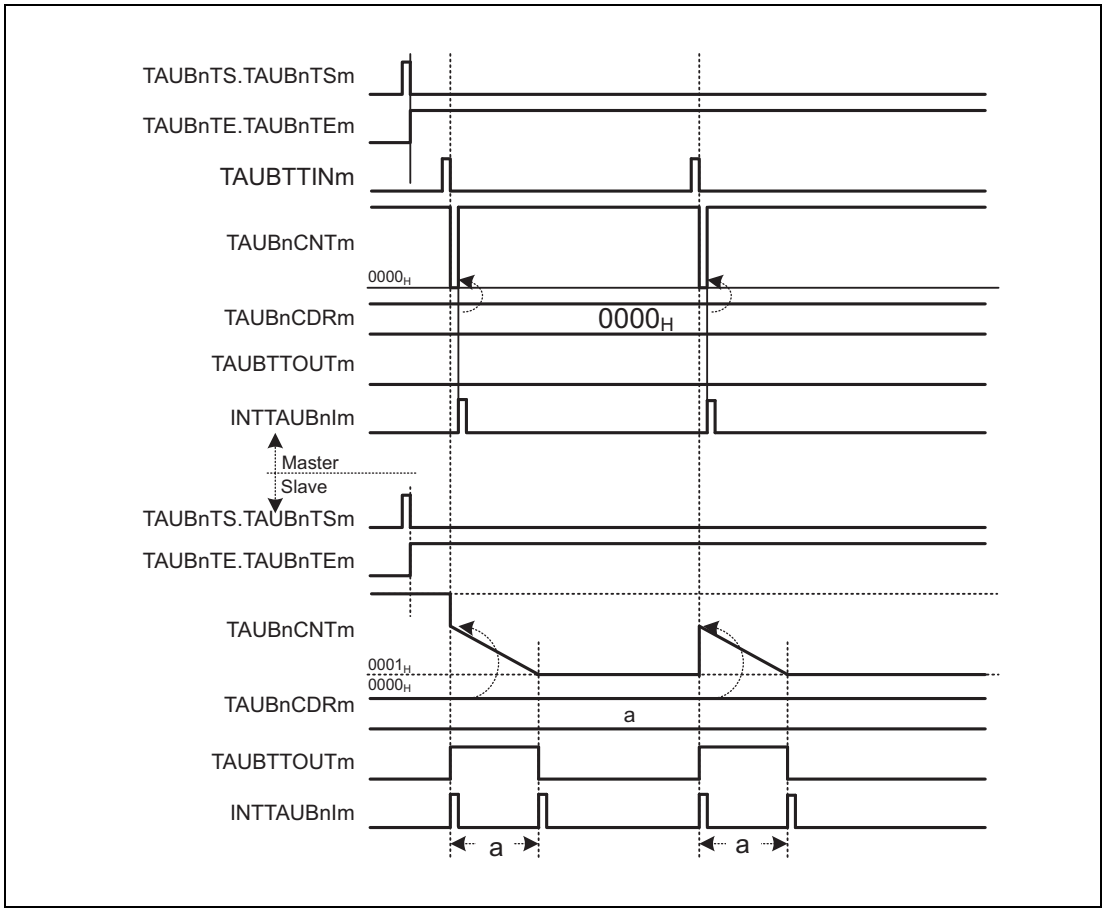


Figure 24.86 TAUBnCDRm (master) = 0000_H

- When a valid TAUBTTINm input edge is detected, the value 0000_H is written to TAUBnCNTm (master). The counter is set to 0000_H for one count and returns to $FFFF_H$.
  Thus, the slave channel starts to count down one count clock later to TAUBTTINm (master).

(2) TAUBnCDRm (slave) = 0000_H

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

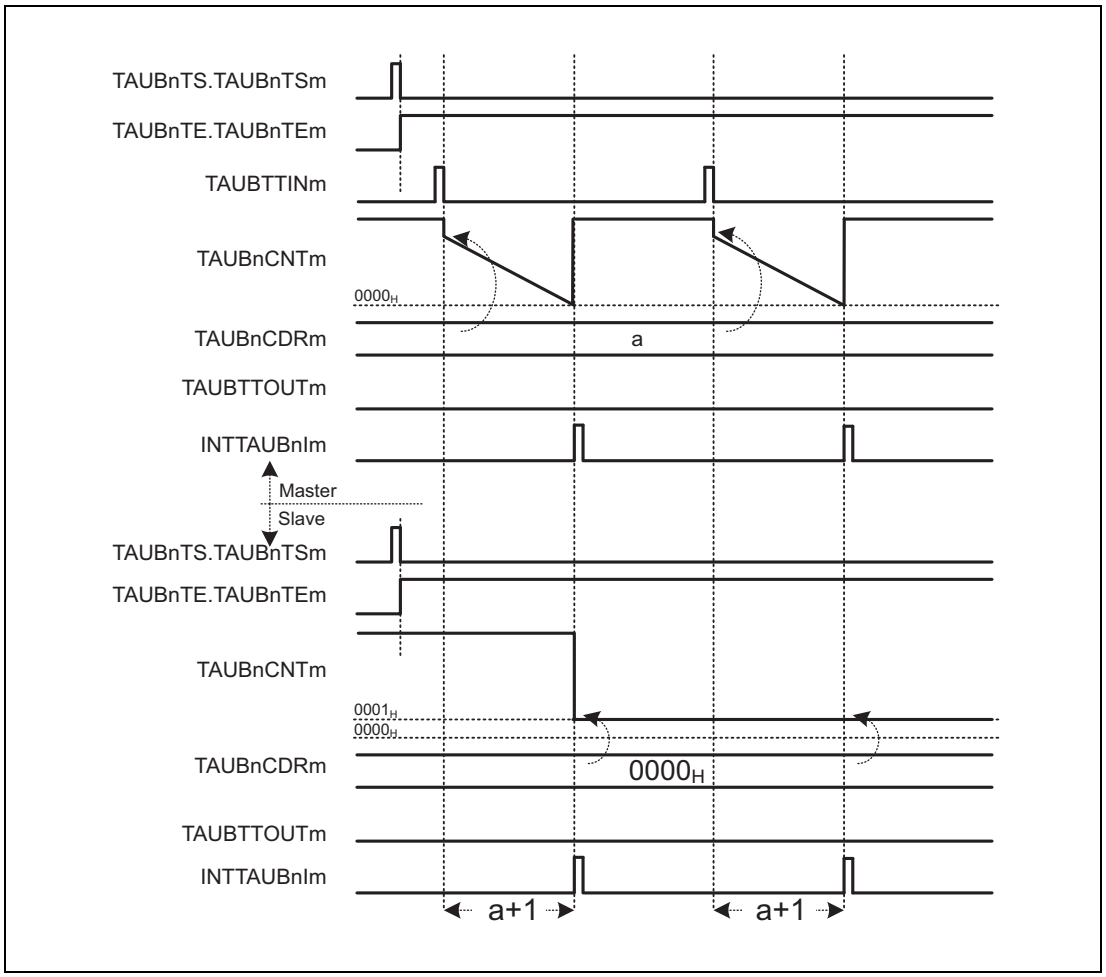


Figure 24.87 TAUBnCDRm (slave) = 0000_H

- TAUBTTOUTm remains in an inactive state, because the pulse width is zero.

(3) TAUBnCMORm.TAUBnMD0 = 0

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

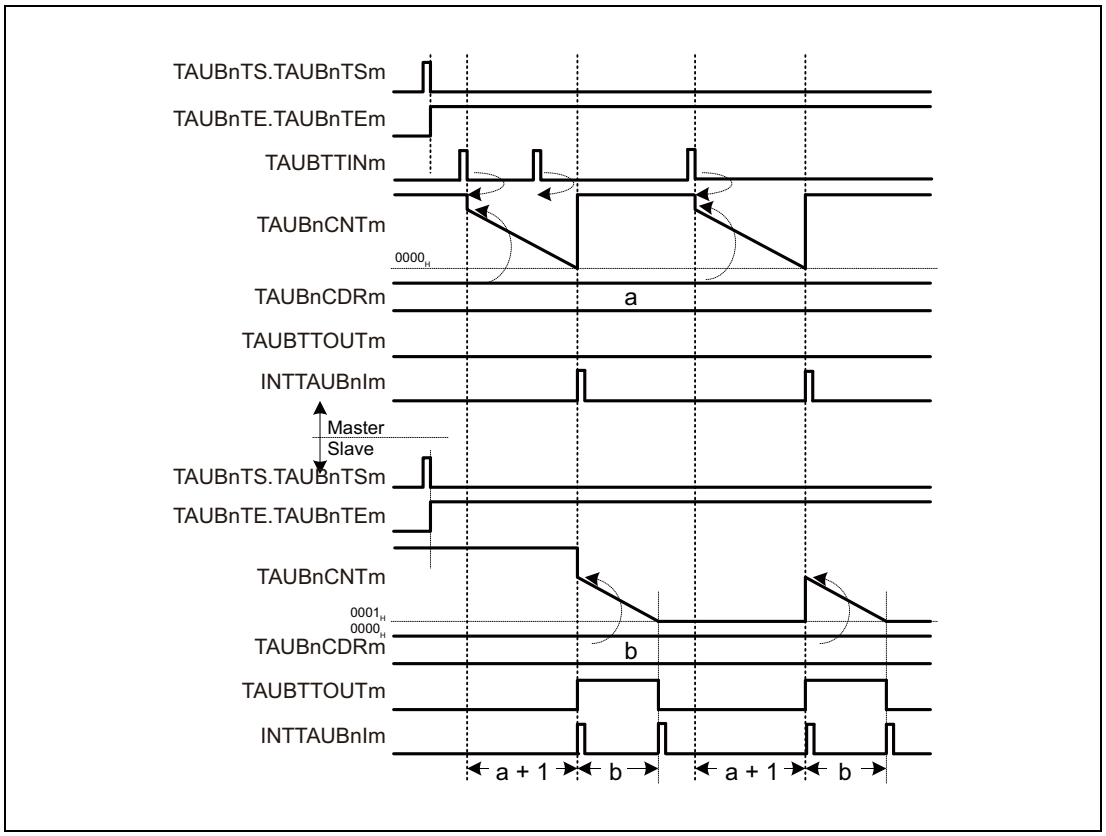


Figure 24.88 TAUBnCMORm.TAUBnMD0 = 0

- Even when an effective edge is input to TAUBTTINm while the counter of the master channel counts down, the counter continues counting down.

(4) TAUBnCMORm.TAUBnMD0 = 1

The following settings apply to this diagram.

- Start trigger detection enabled during counting (TAUBnCMORm.TAUBnMD0 = 1)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

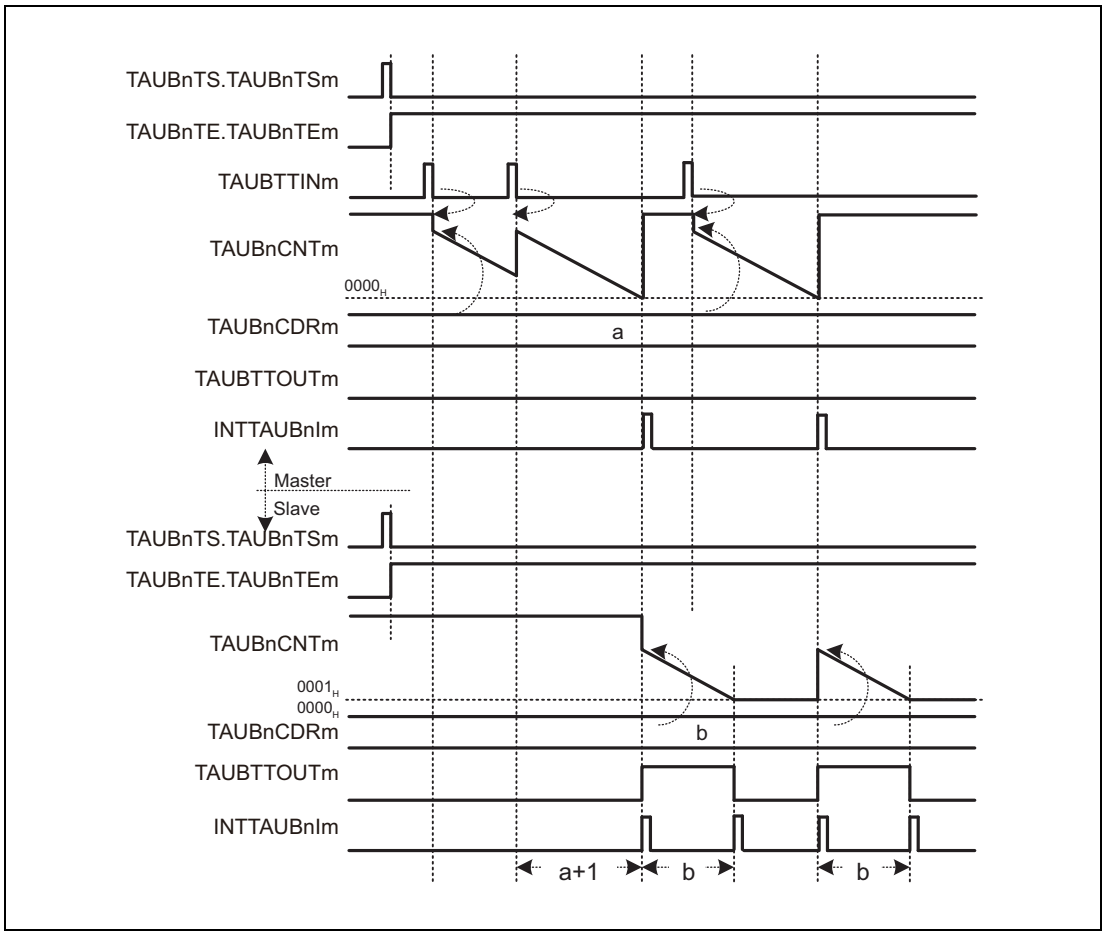


Figure 24.89 TAUBnCMORm.TAUBnMD0 = 1

- If a valid TAUBTTINm input edge is detected while the counter of the master channel counts down, TAUBnCNTm reloads the value of TAUBnCDRm. The counter restarts to count down.

This means the delay of the INTTAUBnIm generation interval is extended by the value of TAUBnCNTm at the time when a valid TAUBTTINm input edge is detected.

(5) Stopping and restarting the operation

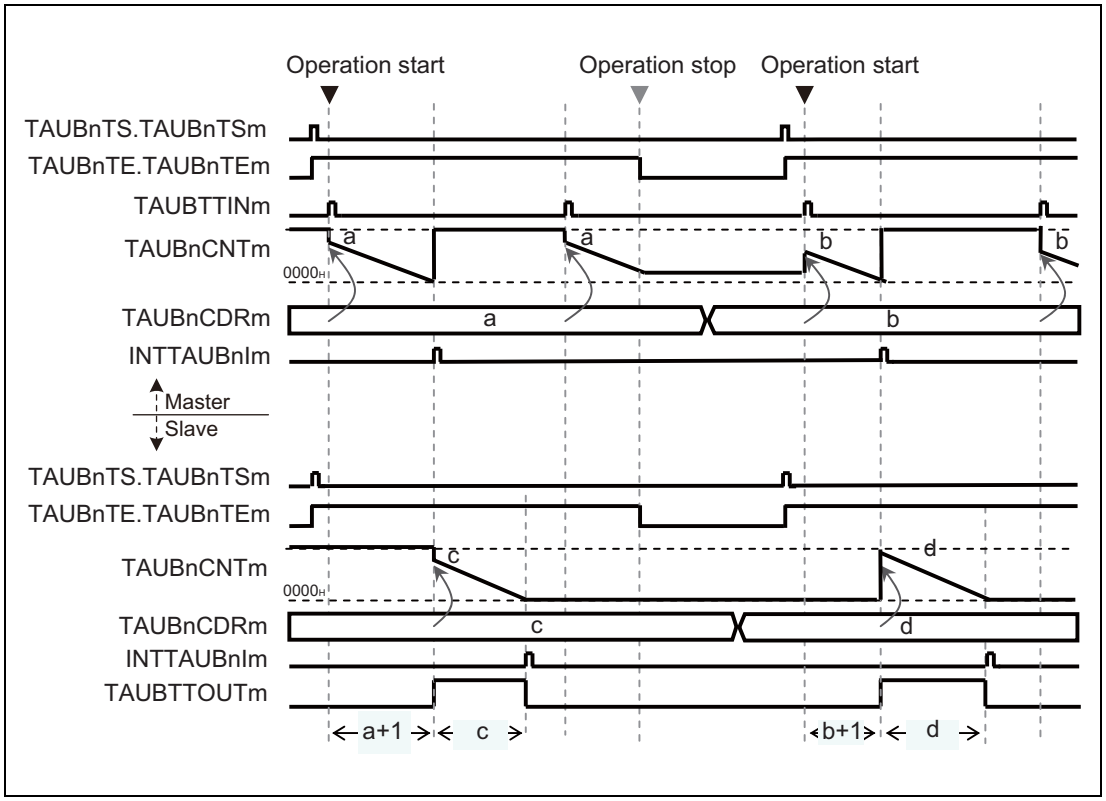


Figure 24.90 Stopping and Restarting the Operation

Setting TTm of the master and slave channels to 1 clears TAUBnTE.TAUBnTEm to 0, thereby stopping the count operation. If this happens, TAUBnCNTm and TAUBTTOUTm stop operation with the values retained.

Setting TAUBnTS.TAUBnTSm of the master and slave channels to 1 sets TAUBnTE.TAUBnTEm to 1.

When the start trigger is detected while the TAUBnTE.TAUBnTEm is set to 1, the TAUBnCDRm value is transferred to TAUBnCNTm and the operation restarts.

(6) Restarting the master channel while the slave channel is counting

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

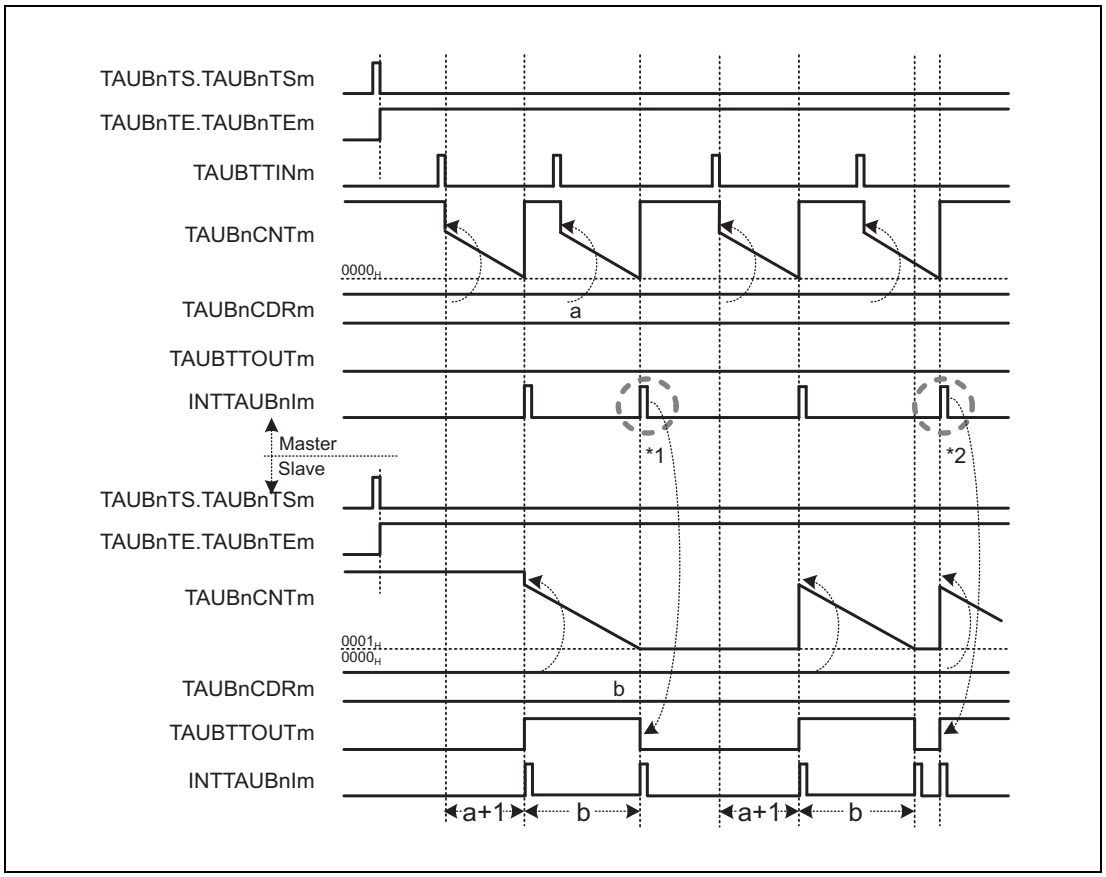


Figure 24.91 TAUBTTINm Input Interval ≤ Delay Time + Pulse Width + 1

- If the master channel generates an interrupt before the counter of the slave channel has reached 0001_H or exactly when 0001_H is reached (*1), the interrupt (master) is ignored.
- If an interrupt of the master channel occurs while the counter of the slave channel is awaiting the next trigger, the value of TAUBnCDRm (slave) is reloaded. An interrupt is generated and TAUBTTOUTm toggles. If TAUBnCNTm (master) has started to count down while the TAUBnCNTm (slave) is still counting (*2), TAUBTTOUTm is not output with the expected delay time.
- To generate the correct one-shot pulse, the start trigger for the master channel must be detected while the master and slave channels are waiting for the start trigger, and not while they are counting.

24.14.3 Delay Pulse Output Function

24.14.3.1 Overview

Summary

This function outputs two signals. The reference signal has a defined pulse width and pulse cycle specified using the master channel and slave channel 1.
Slave channels 2 and 3 output the reference signal with a specified delay. The delay signal is identical to the reference signal, but delayed by amount specified in slave channel 2.

The signal values are specified in the following way:

- The pulse cycle is specified using the master channel.
- The duty cycle of the reference signal is specified using slave channel 1.
  The duty cycle of the delay signal is specified using slave channel 3.
- The delay is specified in slave channel 2.

Prerequisites

- Four channels
- The operation mode of the master channel must be set to interval timer mode, see **Table 24.115, Contents of the TAUBnCMORm Register for the Master Channel of the Delay Pulse Output Function**.
- The operation mode of slave channels 1 and 2 must be set to one-count mode, see **Table 24.118, Contents of the TAUBnCMORm Register for the Slave Channel 1 of the Delay Pulse Output Function** and **Table 24.122, Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Delay Pulse Output Function**.
- The operation mode of slave channel 3 must be set to pulse one-count mode, see **Table 24.125, Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Delay Pulse Output Function**
- TAUBTTOUTm is not used for the master channel and slave channel 2
- The channel output mode of slave channel 1 must be set to synchronous channel output mode 1.
- The channel output mode of slave channel 3 must be set to independent channel output mode 1.

Description

The counters of the channel group are enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm to 1, enabling count operation.

- Master channel:

  The current value of TAUBnCDRm is loaded to TAUBnCNTm and the counter starts to count down from this value. INTTAUBnIm is generated on the master channel.
  When the counter of the master channel reaches 0000_H, indicating that the pulse time has elapsed, INTTAUBnIm is generated. The counter reloads the TAUBnCDRm value and counts down.

- Slave channels 1 and 2:

  When the slave channels 1 and 2 detect an interrupt from the master channel, they start to count down from the current value of TAUBnCDRm. The TAUBTTOUTm signal (slave 1) is set.

  – Slave channel 1:
  When the counter of slave channel 1 reaches 0000_H (duty time has elapsed) INTTAUBnIm is generated and the TAUBTTOUTm signal is reset. The counter is reset to $FFFF_H$ and awaits the next INTTAUBnIm of the master channel.

  – Slave channel 2:
  When the counter of slave channel 2 reaches 0000_H, delay time has elapsed and INTTAUBnIm is generated. The counter is reset to $FFFF_H$ and awaits the next INTTAUBnIm of the master channel.
  INTTAUBnIm (slave channel 2) triggers the counter of slave channel 3.

- Slave channel 3:
  When slave channel 3 detects an interrupt from slave channel 2, it starts to count down from the current value of TAUBnCDRm. INTTAUBnIm is generated and the TAUBTTOUTm signal (slave 3) is set.
  When the counter of slave channel 3 reaches 0001_H, INTTAUBnIm is generated and the TAUBTTOUTm signal is reset.
  The output from slave channel 3 is the delayed PWM pulse.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channels, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. Please see **Section 24.6, Simultaneous Rewrite**.

24.14.3.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUBnCDRm (slave 1)) × count clock cycle

Delay = (TAUBnCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUBnCDRm (slave 3)) × count clock cycle

Where the setting of the delay is within the following range:

$0000_H \leq$ TAUBnCDRm (slave 2) < TAUBnCDRm (master)

NOTES

1. The output waveform of TAUBTTOUTm (slave 3) is the output waveform of TAUBTTOUTm (slave 1) delayed for the delay generated by slave 2. It cannot be delayed for more than the pulse cycle.
2. When INTTAUBnIm of slave 2 occurs while slave 3 is counting, slave 3 restarts the operation. Therefore, the output waveform of TAUBTTOUTm (slave 3) retains the active level. (In this case, TAUBTTOUTm (Slave-CH-3) cannot output the waveform of the delayed basic pulse of TAUBTTOUTm (Slave-CH-1).)

24.14.3.3 Block Diagram and General Timing Diagram

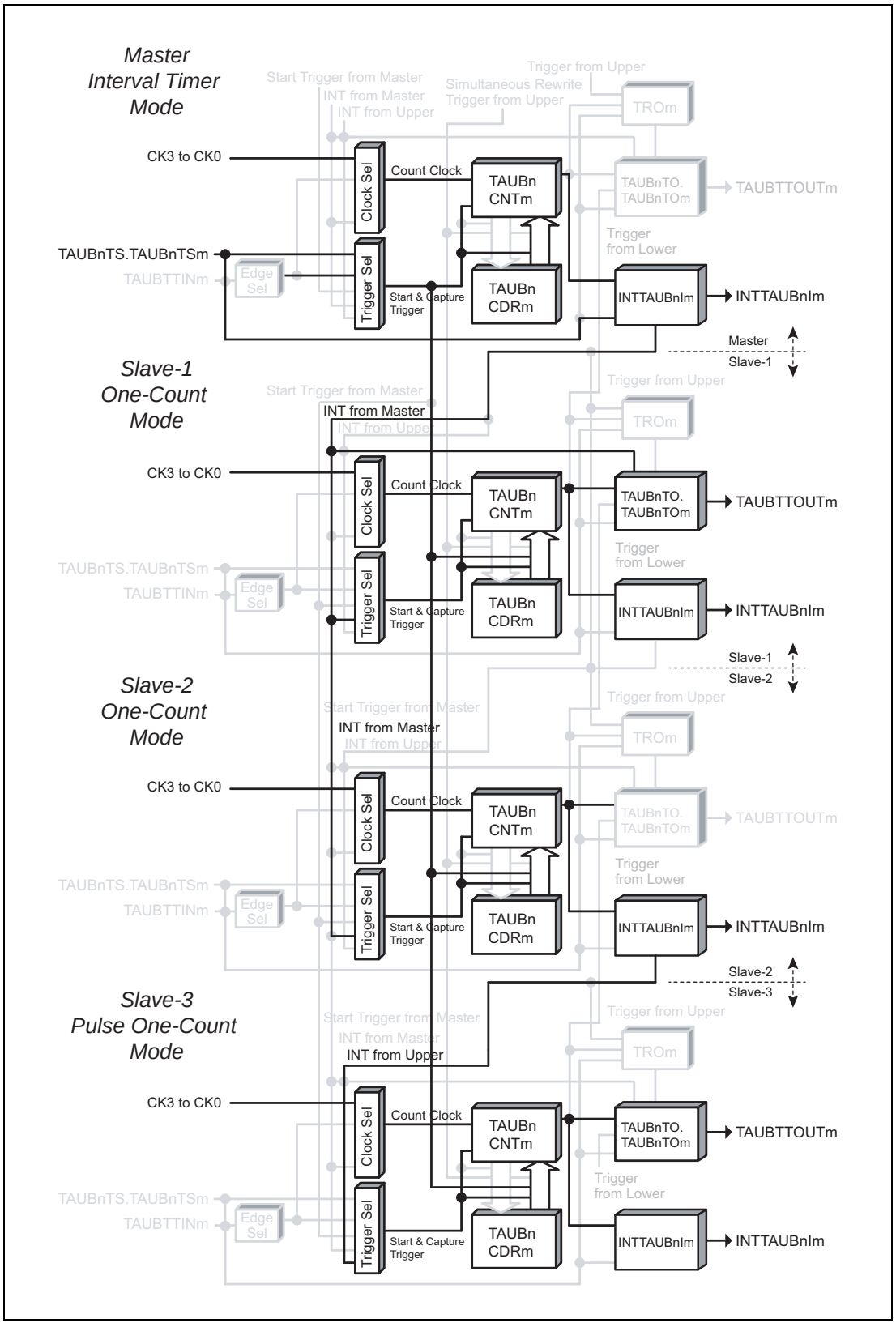


Figure 24.92 Block Diagram for Delay Pulse Output Function

The following settings apply to the general timing diagram.

- Slave channel 1: Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3: Positive logic (TAUBnTOL.TAUBnTOLm = 0)

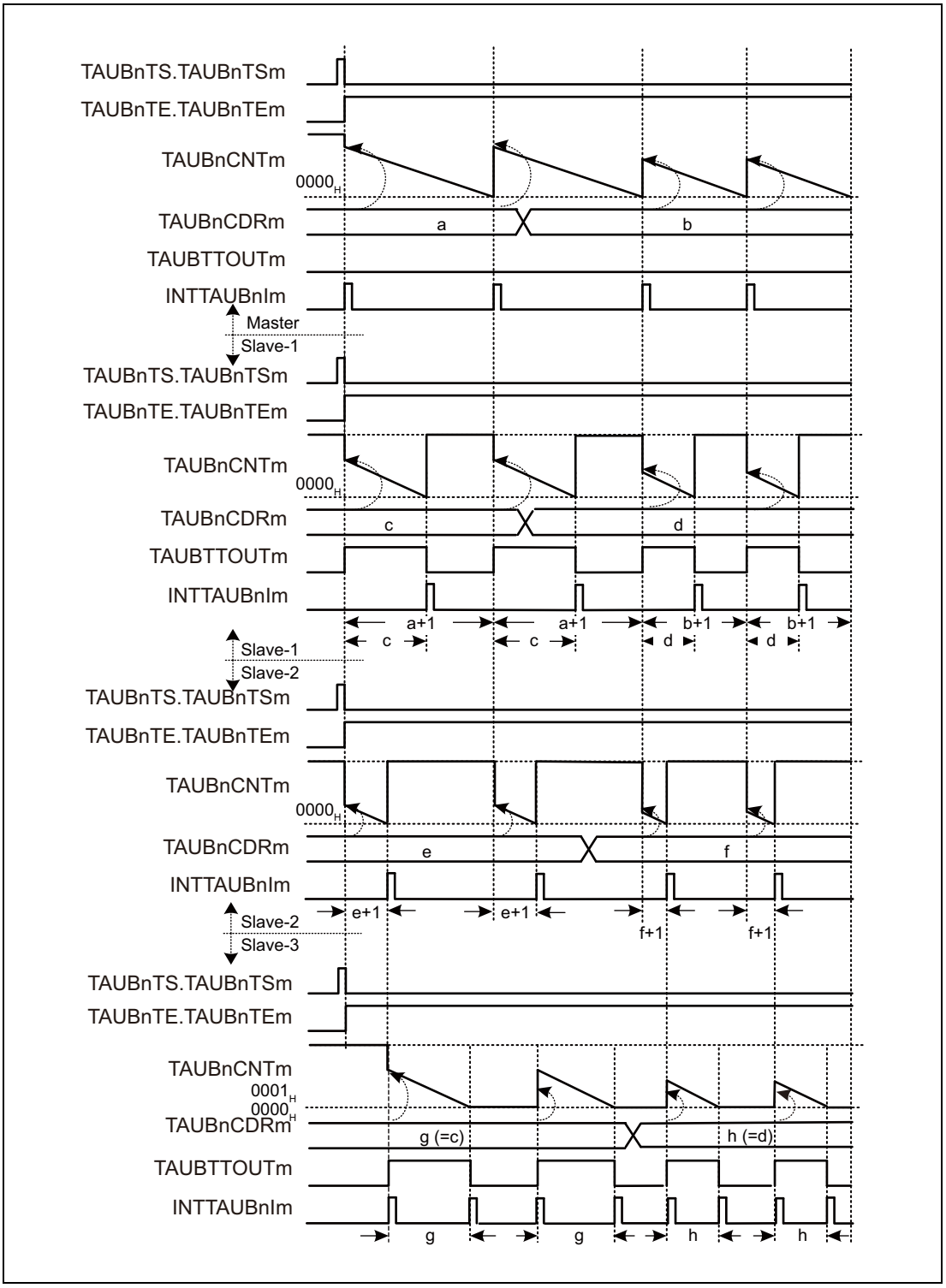


Figure 24.93 General Timing Diagram for Delay Pulse Output Function

24.14.3.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.115 Contents of the TAUBnCMORm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the master channel

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

Table 24.116 Contents of the TAUBnCMURm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by the master channel of this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.117 Simultaneous Rewrite Settings for the Master Channel of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

24.14.3.5 Register Settings for Slave Channel 1

(1) TAUBnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.118 Contents of the TAUBnCMORm Register for the Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 100_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for slave channel 1

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

Table 24.119 Contents of the TAUBnCMURm Register for the Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 1

Table 24.120 Control Bit Settings for Slave Channel 1 of the Synchronous Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for slave channel 1

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.121 Simultaneous Rewrite Settings for Slave Channel 1 of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

24.14.3.6 Register Settings For Slave Channel 2

(1) TAUBnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.122 Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 100_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for slave channel 2

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

Table 24.123 Contents of the TAUBnCMURm Register for the Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 2

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite for slave channel 2

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.124 Simultaneous Rewrite Settings for Slave Channel 2 of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

24.14.3.7 Register Settings for Slave Channel 3

(1) TAUBnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.125 Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channels must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 101_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for slave channel 3

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

Table 24.126 Contents of the TAUBnCMURm Register for the Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 3

Table 24.127 Control Bit Settings for Independent Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for slave channel 3

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.128 Simultaneous Rewrite Settings for Slave Channel 3 of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

24.14.3.8 Operating Procedure for Delay Pulse Output Function

Table 24.129 Operating Procedure for Delay Pulse Output Function (1/2)

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.3.4, Register Settings for the Master Channel**. Slave channel 1: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.3.5, Register Settings for Slave Channel 1**. Slave channel 2: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.3.6, Register Settings For Slave Channel 2**. Slave channel 3: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.3.7, Register Settings for Slave Channel 3**. Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.

Table 24.129 Operating Procedure for Delay Pulse Output Function (2/2)

Restart operation

	Operation	Status of TAUBn
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master channel and slave channels 1 and 2 start. INTTAUBnIm is generated on the master channel and TAUBTTOUTm (slave 1) is set.
During operation	TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master channel and slave channels 1 and 2 load TAUBnCDRm and count down. When the counter of the master channel reaches 0000_H: • INTTAUBnIm (master) is generated • TAUBnCNTm (master) loads the TAUBnCDRm value and continues count operation • TAUBnCNTm (slave 1 and slave 2) reload the TAUBnCDRm value and start counting down • TAUBTTOUTm (slave 1) is set When TAUBnCNTm (slave 1) reaches 0000_H: • INTTAUBnIm (slave 1) is generated • TAUBTTOUTm (slave 1) is reset When TAUBnCNTm (slave 2) reaches 0000_H: • INTTAUBnIm (slave 2) is generated • INTTAUBnIm (slave 3) is generated • TAUBTTOUTm (slave 3) is set • TAUBnCNTm (slave 3) reloads the TAUBnCDRm value and starts counting down When TAUBnCNTm (slave 3) reaches 0001_H: • INTTAUBnIm (slave 3) is generated • TAUBTTOUTm (slave 3) is reset
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

24.14.3.9 Specific Timing Diagrams

(1) Duty cycle (slave 3) = 100%

The following values apply to the figure below.

- TAUBnCDRm (master) = $000A_H$
- TAUBnCDRm (slave 1) = $000B_H$
- TAUBnCDRm (slave 2) = 0000_H
- TAUBnCDRm (slave 3) = $000B_H$

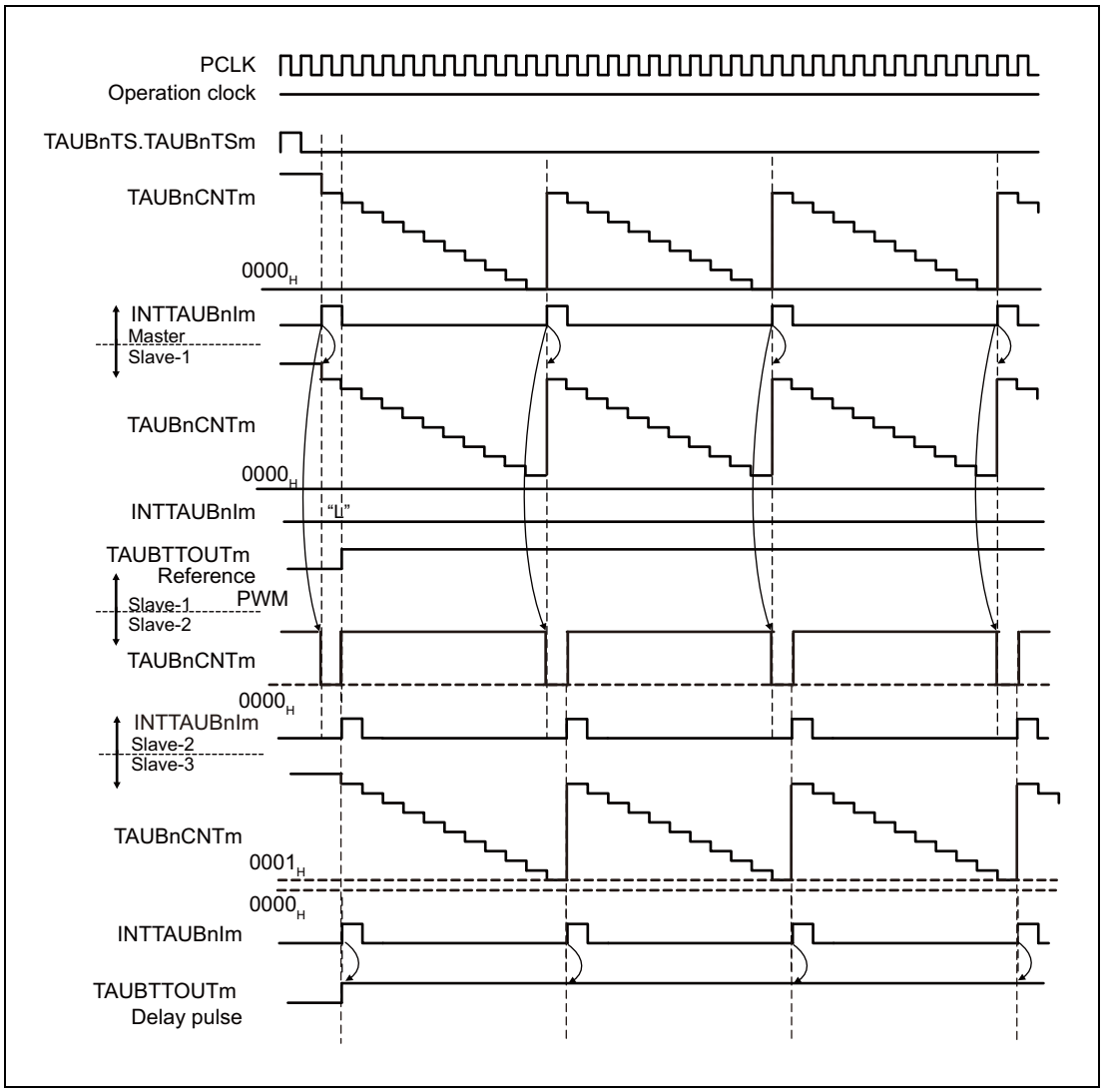


Figure 24.94 Duty Cycle (slave 3) = 100%

- If the value of TAUBnCDRm (slave 1 and 3) is higher than the value of TAUBnCDRm (master), the counter of slave channel 1 cannot reach 0000H and cannot generate interrupt request signals. TAUBTTOUTm of channels 1 and 3 remain in the active state.

(2) TAUBTTOUTm (slave 1) = TAUBTTOUTm (slave 3)

The following values apply to the figure below.

- TAUBnCDRm (master) = $000A_H$
- TAUBnCDRm (slave 1) = 0005_H
- TAUBnCDRm (slave 2) = 0000_H
- TAUBnCDRm (slave 3) = 0005_H

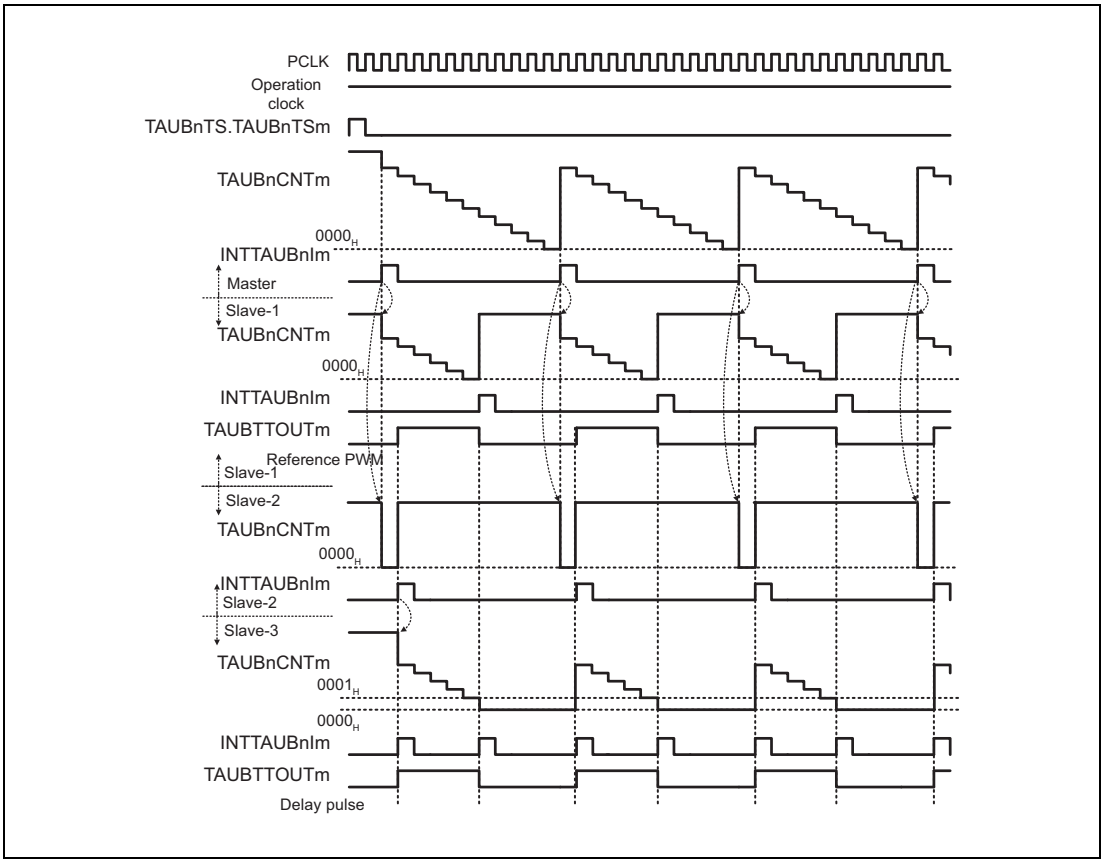


Figure 24.95 TAUBTTOUTm (slave 1) = TAUBTTOUTm (slave 3)

- If TAUBnCDRm (slave 2) = 0000_H, the counter of slave channel 3 starts counting one count clock later than the counter of slave channel 1. The reference pulse and the delay pulse are output with a delay of one clock count.

24.14.4 A/D Conversion Trigger Output Function Type 1

24.14.4.1 Overview

Summary

This function is identical to **Section 24.14.1, PWM Output Function** except that TAUBTTOUTm is not output.

This is achieved by setting the channel output mode of the slave to independent channel output mode controlled by software.

24.14.4.2 Block Diagram and General Timing Diagram

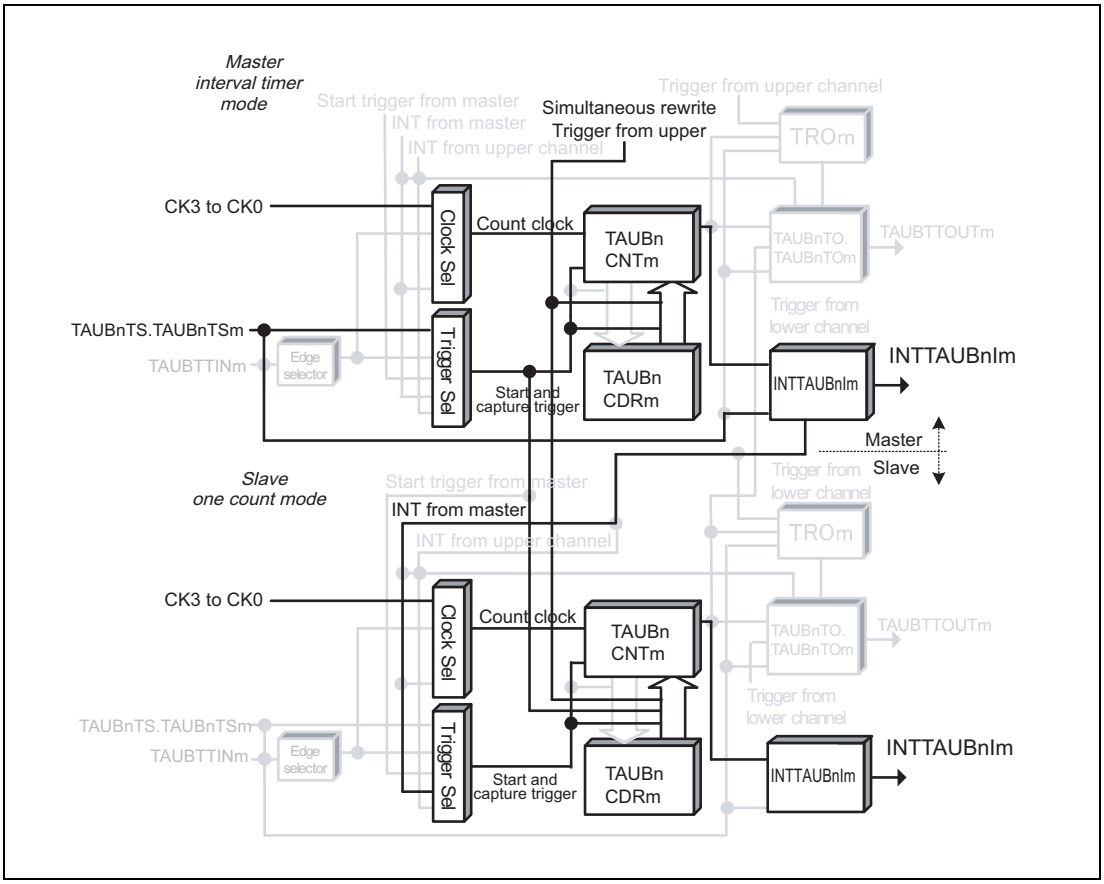


Figure 24.96 Block Diagram for A/D Conversion Trigger Output Function Type 1

The following settings apply to the general timing diagram.

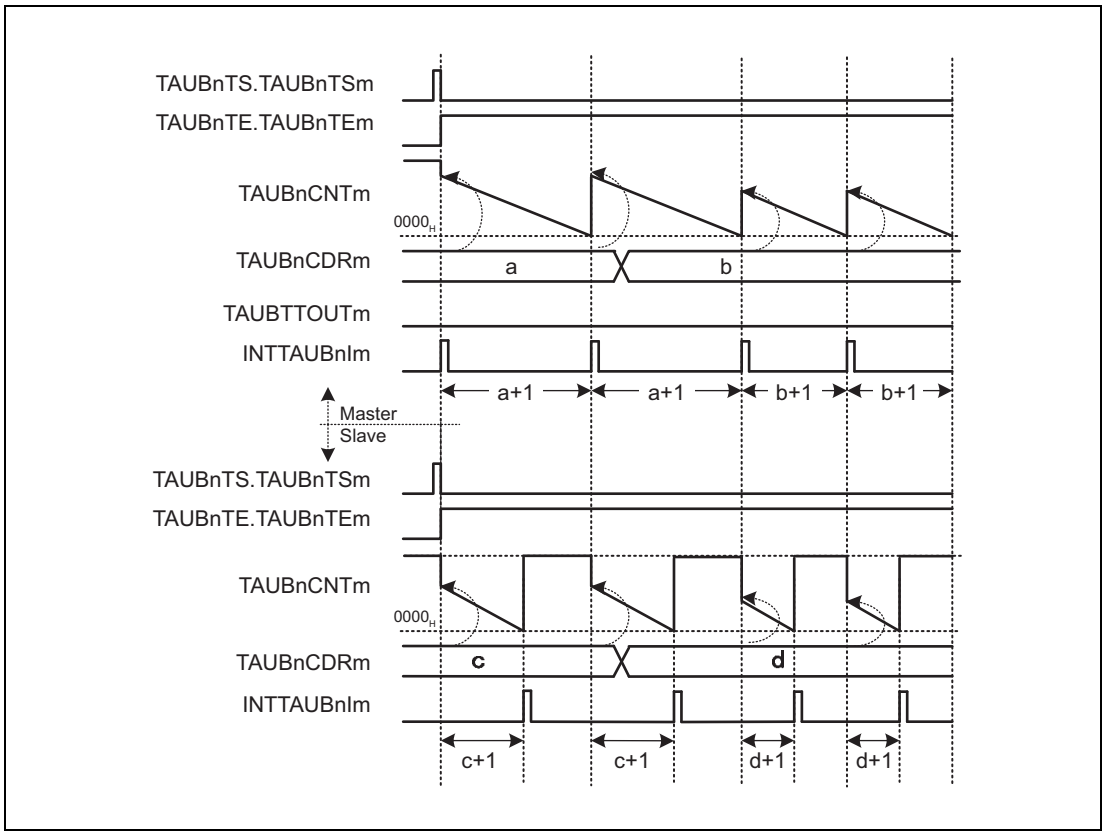


Figure 24.97 General Timing Diagram for A/D Conversion Trigger Output Function Type 1

24.14.5 Triangular Wave PWM Output Function

24.14.5.1 Overview

Summary

This function generates multiple triangular wave PWM outputs by using a master and one or more slave channels. It enables the pulse cycle (frequency) and the duty cycle of TAUBTTOUTm to be set using the master and slave channel(s) respectively.

The master channel generates a carrier cycle. The first cycle of the master channel controls the down status and the second cycle controls the up status of the slaves counter.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode, see **Table 24.130, Contents of the TAUBnCMORm Register for the Master Channel of the Triangular Wave PWM Output Function**.
- The operation mode of the slave channel(s) must be set to up down count mode, see **Table 24.134, Contents of the TAUBnCMORm Register for the Slave Channel of the Triangular Wave PWM Output Function**.
- The output mode of the master channel must be set to independent channel output mode 1.
- The output mode of the slave channel(s) must be set to synchronous channel output mode 2.
- The following settings establish TAUBTTOUTm at high level for the down status of the carrier cycle.
  - If the TAUBnCMORm.TAUBnMD0 (master) bit is set to 0,TAUBnTO.TAUBnTOm must be set to 1 while TAUBnTOE.TAUBnTOEm is 0. (recommended)
  - If the TAUBnCMORm.TAUBnMD0 (master) bit is set to 1, TAUBnTO.TAUBnTOm must be set to 0 while TAUBnTOE.TAUBnOEm is 0.

Functional description

The counters are enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1 for every channel. This in turn sets TAUBnTE.TAUBnTEm, enabling count operation. The current values of TAUBnCDRm (master and slave) are loaded to TAUBnCNTm (master and slave) and the counters start to count down from these values. If the master channel TAUBnCMORm.TAUBnMD0 bit is set to 1, an interrupt is generated and TAUBTTOUTm signal of the master toggles.

- Master channel:
  When the counter of the master channel reaches 0000_H (pulse cycle time has elapsed) INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. TAUBnCNTm then reloads the TAUBnCDRm value and counts down.

- Slave channel:
  The INTTAUBnIm of the master channel triggers the counter of the slave channel:
  - If the slave counter currently counts down, it changes count direction.
  - If the slave counter currently counts up, the value of TAUBnCDRm is reloaded and the counter counts down.

  When the counter of the slave channel reaches 0001_H while counting up or down, INTTAUBnIm is generated and the TAUBTTOUTm (slave) signal is set or reset.

  The counter continues to count down or up and awaits the next INTTAUBnIm of the master channel.

  TAUBTTOUTm can be switched between positive and negative phase setting TAUBnTOL.TAUBnTOLm during operation.

The counters can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values.

Conditions

Simultaneous rewrite can be used with this function. Please see **Section 24.6, Simultaneous Rewrite**.

24.14.5.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUBnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUBnCDRm (master) + 1) × 2 × count clock cycle

Duty cycle =

[(TAUBnCDRm (master) + 1 - TAUBnCDRm (slave))/ (TAUBnCDRm (master) + 1)] × 100

- Duty cycle = 100%
  TAUBnCDRm (slave) = 0000_H
- Duty cycle = 0%
  TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1

24.14.5.3 Block Diagram and General Timing Diagram

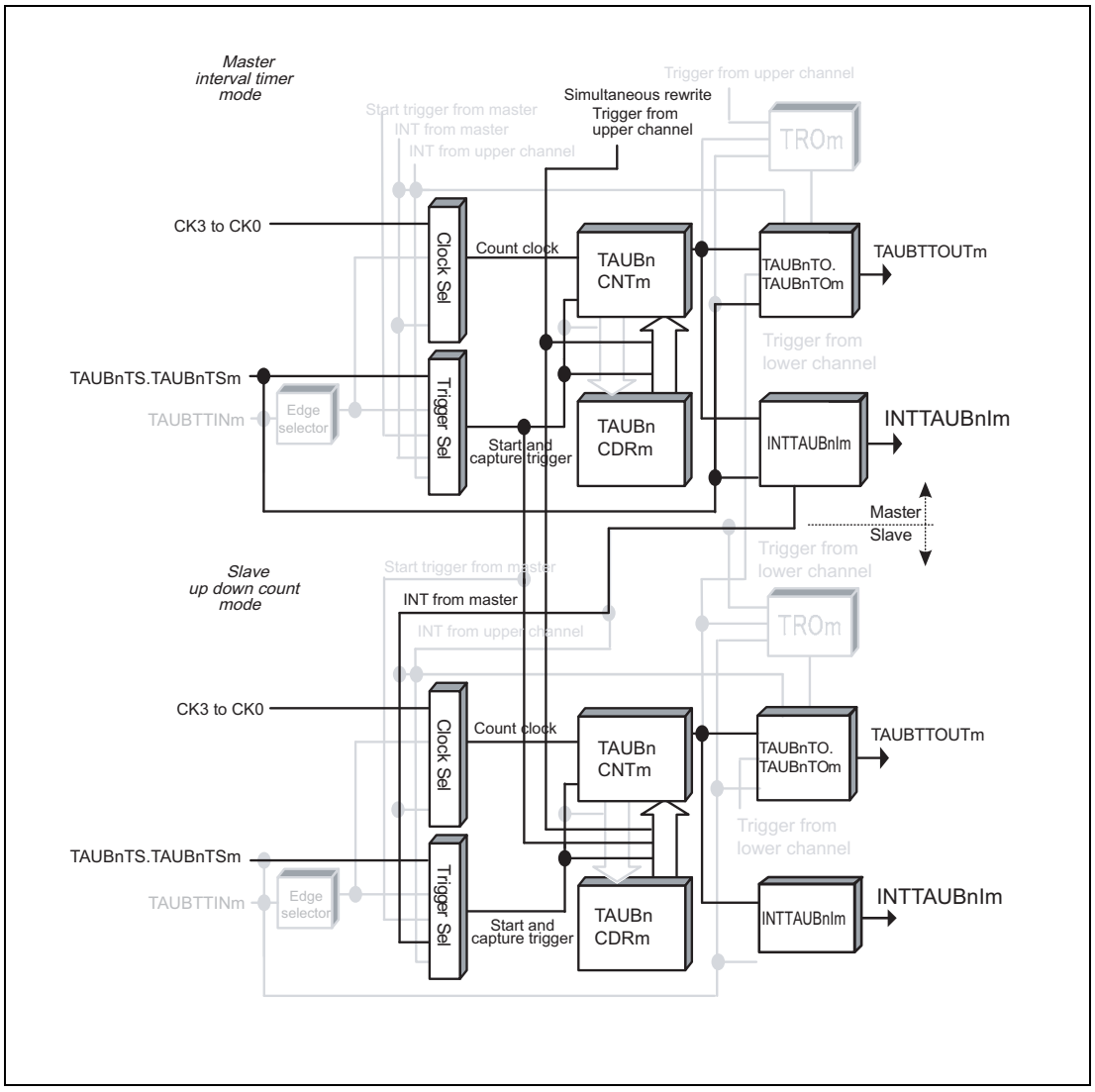


Figure 24.98 Block Diagram for Triangular Wave PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

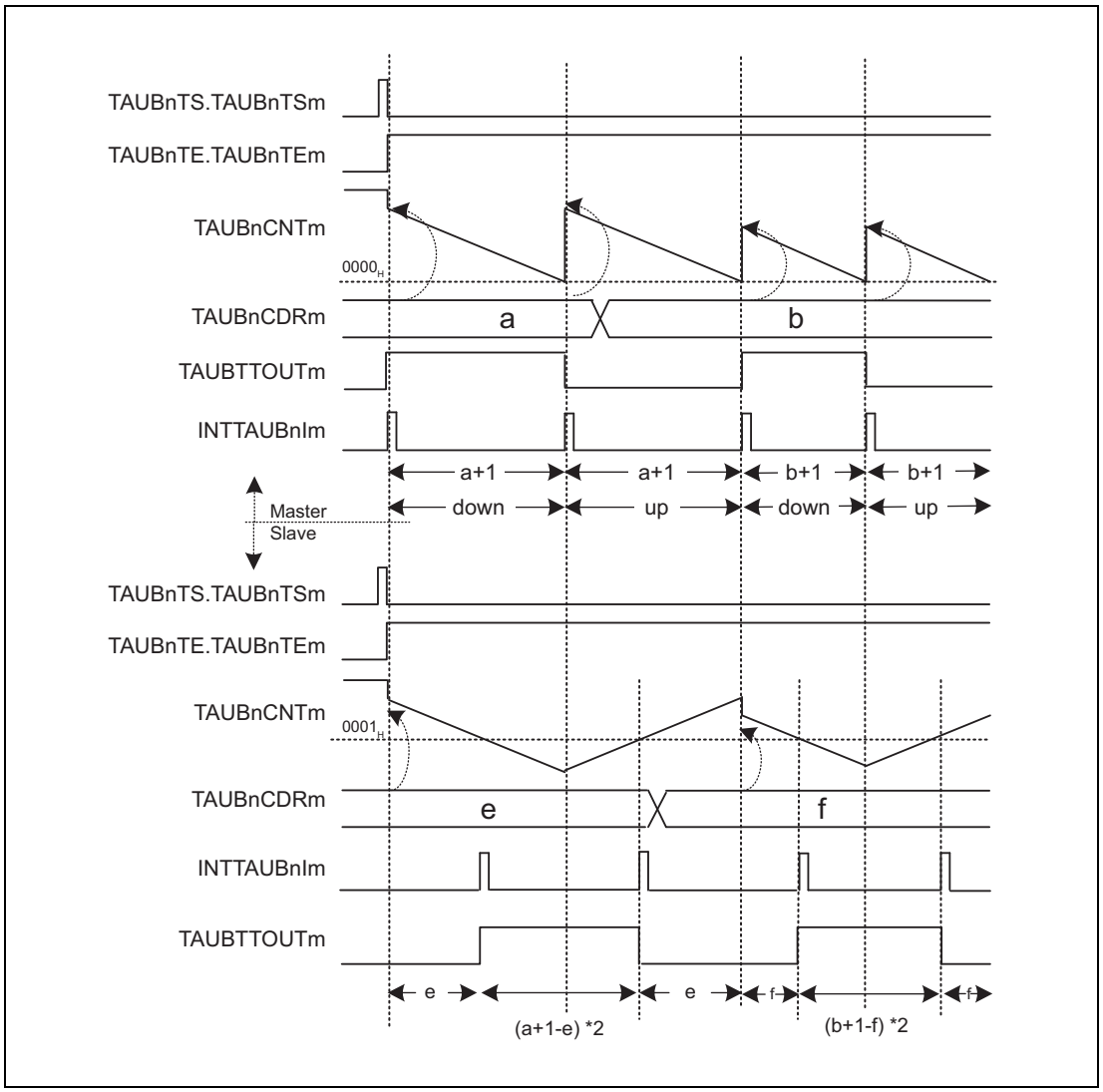


Figure 24.99 General Timing Diagram for Triangular Wave PWM Output Function

24.14.5.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.130 Contents of the TAUBnCMORm Register for the Master Channel of the Triangular Wave PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated and TAUBTTOUTm does not toggle at operation start 1: Generates INTTAUBnIm and toggles TAUBTTOUTm at operation start

(2) TAUBnCMURm for the master channel

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

Table 24.131 Contents of the TAUBnCMURm Register for the Master Channel of the Triangular Wave PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

Table 24.132 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	Write 0_B.
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.133 Simultaneous Rewrite Settings for the Master Channel of the Triangular Wave PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

NOTE

If TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel that generates the simultaneous rewrite trigger signal.

24.14.5.5 Register Settings for the Slave Channel(s)

(1) TAUBnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.134 Contents of the TAUBnCMORm Register for the Slave Channel of the Triangular Wave PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 111_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1001_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm for the slave channel(s)

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

Table 24.135 Contents of the TAUBnCMURm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the slave channel(s)

Table 24.136 Control Bit Settings for Synchronous Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.137 Simultaneous Rewrite Settings for the Slave Channel of the Triangular Wave PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

24.14.5.6 Operating Procedure for Triangular Wave PWM Output Function

Table 24.138 Operating Procedure for Triangular Wave PWM Output Function

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.5.4, Register Settings for the Master Channel** Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.5.5, Register Settings for the Slave Channel(s)** Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnIm (master) is generated on the master channel when TAUBnCMORm.TAUBnMD0 set to 1.
During operation	TAUBnCDRm can be changed at any time. TAUBnTOL.TAUBnTOLm can be changed. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master and slave channels loads TAUBnCDRm and counts down. When the counter of the master channel reaches 0000_H: • INTTAUBnIm (master) is generated • TAUBTTOUTm (master) toggles • TAUBnCNTm (master) loads the TAUBnCDRm value and continues count operation. • TAUBnCNTm (slave) loads the TAUBnCDRm value or counts in the reverse direction. When TAUBnCNTm of the slave = 0001_H: • INTTAUBnIm (slave) is generated • TAUBTTOUTm (slave) is set (in count-down status) or reset (in count-up status)
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

24.14.5.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnCDRm = a = 5_H
- Slave channel:
  - TAUBnCDRm = 6_H

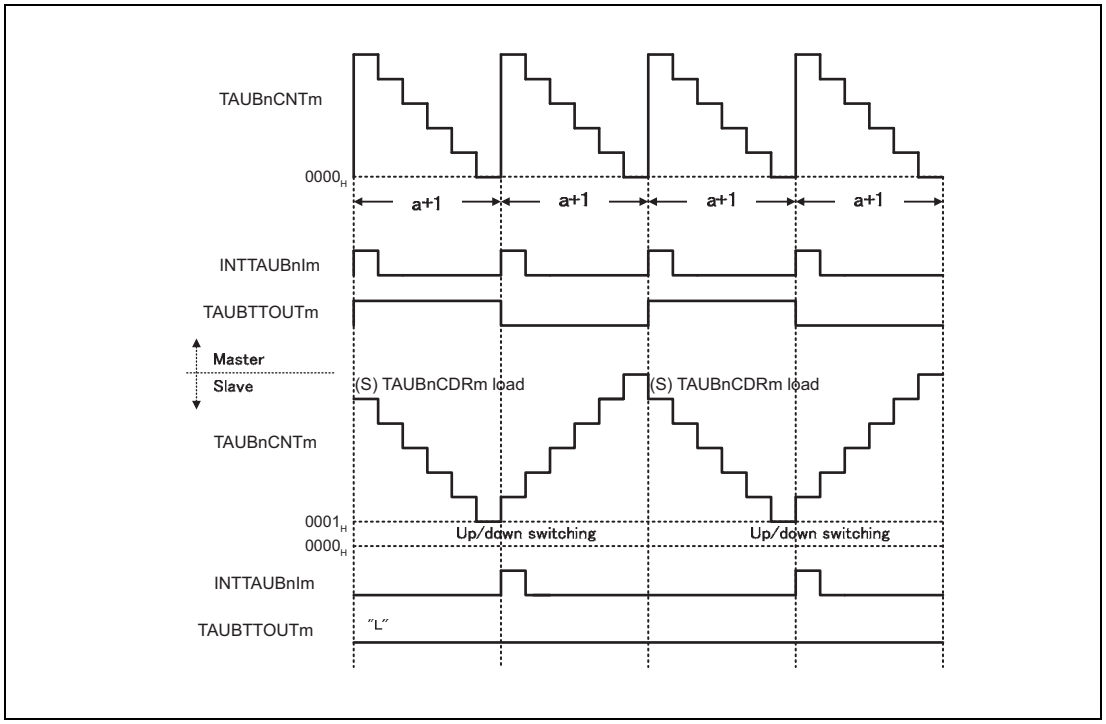


Figure 24.100 TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1

- If TAUBnCDRm (slave) ≥ TAUBnCDRm (master) +1, INTTAUBnIm of slave channel is not generated during counting down. The set signal is never detected, so TAUBTTOUTm remains at low state.

(2) Duty cycle = 100%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnCDRm = a = 5_H
- Slave channel:
  - TAUBnCDRm = 0_B

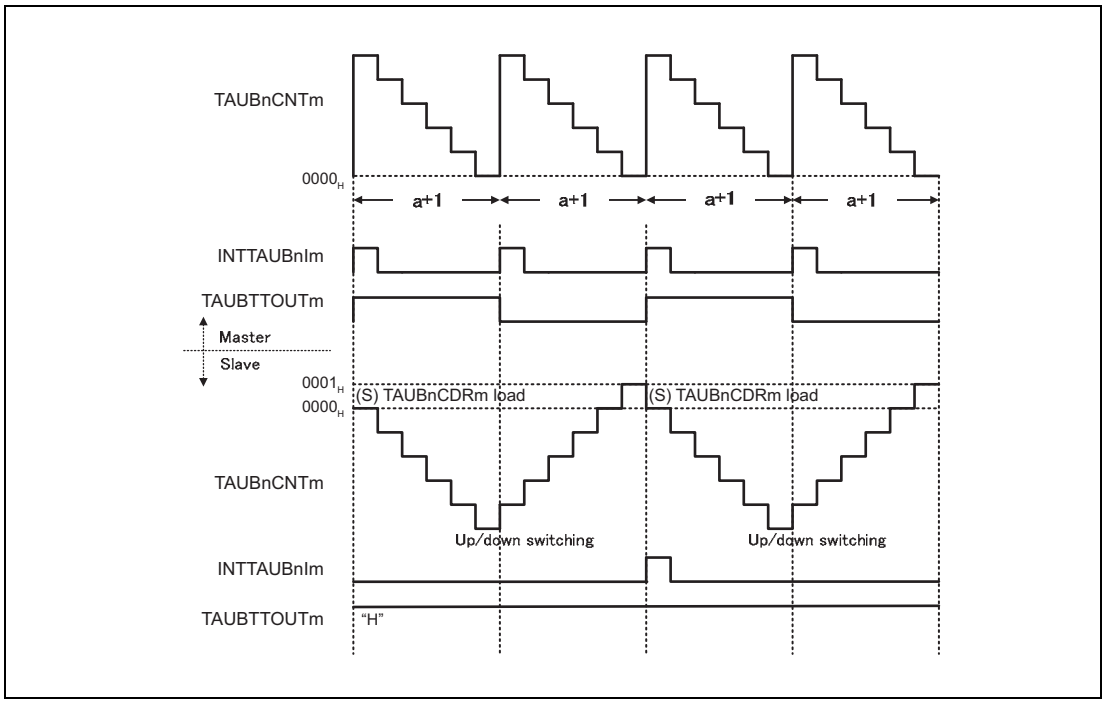


Figure 24.101 TAUBnCDRm (slave) = 0000_H

- If TAUBnCDRm (slave) = 0000_H, INTTAUBnIm of slave channel is not generated during counting up. The reset signal is never detected, so TAUBTTOUTm remains at high state.

24.14.6 Triangular Wave PWM Output Function with Dead Time

24.14.6.1 Overview

Summary

This function generates multiple triangular wave PWM outputs with a defined dead time by using a master and two or more slave channels. The resulting PWM signals with the dead time are output via TAUBTTOUTm of the slave channels 2 and 3. It enables the pulse cycle (frequency) and the duty cycle of TAUBTTOUTm to be set using the master and slave channel(s) respectively.

The master generates a carrier cycle. The first pulse controls the down status and the second pulse controls the up status of the slaves counter.

An interrupt on slave 2 causes TAUBTTOUTm of the slave channels to be set or reset.
Depending on the settings of TAUBnTDL.TAUBnTDLm, delay time is added to positive or negative logic side of the signal (i.e. whether TAUBTTOUTm is set or reset immediately or after dead time has elapsed). The duration of the dead time is specified by slave channel 3.

Prerequisites

- Three channels. Select an even channel CH (a) and an odd channel CH (a + 1) for slave channel 2 and 3 respectively.
- The operation mode of the master channel must be set to interval timer mode, see **Table 24.140, Contents of the TAUBnCMORm Register for the Master Channel of the Triangular Wave PWM Output Function with Dead Time.**
- Slave channel 1 is not used for this function. This ensures that slave channel 2 is an even-numbered channel (a), and slave channel 3 is an odd-numbered channel (a + 1). Slave channel 1 can be used as a separate timer (independent function).
- The operation mode of slave channel 2 must be set to up down mode, see **Table 24.144, Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Triangular Wave PWM Output Function with Dead Time.**
  Furthermore, slave channel 2 must be an even channel.
- The operation mode of slave channel 3 must be set to one-count mode, see **Table 24.148, Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Triangular Wave PWM Output Function with Dead Time.**
  Furthermore, slave channel 3 must be an odd channel.
- The channel output mode of the master channel must be set to independent channel output mode 1.
- The channel output mode of the slave channels 2 and 3 must be set to synchronous channel output mode 2.
- The following settings establish TAUBTTOUTm at high level for the down status of the carrier cycle.
  - If the TAUBnCMORm.MD0 (master) bit is set to 0,TAUBnTO.TAUBnTOm must be set to 1 while TAUBnTOE.TAUBnTOEm is 0. (recommended)
  - If the TAUBnCMORm.MD0 (master) bit is set to 1, TAUBnTO.TAUBnTOm must be set to 0 while TAUBnTOE.TAUBnTOEm is 0.

NOTE

Slave channel 1 is not used for Triangular Wave PWM Output Function with Dead Time. Slave channel 1 can be used as a separate timer (independent function).

Functional description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm to 1, enabling count operation. The current values of TAUBnCDRm are loaded to TAUBnCNTm and the counters start to count down from these TAUBnCDRm values. If the master channel TAUBnCMORm.TAUBnMD0 bit is set to 1, an interrupt is generated and TAUBTTOUTm signal of the master toggles.

- Master channel:
  When the counter of the master channel reaches 0000_H, INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. The counter reloads the TAUBnCDRm value and counts down.
- Slave channel 2:
  The INTTAUBnIm of the master channel triggers the counter of the slave channel 2:
  - If the slave counter currently counts down, it changes count direction.
  - If the slave counter currently counts up, the value of TAUBnCDRm is reloaded and the counter counts down.

  The counter continues to count down or up and awaits the next INTTAUBnIm of the master channel.
  When the counter value of slave channel 2 reaches 0001_H, INTTAUBnIm is generated.
- Slave channel 3:
  INTTAUBnIm of slave channel 2 triggers the counter of slave channel 3. The current value of TAUBnCDRm (slave 3) is loaded to TAUBnCNTm (slave 3) and the counter starts to count down from this TAUBnCDRm value.
  When the counter reaches 0000_H, INTTAUBnIm is generated. The counter is reset to $FFFF_H$ and awaits the next INTTAUBnIm of slave channel 2.

The TAUBnTDL.TAUBnTDLm settings of the corresponding channel specify whether it is set/reset immediately, or after dead time has elapsed, as shown in **Table 24.139, Behavior of TAUBTTOUTm when an Interrupt Occurs on Slave Channel 2**.

The TAUBnTOL.TOLm settings specify whether set corresponds to a high signal (TAUBnTOL.TAUBnTOLm = 0) or a low signal (TAUBnTOL.TAUBnTOLm = 1).

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values.

TAUBnCDRm value of slave channel 2 can be set to 0000_H to output 100% TAUBTTOUTm.

NOTE

If a forced restart is executed during operation, TAUBTTOUTm is not output as a triangle PWM signal.

Conditions

Simultaneous rewrite can be used with this function. Please see **Section 24.6, Simultaneous Rewrite**.

TAUBnTOL.TAUBnTOLm and TAUBnTDL.TAUBnTDLm bits must be set before the counter starts, and slave channels 2 and 3 should have opposite TAUBnTOL.TAUBnTOLm settings or opposite TAUBnTDL.TAUBnTDLm settings.

Table 24.139 Behavior of TAUBTTOUTm when an Interrupt Occurs on Slave Channel 2

TAUBnTDL.TAUBnTDLm	Count Direction of Slave Channel 2 when Interrupt is Generated	TAUBTTOUTm Set/Reset Timing
0	Down	Set after dead time has elapsed
	Up	Reset immediately after the interrupt
1	Down	Set immediately
	Up	Reset after dead time has elapsed

24.14.6.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUBnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUBnCDRm (master) + 1) × 2 × count clock cycle

PWM signal width (positive phase) = [(TAUBnCDRm (master) + 1 – TAUBnCDRm (slave 2)) × 2 – (TAUBnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (negative phase) = [(TAUBnCDRm (master) + 1 – TAUBnCDRm (slave 2)) × 2 + (TAUBnCDRm (slave 3) + 1)] × count clock cycle

24.14.6.3 Block Diagram and General Timing Diagram

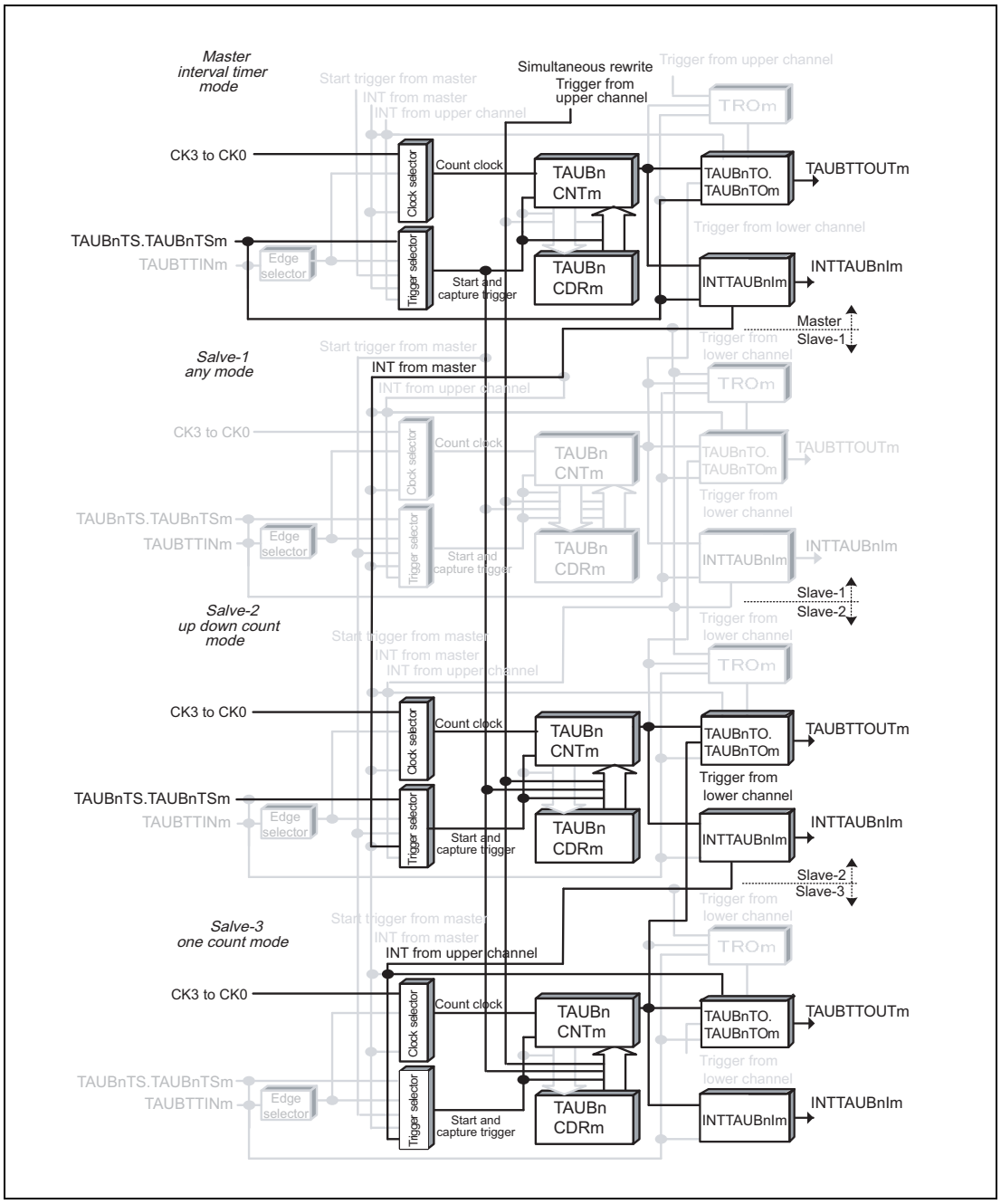


Figure 24.102 Block Diagram for Triangular Wave PWM Output Function with Dead Time

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
- Slave channel 2:
  - INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
  - TAUBnTDL.TAUBnTDLm = 0
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Enables start trigger detection during counting (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnTDL.TAUBnTDLm = 1
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)

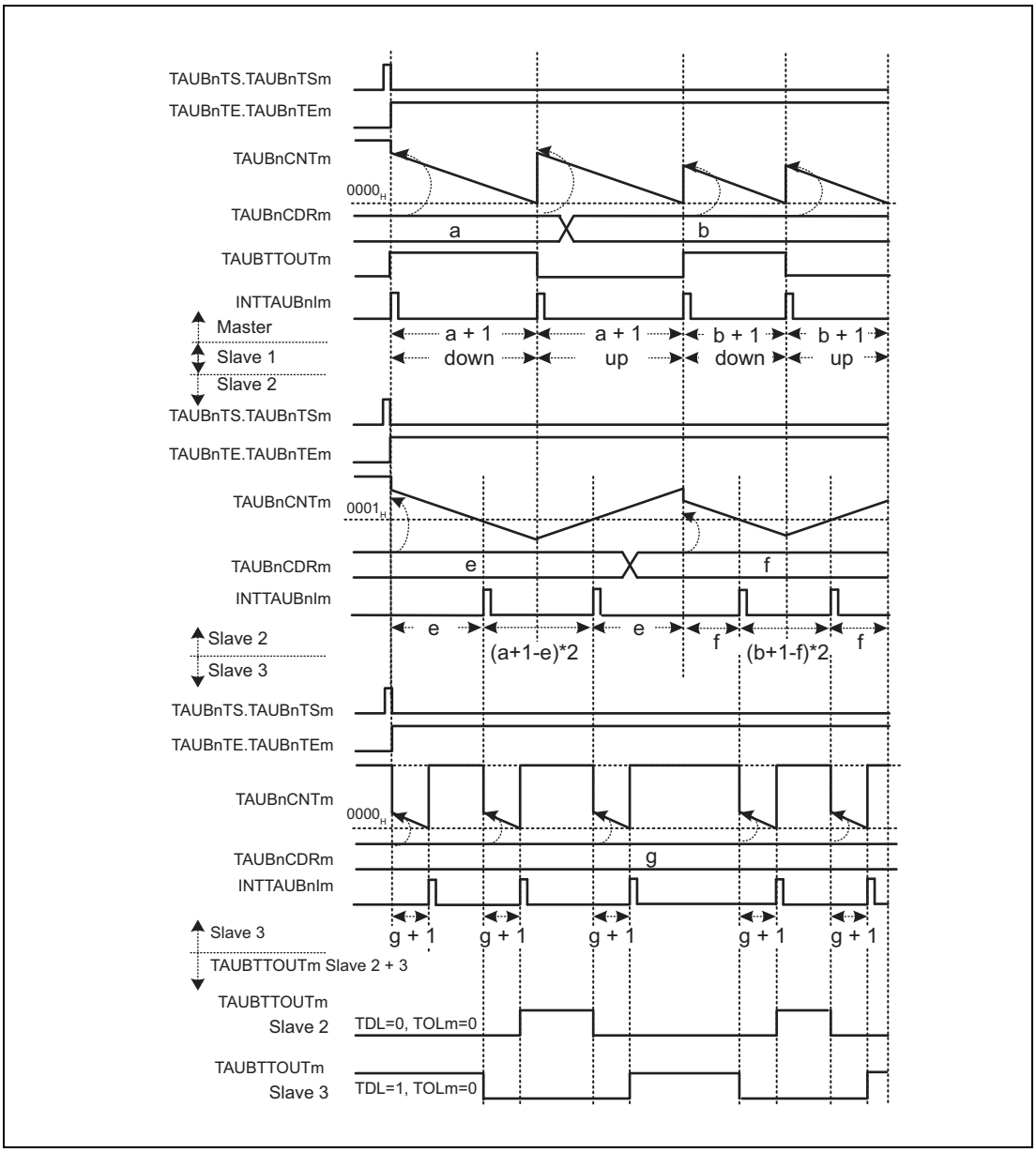


Figure 24.103 General Timing Diagram for Triangular Wave PWM Output Function with Dead Time

24.14.6.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.140 Contents of the TAUBnCMORm Register for the Master Channel of the Triangular Wave PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated and TAUBTTOUTm does not toggle at operation start 1: Generates INTTAUBnIm and toggles TAUBTTOUTm at operation start

(2) TAUBnCMURm for the master channel

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

Table 24.141 Contents of the TAUBnCMURm Register for the Master Channel of the Triangular Wave PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

Table 24.142 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	Write 0_B.
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.143 Simultaneous Rewrite Settings for the Master Channel of the Triangular Wave PWM Output Function with Dead Time

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

NOTE

If TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel that generates the simultaneous rewrite trigger signal.

24.14.6.5 Register Settings for Slave Channel 2

(1) TAUBnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.144 Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Triangular Wave PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 111_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1001_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm for slave channel 2

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

Table 24.145 Contents of the TAUBnCMURm Register for the Slave Channel 2 of the Triangular Wave PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 2

Table 24.146 Control Bit Settings for Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 1_B.
TAUBnTDL.TAUBnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase

CAUTION

Set TAUBnTDL.TAUBnTDLm exclusively to the odd channel.

(4) Simultaneous rewrite for slave channel 2

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.147 Simultaneous Rewrite Settings for Slave Channel 2 of the Triangular Wave PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

24.14.6.6 Register Settings for Slave Channel 3

(1) TAUBnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 24.148 Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Triangular Wave PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 110_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for slave channel 3

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

Table 24.149 Contents of the TAUBnCMURm Register for the Slave Channel 3 of the Triangular Wave PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 3

Table 24.150 Control Bit Settings for Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 1_B.
TAUBnTDL.TAUBnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase

CAUTION

Set TAUBnTDL.TAUBnTDLm exclusively to the even channel.

(4) Simultaneous rewrite for slave channel 3

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 24.151 Simultaneous Rewrite Settings for Slave Channel 3 of the Triangular Wave PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

24.14.6.7 Operating Procedure for Triangular Wave PWM Output Function with Dead Time

Table 24.152 Operating Procedure for Triangle PWM Output with Dead Time

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.6.4, Register Settings for the Master Channel** Slave channel 2: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.6.5, Register Settings for Slave Channel 2** Slave channel 3: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 24.14.6.6, Register Settings for Slave Channel 3** Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnIm (master) is generated when TAUBnCMORm.TAUBnMD0 is set to 1 on the master channel.
During operation	TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master channel and slave channel 2 load TAUBnCDRm and count down. When the counter of the master channel reaches 0000_H: • INTTAUBnIm (master) is generated • TAUBnCNTm (master) reloads the TAUBnCDRm value and continues count operation • TAUBnCNTm (slave 2) reloads the TAUBnCDRm value or counts in the reverse direction When TAUBnCNTm (slave 2) reaches 0001_H: • INTTAUBnIm (slave 2) is generated • TAUBnCNTm of slave channel 3 loads the TAUBnCDRm value and counts down When TAUBnCNTm of slave channel 3 = 0000_H: • INTTAUBnIm is generated
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

24.14.6.8 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

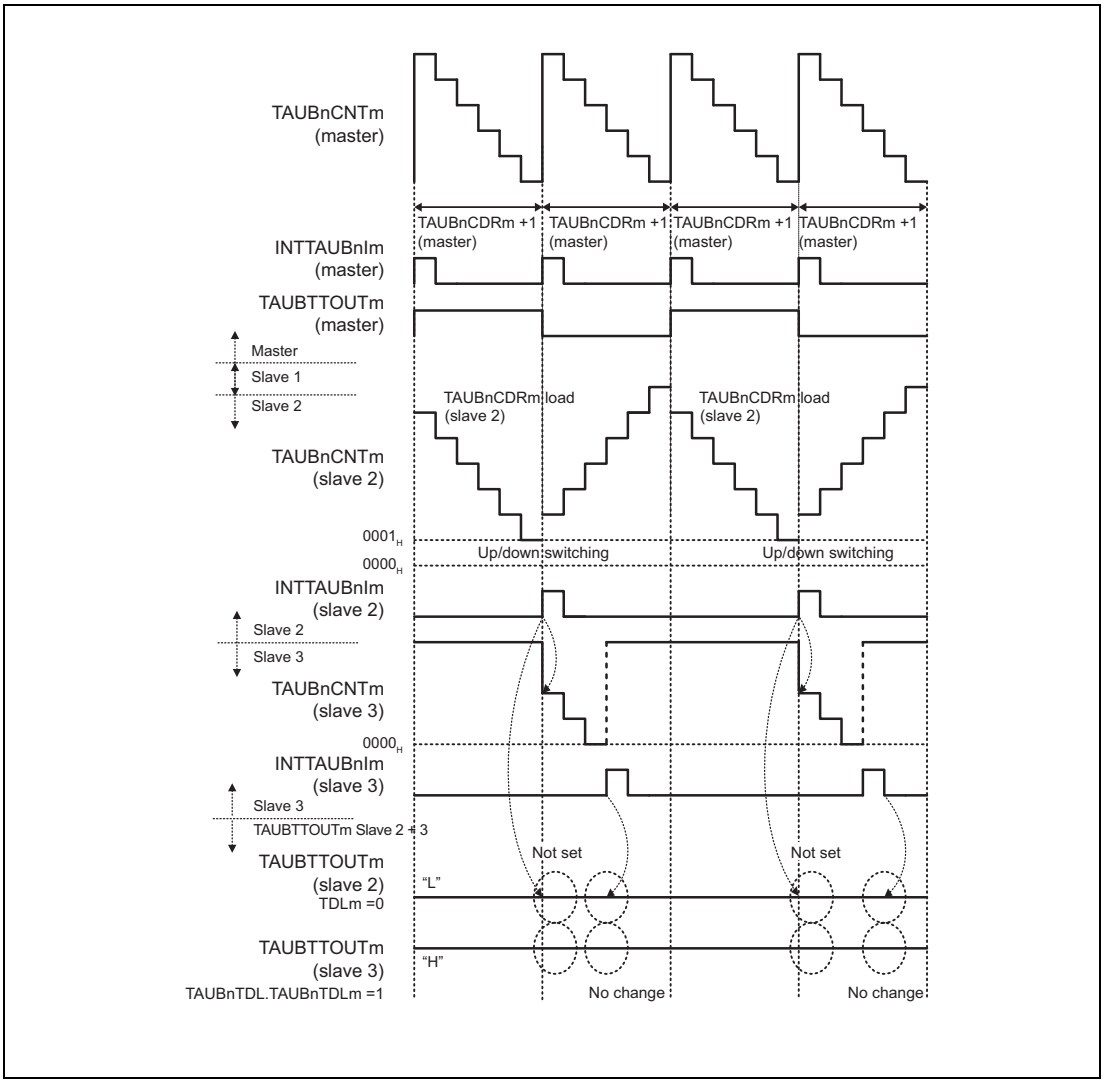


Figure 24.104 TAUBnCDRm (slave 2) ≥ TAUBnCDRm (master) + 1

- If TAUBnCDRm (slave 2) ≥ TAUBnCDRm (master), the counter of slave channel cannot reach 0000_H during counting down. Therefore TAUBTTOUTm cannot toggle, i.e. it remains at its initial state. The interrupt from slave channel 2 occurs during count up, therefore it is a reset signal.

(2) Duty cycle = 100%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

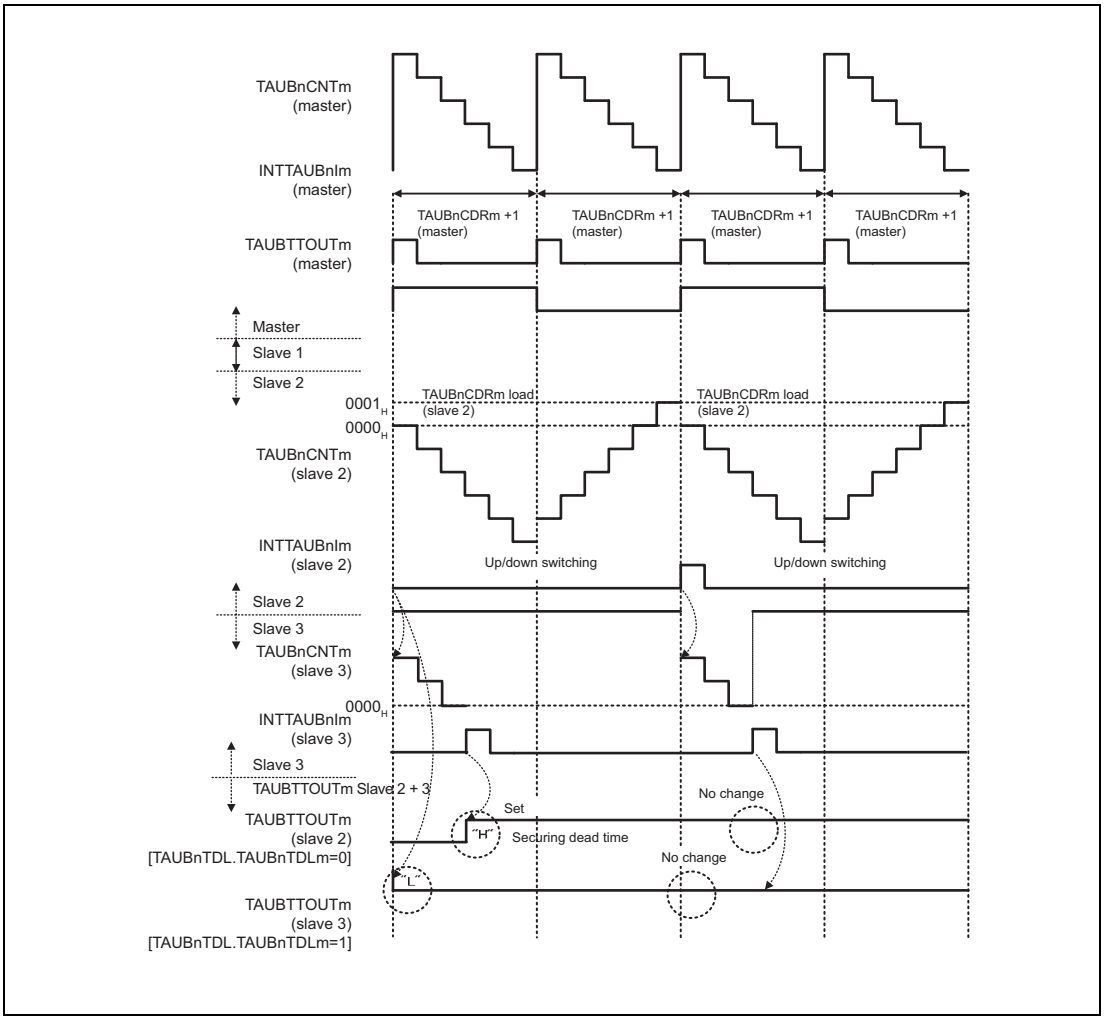


Figure 24.105 TAUBnCDRm (slave 2) = 0000_H

- If TAUBnCDRm (slave 2) = 0000_H the counter of slave channel cannot reach 0001_H while counting up and therefore cannot generate an INTTAUBnIm while counting up.
  - The set conditions for a channel in which TAUBnTDL.TAUBnTDLm = 0 are met after dead time has elapsed. TAUBTTOUTm toggles but remains in the new state because the reset conditions never occur for such a channel.
  - Slave channel 3 in the diagram above is set when the counter starts.
    However, the reset conditions for a channel in which TAUBnTDL.TAUBnTDLm = 1 never occur so TAUBTTOUTm remains in its initial state for such a slave channel.

(3) TAUBTTOUTm (slave 2) = 0% and TAUBTTOUTm (slave 3) ≥ 0%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

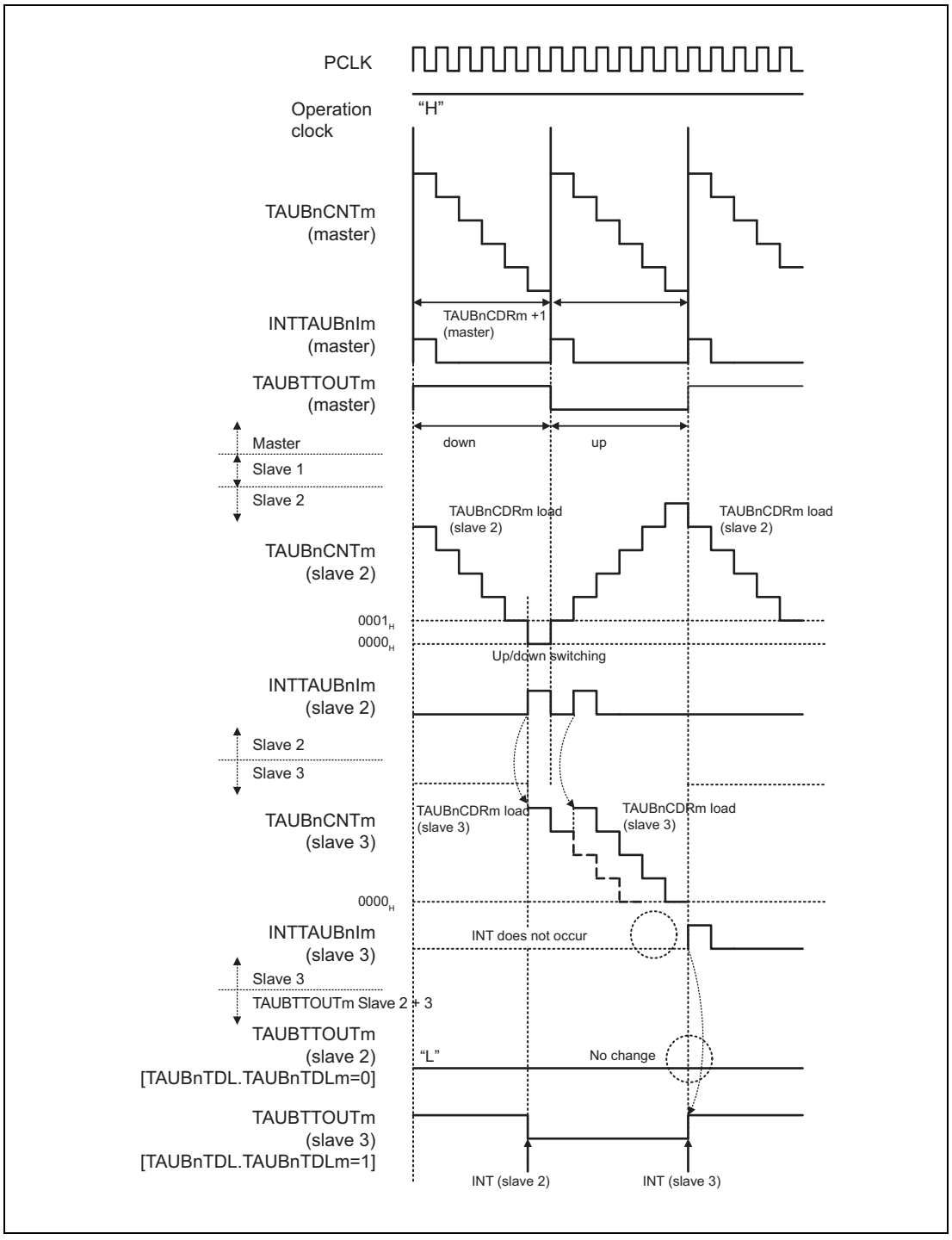


Figure 24.106 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0005_H TAUBnCDRm (slave 3) = 0004_H

- When the counter of slave channel 2 reaches 0000_H after detecting that the counter reached 0001_H, INTTAUBnIm (slave 2) is generated. The counter of slave channel 3 starts to count down.
- If another INTTAUBnIm (slave 2) is generated while the counter of slave channel 3 is still counting down, the value of TAUBnCDRm (slave 3) is reloaded and the counter restarts counting down from this value.
- In the diagram above, the first interrupt on channel 2 occurs while the counter is counting down, and the second while it is counting up.
- After the first interrupt, a slave for which TAUBnTDL.TAUBnTDLm = 0 waits for dead time to elapse before setting. However, if another interrupt occurs on slave 2, before the dead time has elapsed, the signal acts as a reset signal because the counter is counting up, meaning that a channel for which TAUBnTDL.TAUBnTDLm = 0 always remains inactive.
- TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 1 is set and reset as normal when the corresponding INTTAUBnIm is generated.

(4) TAUBTTOUTm (slave 2) > 0% and TAUBTTOUTm (slave 3) = 100%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

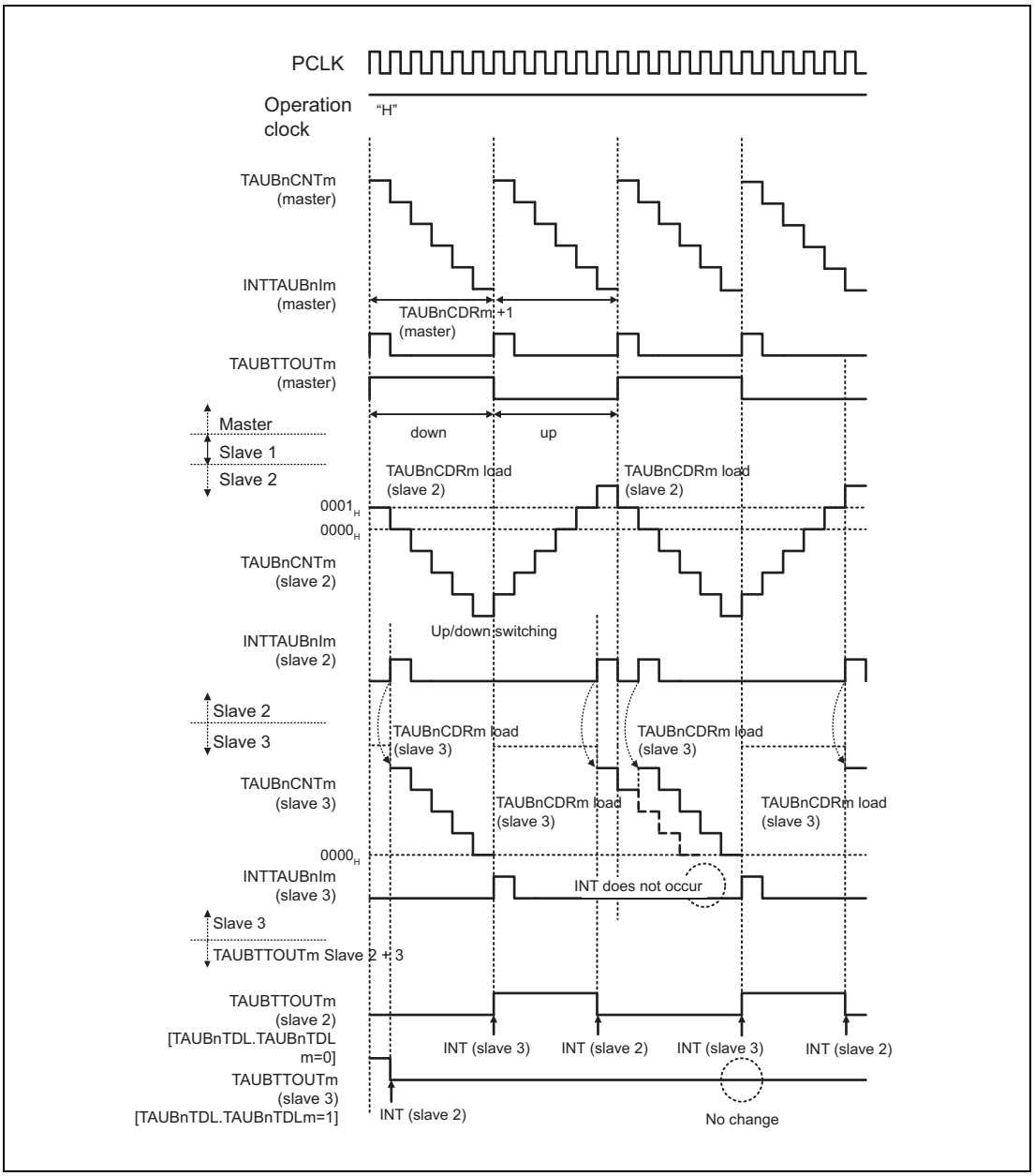


Figure 24.107 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0001_H TAUBnCDRm (slave 3) = 0004_H PWM Signal Width (negative phase) ≥ Carrier Cycle

- After the second interrupt on slave channel 2, a slave for which TAUBnTDL.TAUBnTDLm = 1 is reset after the dead time has elapsed. However if another interrupt occurs on slave 2 before the dead time has elapsed, slave 3 is restarted, and then if an interrupt on slave channel 3 is generated, the signal acts as a setting signal because the counter is counting up, meaning that a channel for which TAUBnTDL.TAUBnTDLm = 1 always remains active.
- TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 0 is set and reset successfully when the corresponding INTTAUBnIm is generated.

(5) Inhibited INTTAUBnIm to set TAUBTTOUTm positive phase period

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

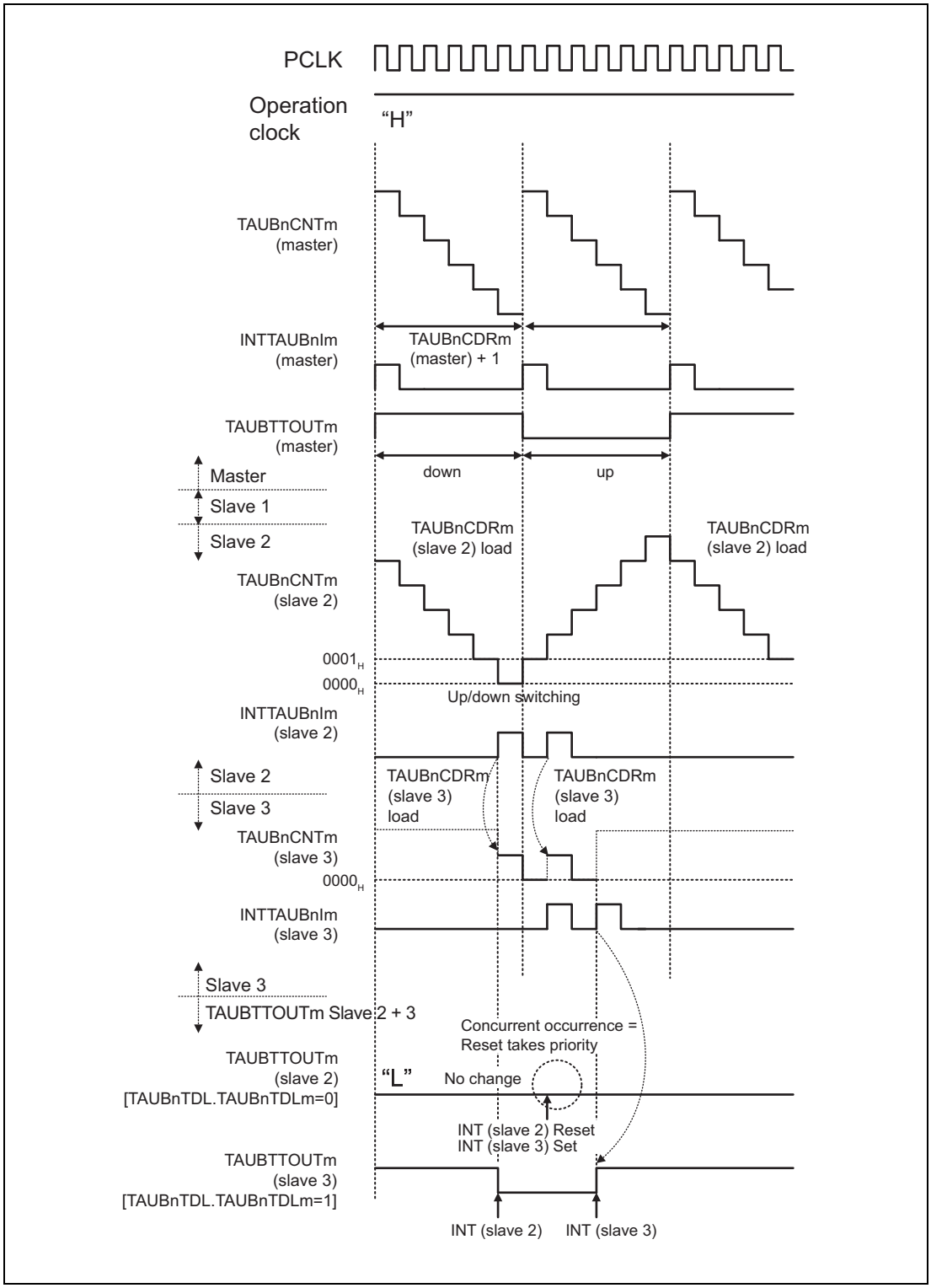


Figure 24.108 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0005_H, TAUBnCDRm (slave 3) = 0001_H PWM Signal Width (positive phase) = 0

- The counter of slave channel 3 reaches 0000_H and generates an INTTAUBnIm to set the TAUBTTOUTm of slave channel for which TAUBnTDL.TAUBnTDLm = 0 (slave channel 2 in this example).
- If slave channel 2 generates an INTTAUBnIm and resets TAUBTTOUTm simultaneously, this reset signal is given priority if TAUBnTOL.TAUBnTOLm = 0 (if TAUBnTOL.TAUBnTOLm = 1, the set signal is given priority).
- TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 0 remains in the value after reset.

(6) Inhibited INTTAUBnIm to set TAUBTTOUTm negative phase period

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

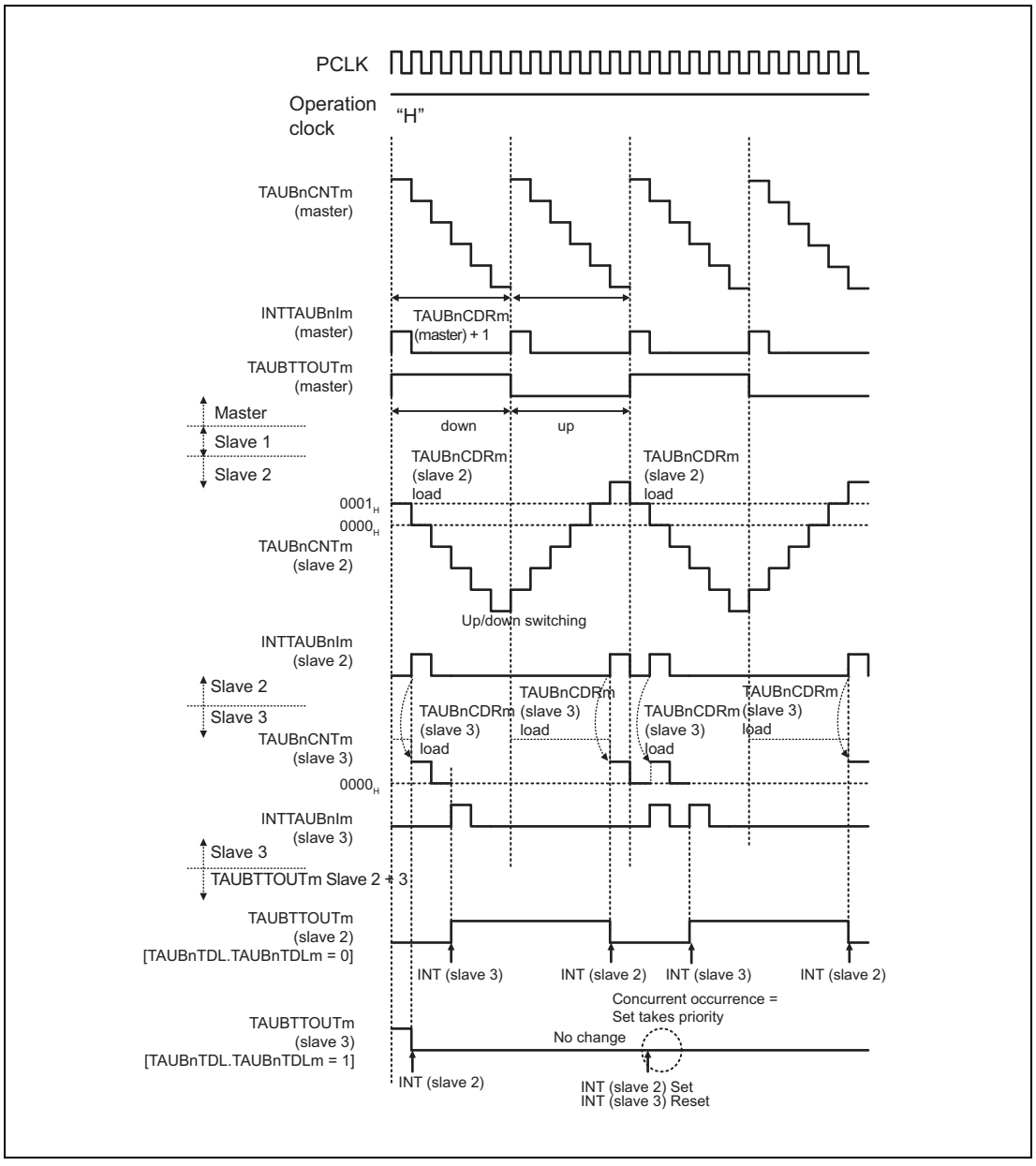


Figure 24.109 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0001_H, TAUBnCDRm (slave 3) = 0001_H PWM Signal Width (negative phase) = Carrier Cycle

- The counter of slave channel 3 reaches 0000_H and generates an INTTAUBnIm to set the TAUBTTOUTm of slave channel for which TAUBnTDL.TAUBnTDLm = 1 (slave 3 in this example).

- If slave channel 2 generates an INTTAUBnIm and resets TAUBTTOUTm simultaneously, the set signal is given priority if TAUBnTOL.TAUBnTOLm = 1 (if TAUBnTOL.TAUBnTOLm = 0, the reset signal is given priority).
- TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 1 remain in the value after reset.

(7) Slave 2 TAUBnCDRm = 0000_H (Duty cycle = 100%)

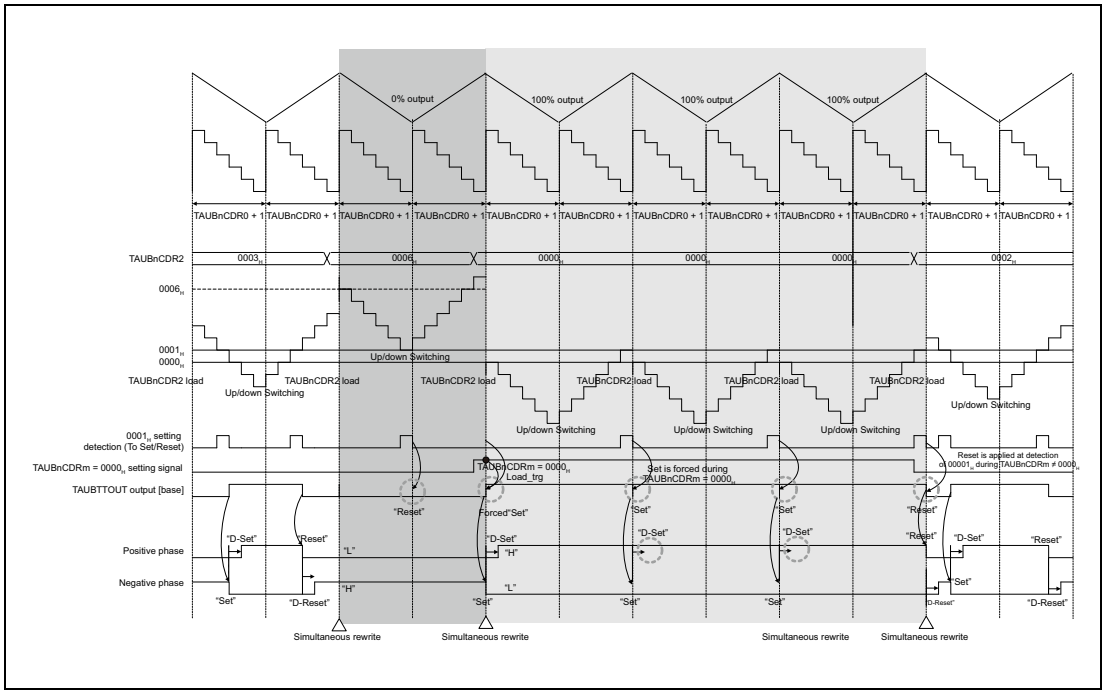


Figure 24.110 Slave 2 TAUBnCDRm = 0000_H (Duty cycle = 100%)

When rewriting (slave channel 2) TAUBnCDRm ≠ 0000_H to (slave channel 2) TAUBnCDRm = 0000_H (100% output), set the negative phase at the start of the carrier cycle, and set the positive phase after dead time is secured.

When rewriting (slave channel 2) TAUBnCDRm = 0000_H (100% output) to (slave channel 2) TAUBnCDRm ≠ 0000_H, reset the positive phase at the end of the carrier cycle, and reset the negative phase after dead time is secured.

24.14.7 A/D Conversion Trigger Output Function Type 2

24.14.7.1 Overview

Summary

This function is identical to Section 24.14.5, Triangular Wave PWM Output Function, except that TAUBTTOUTm is not output.

This function is enabled by setting the channel output mode of the slave to independent channel output mode controlled by software.

24.14.7.2 Block Diagram and General Timing Diagram

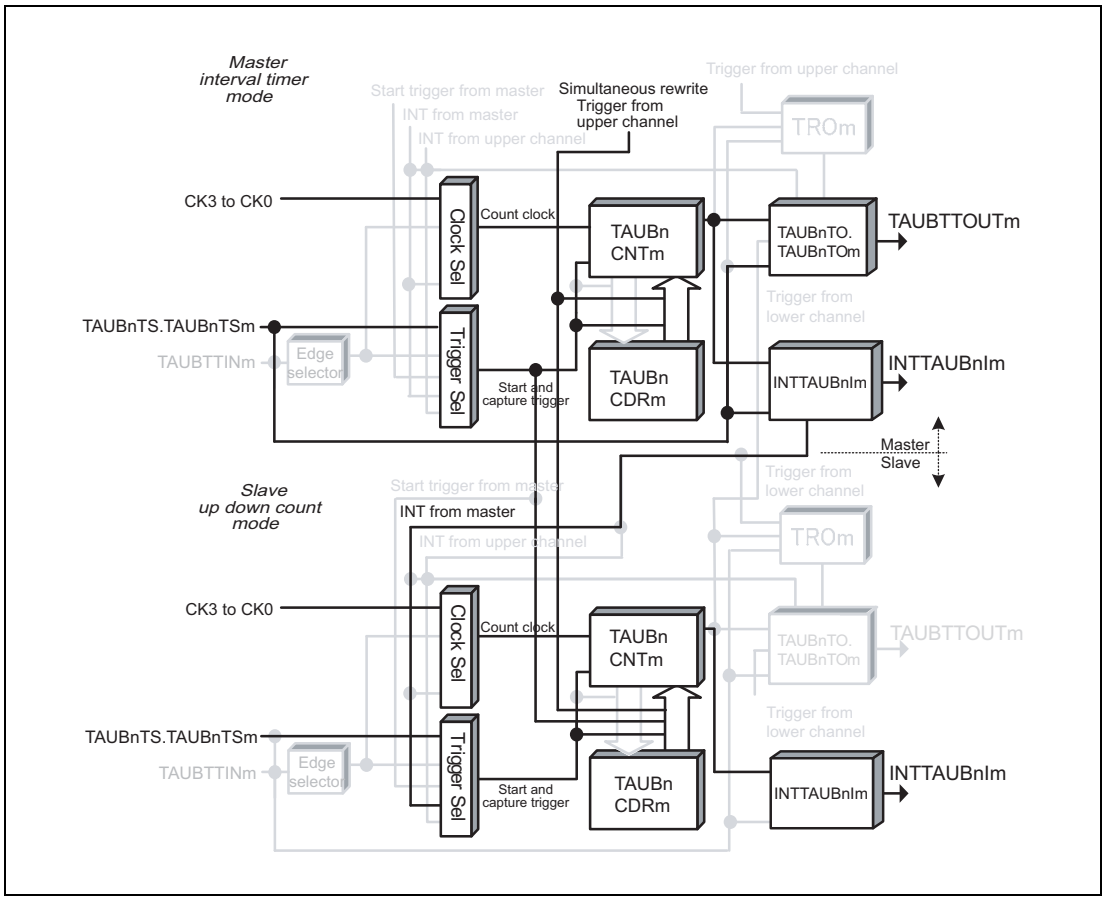


Figure 24.111 Block Diagram for A/D Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

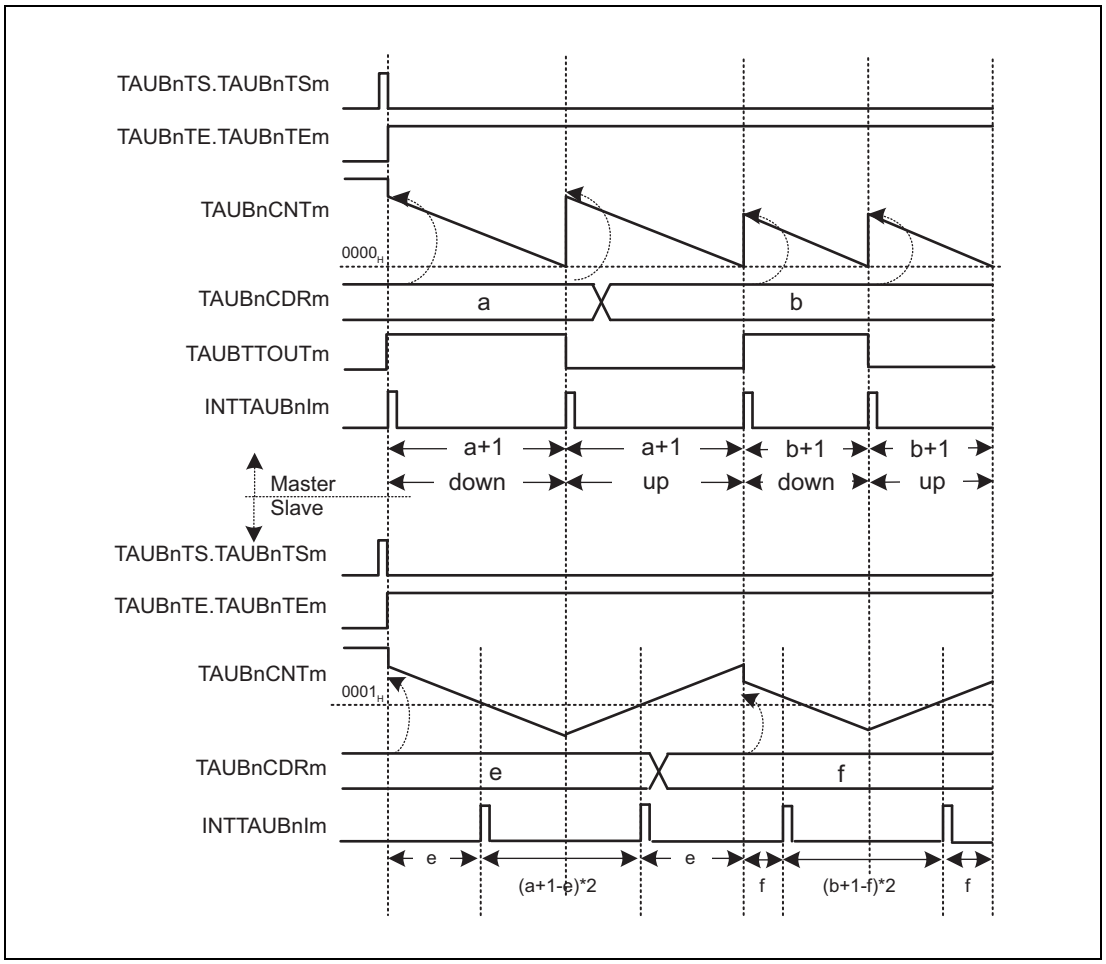


Figure 24.112 General Timing Diagram for A/D Conversion Trigger Output Function Type 2

Section 25 Timer Array Unit D (TAUD)

This section contains a generic description of the timer array unit D (TAUD).

The first part of this section describes the RH850/F1M specific features such as the number of units and the register base addresses.The remainder of the section describes the functions and registers of the TAUD.

25.1 Features of RH850/F1M TAUD

25.1.1 Number of Units and Channels

This microcontroller has the following number of TAUD units.

Table 25.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	1		
Name	TAUDn (n = 0)		

TAUDn has the following timers for the quantity of channels.

Table 25.2 TAUDn Unit Configurations and Channels

Unit Name (Channel Name) TAUDn	Channels per Unit	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
TAUD0	16	√	√	√

Table 25.3 Indices

Index	Description
n	Throughout this section, the individual TAUD units are identified by the index “n”; for example, TAUDnTOM is the TAUDn channel output mode register.
m	The TAUD has 16 channels. Throughout this section, the individual channels are identified by the index “m” (m = 0 to 15), thus a certain channel is denoted as CHm. The even numbered channels (m = 0, 2, 4, 6, 8, 10, 12, 14) are denoted as CHm_even. The odd numbered channels (m = 1, 3, 5, 7, 9, 11, 13, 15) are denoted as CHm_odd.

25.1.2 Register Base Address

TAUDn base address is listed in the following table.

TAUDn register addresses are given as offsets from the base address.

Table 25.4 Register Base Address

Base Address Name	Base Address
<TAUD0_base>	FFE2 0000_H

25.1.3 Clock Supply

The TAUDn clock supply is shown in the following table.

Table 25.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
TAUDn	PCLK	CKSCLK_IPERI1
	Register access clock	CKSCLK_IPERI1

25.1.4 Interrupt Requests

TAUDn interrupt requests are listed in the following table.

Table 25.6 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
TAUD0			
INTTAUD0I0	Channel 0 interrupt	8, 132	0
INTTAUD0I1	Channel 1 interrupt	48	15
INTTAUD0I2	Channel 2 interrupt	9, 158	64
INTTAUD0I3	Channel 3 interrupt	49	76
INTTAUD0I4	Channel 4 interrupt	10, 133	1
INTTAUD0I5	Channel 5 interrupt	50	16
INTTAUD0I6	Channel 6 interrupt	11, 134	65
INTTAUD0I7	Channel 7 interrupt	51	77
INTTAUD0I8	Channel 8 interrupt	12, 135	2
INTTAUD0I9	Channel 9 interrupt	52	17
INTTAUD0I10	Channel 10 interrupt	13, 159	66
INTTAUD0I11	Channel 11 interrupt	53	78
INTTAUD0I12	Channel 12 interrupt	14, 160	3
INTTAUD0I13	Channel 13 interrupt	54	18
INTTAUD0I14	Channel 14 interrupt	15, 161	67
INTTAUD0I15	Channel 15 interrupt	55	79

25.1.5 Reset Sources

TAUDn reset sources are listed in the following table. TAUDn is initialized by these reset sources.

Table 25.7 Reset Sources

Unit Name	Reset Source
TAUDn	All reset sources (ISORES)

25.1.6 External Input/Output Signals

External input/output signals of TAUDn are listed below.

Table 25.8 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal Name
TAUD0		
TAUDTTIN0	Channel 0 input*1	TAUD0I0
TAUDTTIN1	Channel 1 input*1	TAUD0I1
TAUDTTIN2	Channel 2 input*1	TAUD0I2
TAUDTTIN3	Channel 3 input*1	TAUD0I3
TAUDTTIN4	Channel 4 input*1	TAUD0I4
TAUDTTIN5	Channel 5 input*1	TAUD0I5
TAUDTTIN6	Channel 6 input*1	TAUD0I6
TAUDTTIN7	Channel 7 input*1	TAUD0I7
TAUDTTIN8	Channel 8 input*1	TAUD0I8
TAUDTTIN9	Channel 9 input*1	TAUD0I9
TAUDTTIN10	Channel 10 input*1	TAUD0I10
TAUDTTIN11	Channel 11 input*1	TAUD0I11
TAUDTTIN12	Channel 12 input*1	TAUD0I12
TAUDTTIN13	Channel 13 input*1	TAUD0I13
TAUDTTIN14	Channel 14 input*1	TAUD0I14
TAUDTTIN15	Channel 15 input*1	TAUD0I15
TAUDTTOUT0	Channel 0 output	TAUD0O0
TAUDTTOUT1	Channel 1 output	TAUD0O1
TAUDTTOUT2	Channel 2 output	TAUD0O2
TAUDTTOUT3	Channel 3 output	TAUD0O3
TAUDTTOUT4	Channel 4 output	TAUD0O4
TAUDTTOUT5	Channel 5 output	TAUD0O5
TAUDTTOUT6	Channel 6 output	TAUD0O6
TAUDTTOUT7	Channel 7 output	TAUD0O7
TAUDTTOUT8	Channel 8 output	TAUD0O8
TAUDTTOUT9	Channel 9 output	TAUD0O9
TAUDTTOUT10	Channel 10 output	TAUD0O10
TAUDTTOUT11	Channel 11 output	TAUD0O11
TAUDTTOUT12	Channel 12 output	TAUD0O12
TAUDTTOUT13	Channel 13 output	TAUD0O13
TAUDTTOUT14	Channel 14 output	TAUD0O14
TAUDTTOUT15	Channel 15 output	TAUD0O15

Note 1. When channel input pins are to be used, noise filters must be set. For details, see **Section 2.12, Noise Filter & Edge/Level Detector**.

CAUTION

When port P0_0 is used as TAUD0I2 or TAUD0O2, port P0_0 (the $\overline{\text{RESETOUT}}$ signal) outputs a low level during a reset and after release from the reset state.

For details, see **Section 2.11.1.1, P0_0: $\overline{\text{RESETOUT}}$**.

25.1.7 Internal Input/Output Signals

The internal input/output signals of TAUDn are listed below.

Table 25.9 Internal Input/Output Signals

Unit Signal Name	Description	Connected to
TAUDnTSSTm	Simultaneous channel start trigger input	PIC
TAUDnTUDCm (m = 0, 2, 8)	TAUD master up/down signal output	PIC

25.1.8 TAUD0 Input Selection

The output from port TAUD0Im (m = 0 to 15) can be input to TAUDTTINm (m = 0 to 15) as shown in the following figure.

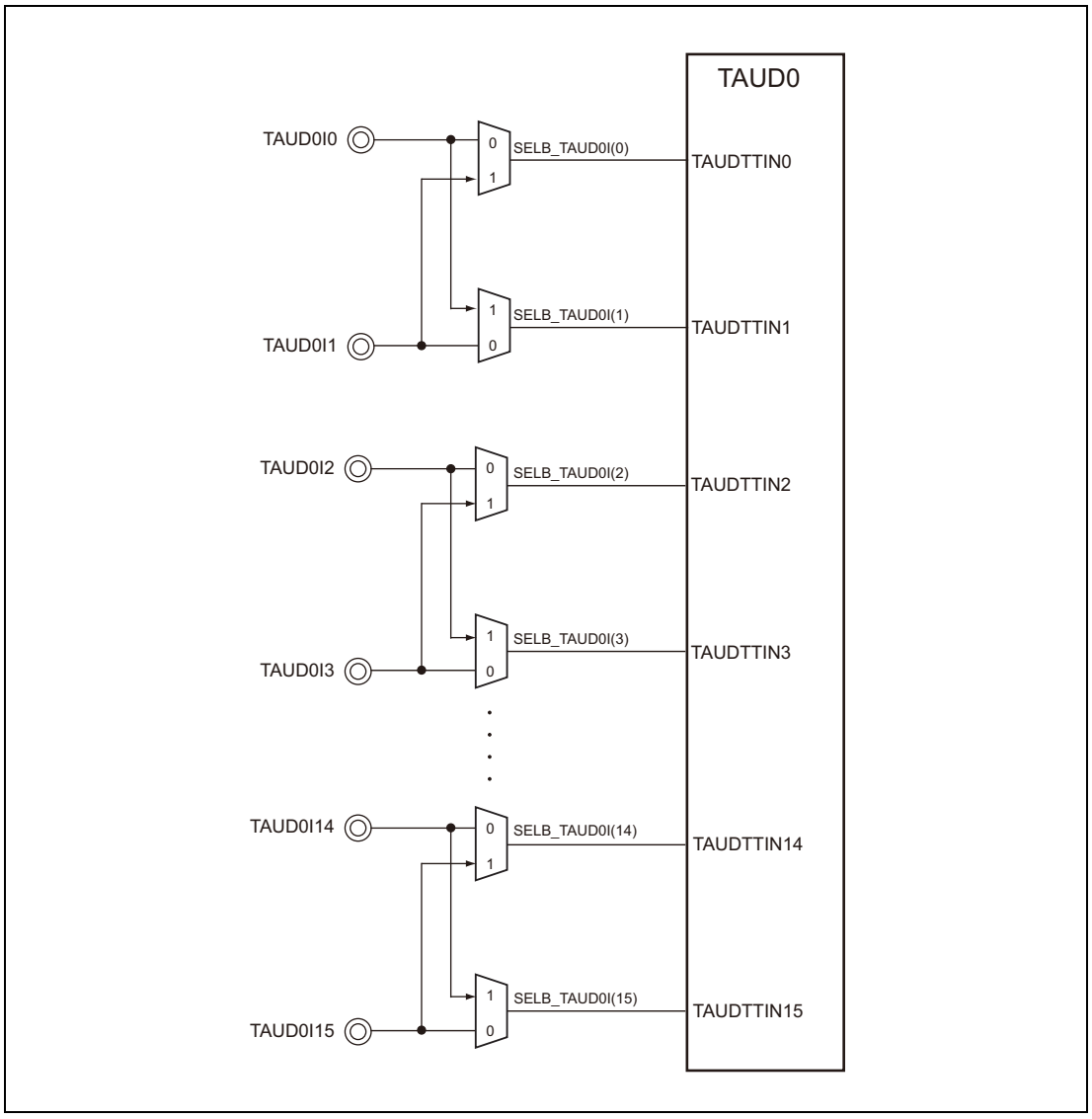


Figure 25.1 Selection of Signals Input to TAUD0

The following table shows the method of selecting input signals to several TAUD0 inputs.

Table 25.10 TAUD0 Input Selection

Input Signal	Function	Settings
TAUDTTIN [m]	Port TAUD0I[m]	SELB_TAUD0I [m] = 0
	Port TAUD0I[m + 1]	SELB_TAUD0I [m] = 1
TAUDTTIN [m + 1]	Port TAUD0I[m + 1]	SELB_TAUD0I [m + 1] = 0
	Port TAUD0I[m]	SELB_TAUD0I [m + 1] = 1

25.1.8.1 SELB_TAUD0I — TAUDTTINm Input Signal Selection Register

This register selects the input signals to several TAUDTTINm inputs.

Access: This register can be read or written in 16-bit units.

Address: FFE2 4000_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SELB_TAUD0I 15	SELB_TAUD0I 14	SELB_TAUD0I 13	SELB_TAUD0I 12	SELB_TAUD0I 11	SELB_TAUD0I 10	SELB_TAUD0I 9	SELB_TAUD0I 8	SELB_TAUD0I 7	SELB_TAUD0I 6	SELB_TAUD0I 5	SELB_TAUD0I 4	SELB_TAUD0I 3	SELB_TAUD0I 2	SELB_TAUD0I 1	SELB_TAUD0I 0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.11 SELB_TAUD0I Register Contents

Bit Position	Bit Name	Function
15 to 0	SELB_TAUD0Im	Selects the TAUDTTINm input signal

TAUD Input	Bit [m+1]	Bit [m]	Input signal
TAUDTTIN[m]	x	0	Selection of port TAUD0I[m]
	x	1	Selection of port TAUD0I[m + 1]
TAUDTTIN[m+1]	0	x	Selection of port TAUD0I[m + 1]
	1	x	Selection of port TAUD0I[m]

(m = 0, 2, 4, 6, 8, 10, 12, 14)

CAUTION

Do not change the input signal of each channel during the timer counting.

25.2 Overview

25.2.1 Functions

The TAUD has the following functions:

- 16 channels
- 16-bit counter and 16-bit data register per channel
- Independent channel operation
- Synchronous channel operation (master and slave operation)
- Generation of different types of output signal
- Real-time output
- Counter can be triggered by external signal
- Interrupt generation

The Timer Array Unit D is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 16 channels, each equipped with a 16-bit counter TAUDnCNTm and a 16-bit data register TAUDnCDRm to hold the start or compare value of the counter.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in different operation modes, either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, its operation mode and functions are not affected by those of other channels. When a channel is operated synchronously it is either a master or a slave. A master channel can have multiple slaves, and the state of one channel affects that of the other channels. For example, this means that one channel can control when another starts to count, is reset, etc.

25.2.2 Terms

In this section, the following terms are used.

Independent / synchronous channel operation

Independent or synchronous channel operation describes the dependency of channels on each other:

- If a channel operates independently of all other channels, this is called independent channel operation.
- If a channel operates depending on other channels, this is called synchronous channel operation.

Channel group

In synchronous channel operation, all channels that depend on each other are referred to as a “channel group”.

A channel group has one master channel and one or more slave channels.

Operation mode

An operation mode can be selected for every channel m. The operation mode defines the basic operation and features of a channel.

In synchronous channel operation, every channel in the channel group can operate in a different operation mode.

Examples are “Capture Mode”, “Event Count Mode”, and “Interval Timer Mode”.

Channel output mode

The channel output mode defines the operation of TAUDTTOUTm

- of a single channel (independent output operation) or
- of all channels in a channel group (synchronous output operation).

Examples are “Independent Channel Output Mode 1” and “Synchronous Channel Output Mode 2 with Dead Time Output”.

Channel operation function

The channel operation function defines the complete function and all features

- of a single channel (independent channel operation) or
- of all channels in a channel group (synchronous channel operation).

Upper / lower channel

Depending on the channel number m, a channel with a smaller number or with a larger number is referred to as “upper” or “lower” channel, respectively.

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

Example:

For channel 5, channel 3 is an upper channel and channel 9 is a lower channel.

25.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 25.12 Functional List of TAUD Operations

Operation Function	Example
Independent Channel Operation Functions	**Section 25.12**
Interval Timer Function	**Section 25.12.1**
TAUDTTINm Input Interval Timer Function	**Section 25.12.2**
Clock Divide Function	**Section 25.12.3**
External Event Count Function	**Section 25.12.4**
Delay Count Function	**Section 25.12.5**
One-Pulse Output Function	**Section 25.12.6**
TAUDTTINm Input Pulse Interval Measurement Function	**Section 25.12.7**
TAUDTTINm Input Signal Width Measurement Function	**Section 25.12.8**
TAUDTTINm Input Position Detection Function	**Section 25.12.9**
TAUDTTINm Input Period Count Detection Function	**Section 25.12.10**
TAUDTTINm Input Pulse Interval Judgment Function	**Section 25.12.11**
TAUDTTINm Input Signal Width Judgment Function	**Section 25.12.12**
Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)	**Section 25.12.13**
Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)	**Section 25.12.14**
One-Phase PWM Output Function	**Section 25.12.15**
Independent Channel Real-Time Functions	**Section 25.13**
Real-Time Output Function Type 1	**Section 25.13.1**
Real-Time Output Function Type 2	**Section 25.13.2**
Independent Channel Simultaneous Rewrite Functions	**Section 25.14**
Simultaneous Rewrite Trigger Generation Function Type 1	**Section 25.14.1**
Simultaneous Rewrite Trigger Generation Function Type 2	**Section 25.14.2**
Synchronous Channel Operation Functions	**Section 25.15**
PWM Output Function	**Section 25.15.1**
One-Shot Pulse Output Function	**Section 25.15.2**
Trigger Start PWM Output Function	**Section 25.15.3**
Delay Pulse Output Function	**Section 25.15.4**
Offset Trigger Output Function	**Section 25.15.5**
A/D Conversion Trigger Output Function Type 1	**Section 25.15.6**
Triangle PWM Output Function	**Section 25.15.7**
Triangle PWM Output Function with Dead Time	**Section 25.15.8**
A/D Conversion Trigger Output Function Type 2	**Section 25.15.9**
Interrupt Request Signals Culling Function	**Section 25.15.10**
Synchronous Non-Complementary and Complementary Modulation Output Functions	**Section 25.16**
Non-Complementary Modulation Output Function Type 1	**Section 25.16.1**
Non-Complementary Modulation Output Function Type 2	**Section 25.16.2**
Complementary Modulation Output Function	**Section 25.16.3**

25.2.4 TAUD I/O and Interrupt Request Signals

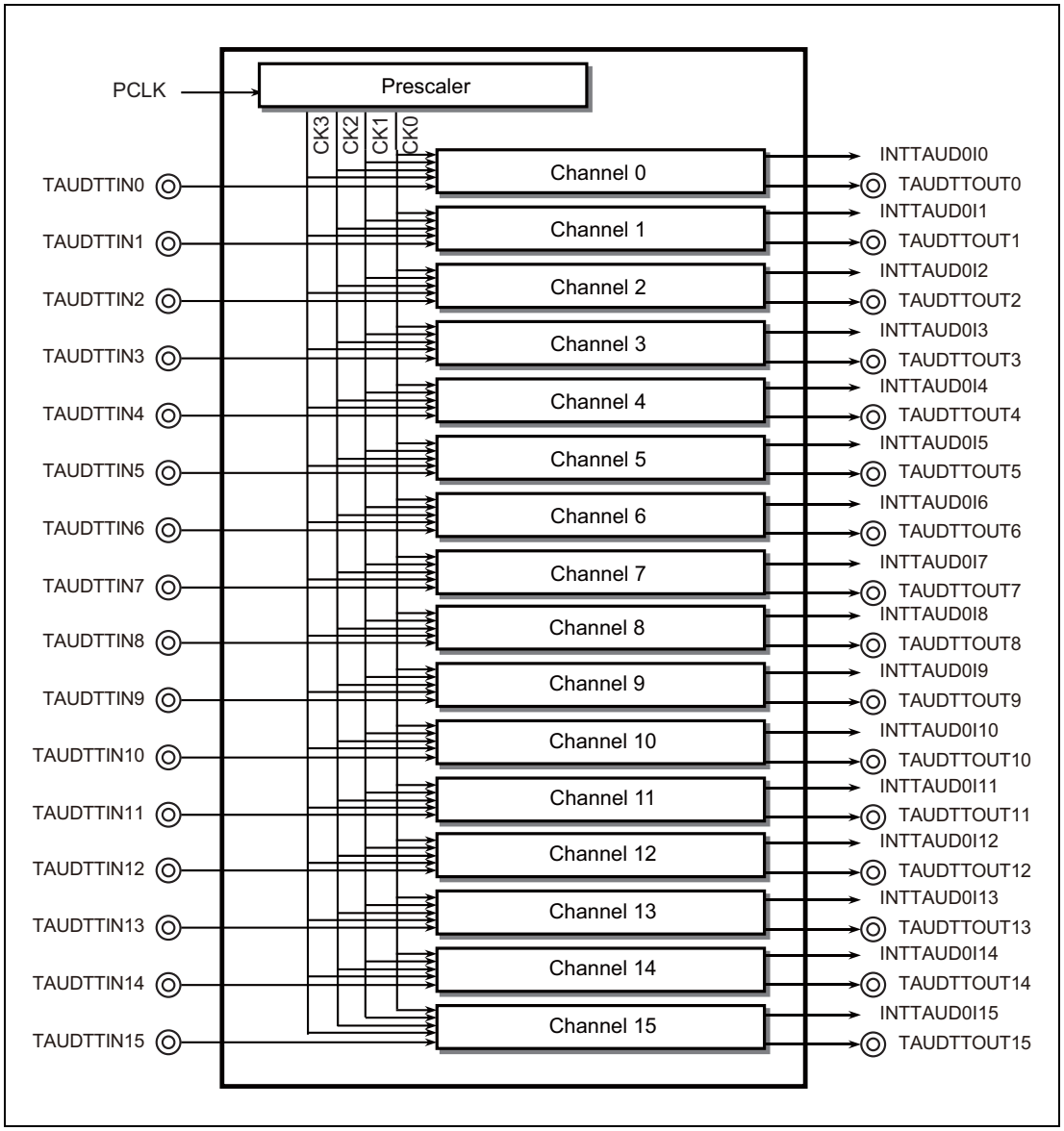


Figure 25.2 TAUD I/O and Interrupt Request Signals

25.2.5 Block Diagram

Figure 25.3 shows the main components of the TAUD.

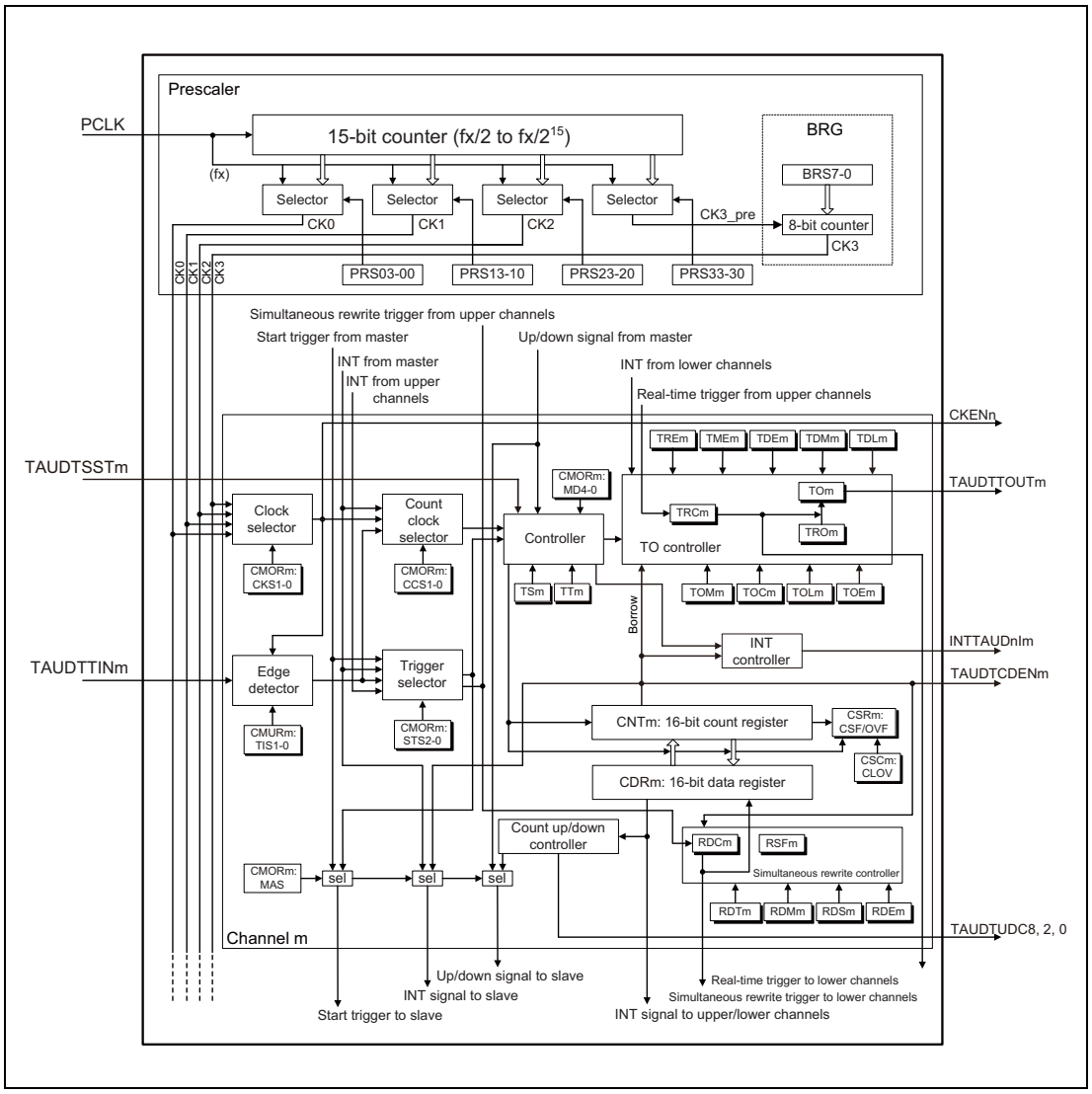


Figure 25.3 Block Diagram of the TAUD

The module name “TAUDn” has been omitted from the register names for the sake of clarity in the above figure.

25.2.6 Description of Blocks

The following describes the functional blocks:

Prescaler block

The prescaler block provides up to four clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK2 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}. The fourth count clock CK3 can be adjusted more precisely by using BRG to set an additional division factor that is not a power of 2.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- INTTAUDnIm from master channel
- TAUDTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUDnCMORm.TAUDnMD[4:0])
- Counter start enable (TAUDnTS.TAUDnTSm) and counter stop (TAUDnTT.TAUDnTTm) When counter start is enabled, status flag TAUDnTE.TAUDnTEm is set.
- Count direction (up/down) (can be controlled by master channel)

Trigger selector

Depending on the selected operation mode, the counter starts automatically when it is enabled (TAUDnTE.TAUDnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

- Synchronous channel start trigger input TAUDnTSSTm
- TAUDTTINm input signal valid edge
- INTTAUDnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUDTTOUTm generation unit.

Simultaneous rewrite controller

Simultaneous rewrite control is a function that can be used in synchronous operating modes. The data registers (TAUDnCDRm) of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUDnTO controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

25.3 Registers

25.3.1 List of Registers

TAUDn registers are listed in the following table.

For details about <TAUDn_base>, see **Section 25.1.2, Register Base Address**.

Table 25.13 List of Registers

Module	Register	Symbol	Address
TAUDn prescaler registers			
TAUDn	TAUDn prescaler clock select register	TAUDnTPS	<TAUDn_base> + 240_H
TAUDn	TAUDn prescaler baud rate setting register	TAUDnBRS	<TAUDn_base> + 244_H
TAUDn control registers			
TAUDn	TAUDn channel data register m	TAUDnCDRm	<TAUDn_base> + m × 4_H
TAUDn	TAUDn channel counter register m	TAUDnCNTm	<TAUDn_base> + 80_H + m × 4_H
TAUDn	TAUDn channel mode OS register m	TAUDnCMORm	<TAUDn_base> + 200_H + m × 4_H
TAUDn	TAUDn channel mode user register m	TAUDnCMURm	<TAUDn_base> + $C0_H$ + m × 4_H
TAUDn	TAUDn channel status register m	TAUDnCSRm	<TAUDn_base> + 140_H + m × 4_H
TAUDn	TAUDn channel status clear trigger register m	TAUDnCSCm	<TAUDn_base> + 180_H + m × 4_H
TAUDn	TAUDn channel start trigger register	TAUDnTS	<TAUDn_base> + $1C4_H$
TAUDn	TAUDn channel enable status register	TAUDnTE	<TAUDn_base> + $1C0_H$
TAUDn	TAUDn channel stop trigger register	TAUDnTT	<TAUDn_base> + $1C8_H$
TAUDn output registers			
TAUDn	TAUDn channel output enable register	TAUDnTOE	<TAUDn_base> + $5C_H$
TAUDn	TAUDn channel output register	TAUDnTO	<TAUDn_base> + 58_H
TAUDn	TAUDn channel output mode register	TAUDnTOM	<TAUDn_base> + 248_H
TAUDn	TAUDn channel output configuration register	TAUDnTOC	<TAUDn_base> + $24C_H$
TAUDn	TAUDn channel output active level register	TAUDnTOL	<TAUDn_base> + 040_H
TAUDn	TAUDn channel dead time output enable register	TAUDnTDE	<TAUDn_base> + 250_H
TAUDn	TAUDn channel dead time output mode register	TAUDnTDM	<TAUDn_base> + 254_H
TAUDn	TAUDn channel dead time output level register	TAUDnTDL	<TAUDn_base> + 54_H
TAUDn	TAUDn channel real-time output register	TAUDnTRO	<TAUDn_base> + $4C_H$
TAUDn	TAUDn channel real-time output enable register	TAUDnTRE	<TAUDn_base> + 258_H
TAUDn	TAUDn channel real-time output control register	TAUDnTRC	<TAUDn_base> + $25C_H$
TAUDn	TAUDn channel modulation output enable register	TAUDnTME	<TAUDn_base> + 50_H
TAUDn reload data registers			
TAUDn	TAUDn channel reload data enable register	TAUDnRDE	<TAUDn_base> + 260_H
TAUDn	TAUDn channel reload data mode register	TAUDnRDM	<TAUDn_base> + 264_H
TAUDn	TAUDn channel reload data control CH select register	TAUDnRDS	<TAUDn_base> + 268_H
TAUDn	TAUDn channel reload data control register	TAUDnRDC	<TAUDn_base> + $26C_H$
TAUDn	TAUDn channel reload data trigger register	TAUDnRDT	<TAUDn_base> + 44_H
TAUDn	TAUDn channel reload status register	TAUDnRSF	<TAUDn_base> + 48_H
TAUDn Emulation Register			
TAUDn	TAUDn emulation register	TAUDnEMU	<TAUDn_base> + 290_H

25.3.2 Details of TAUDn Prescaler Registers

25.3.2.1 TAUDnTPS — TAUDn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3_PRE for all channels of the PCLK prescaler. CK3 is generated by dividing CK3_PRE by the factor specified in TAUDnBRS.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 240_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnPRS3 [3:0]				TAUDnPRS2 [3:0]				TAUDnPRS1 [3:0]				TAUDnPRS0 [3:0]			
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.14 TAUDnTPS Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	TAUDnPRS3 [3:0]	Specifies CK3_PRE clock. CK3_PRE clock is an input clock to BRG unit which supplies the CK3 operation clock to all channels.

TAUDnPRS3[3:0]	CK3_PRE Clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK3 are stopped (TAUDnTE.TAUDnTEm = 0).

Table 25.14 TAUDnTPS Register Contents (2/3)

Bit Position	Bit Name	Function
11 to 8	TAUDnPRS2 [3:0]	Specifies the CK2 clock.

TAUDnPRS2[3:0]	CK2 Clock
0000_B	$PCLK/2^0$
0001_B	$PCLK/2^1$
0010_B	$PCLK/2^2$
0011_B	$PCLK/2^3$
0100_B	$PCLK/2^4$
0101_B	$PCLK/2^5$
0110_B	$PCLK/2^6$
0111_B	$PCLK/2^7$
1000_B	$PCLK/2^8$
1001_B	$PCLK/2^9$
1010_B	$PCLK/2^{10}$
1011_B	$PCLK/2^{11}$
1100_B	$PCLK/2^{12}$
1101_B	$PCLK/2^{13}$
1110_B	$PCLK/2^{14}$
1111_B	$PCLK/2^{15}$

The above bits are rewritable only when all the counters using CK2 are stopped (TAUDnTE.TAUDnTEm = 0).

7 to 4	TAUDnPRS1 [3:0]	Specifies the CK1 clock.

TAUDnPRS1[3:0]	CK1 Clock
0000_B	$PCLK/2^0$
0001_B	$PCLK/2^1$
0010_B	$PCLK/2^2$
0011_B	$PCLK/2^3$
0100_B	$PCLK/2^4$
0101_B	$PCLK/2^5$
0110_B	$PCLK/2^6$
0111_B	$PCLK/2^7$
1000_B	$PCLK/2^8$
1001_B	$PCLK/2^9$
1010_B	$PCLK/2^{10}$
1011_B	$PCLK/2^{11}$
1100_B	$PCLK/2^{12}$
1101_B	$PCLK/2^{13}$
1110_B	$PCLK/2^{14}$
1111_B	$PCLK/2^{15}$

The above bits are rewritable only when all the counters using CK1 are stopped (TAUDnTE.TAUDnTEm = 0).

Table 25.14 TAUDnTPS Register Contents (3/3)

Bit Position	Bit Name	Function
3 to 0	TAUDnPRS0 [3:0]	Specifies the CK0 clock.

TAUDnPRS0[3:0]	CK0 Clock
0000_B	$PCLK/2^0$
0001_B	$PCLK/2^1$
0010_B	$PCLK/2^2$
0011_B	$PCLK/2^3$
0100_B	$PCLK/2^4$
0101_B	$PCLK/2^5$
0110_B	$PCLK/2^6$
0111_B	$PCLK/2^7$
1000_B	$PCLK/2^8$
1001_B	$PCLK/2^9$
1010_B	$PCLK/2^{10}$
1011_B	$PCLK/2^{11}$
1100_B	$PCLK/2^{12}$
1101_B	$PCLK/2^{13}$
1110_B	$PCLK/2^{14}$
1111_B	$PCLK/2^{15}$

The above bits are rewritable only when all the counters using CK0 are stopped (TAUDnTE.TAUDnTEm = 0).

NOTE

The TAUDn clock input PCLK is specified in the first part of this section, **Section 25.1.3, Clock Supply**.

25.3.2.2 TAUDnBRS — TAUDn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUDnTPS.TAUDnPRS3[3:0].

Access: This register can be read or written in 8-bit units.

Address: <TAUDn_base> + 244_H

Value after reset: 00_H

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

Table 25.15 TAUDnBRS Register Contents

Bit Position	Bit Name	Function
7 to 0	TAUDnBRS[7:0]	Specifies a CK3_PRE clock division factor for generating CK3.

TAUDnBRS[7:0]	CK3 Clock
$0000\ 0000_B$	CK3_PRE / 1
$0000\ 0001_B$	CK3_PRE / 2
$0000\ 0010_B$	CK3_PRE / 3
$0000\ 0011_B$	CK3_PRE / 4
...	...
$1111\ 1110_B$	CK3_PRE / 255
$1111\ 1111_B$	CK3_PRE / 256

25.3.3 Details of TAUDn Control Registers

25.3.3.1 TAUDnCDRm — TAUDn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUDnCMORm.TAUDnMD[4:1].

Access: This register can be read or written in 16-bit units.
- When this register functions as a capture register, only reading is possible. Write operation is ignored.
- When this register functions as a compare register, reading and writing is possible.

Address: <TAUDn_base> + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.16 TAUDnCDRm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnCDR [15:0]	Data register for capture/compare values

25.3.3.2 TAUDnCNTm — TAUDn Channel Counter Register

This is a channel m counter register.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUDn_base> + 80_H + m × 4_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.17 TAUDnCNTm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnCNT [15:0]	16-bit counter value

A read value depends on a counter value, a changed operating mode, TAUDnTS.TAUDnTSm or TAUDnTT.TAUDnTTm bit value.

The initial read value of the counter depends on an operating mode and how the counter is stopped.

- Stop by a reset
- Stop by a counter stop trigger (TAUDnTT.TAUDnTTm = 1)

Table 25.18 lists the initial counter read values after the counter is stopped (TAUDnTE.TAUDnTEm = 0) and re-enabled (TAUDnTS.TAUDnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUDnTS.TAUDnTSm = 1) with the counter waiting for a start trigger.

Table 25.18 TAUDnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUDnCNTm Value		
		Start Value*1	After Stop Trigger	After One Count
Interval timer mode	Count down	$FFFF_H$	Stop value	—
Judge mode	Count down	$FFFF_H$	Stop value	—
Capture mode	Count up	0000_H	Stop value	—
Event count mode	Count down	$FFFF_H$	Stop value	—
One-count mode	Count down	$FFFF_H$	Stop value	Stop value
Capture and one-count mode	Count up	0000_H	Stop value	Capture value + 1 (TAUDnCDRm)
Judge and one-count mode	Count down	$FFFF_H$	Stop value	TAUDnCNTm value – 1
Count-up/-down mode	Count down/up	$FFFF_H$	Stop value	—
Pulse one-count mode	Count down	$FFFF_H$	Stop value	0000_H
Count capture mode	Count up	0000_H	Stop value	—
Gate count mode	Count down	$FFFF_H$	Stop value	Stop value
Capture and gate count mode	Count up	0000_H	Stop value	Stop value

Note 1. The value set for TAUDnCNTm when the operating mode is changed after a reset is deasserted.

25.3.3.3 TAUDnCMORm — TAUDn Channel Mode OS Register

This register controls channel m operation.

Access: This register can be read or written in 16-bit units. Writable only when the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 200_H + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.19 TAUDnCMORm Register Contents (1/3)

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Selects an operation clock. An operation clock is used for the TAUDTTINm input edge detection circuit. Setting of TAUDnCMORm.TAUDnCCS[1:0] bits also allows the operation clock to serve as the TAUDnCNTm counter clock.

TAUDn CKS1	TAUDn CKS0	Selection of Operation Clock
0	0	CK0
0	1	CK1
1	0	CK2
1	1	CK3

Bit Position	Bit Name	Function
13, 12	TAUDnCCS[1:0]	Selects a count clock for TAUDnCNTm counter.

TAUDn CCS1	TAUDn CCS0	Selection of Count Clock
0	0	Operation clock specified by TAUDnCMORm.TAUDnCKS[1:0]
0	1	Valid edge of TAUDTTINm input signal
1	0	Setting prohibited
1	1	INTTAUDnIm signal of master channel

Bit Position	Bit Name	Function
11	TAUDnMAS	Specifies whether the channel is a master channel or slave channel during synchronous channel operation. 0: Slave 1: Master This bit setting is valid only for even channels (CHm_even). Odd channels (CHm_odd) are fixed to 0.

Table 25.19 TAUDnCMORm Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUDnSTS[2:0]</td><td>Selects an external start trigger.
<table>
<tr><th>TAUDnSTS2</th><th>TAUDnSTS1</th><th>TAUDnSTS0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of TAUDTTINm input signal, which is specified by TAUDnCMURm.TAUDnTIS[1:0].</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of TAUDTTINm input signal is used as a start trigger and the opposite edge as a stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Triggers simultaneous rewrite.</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INTTAUDnIm is the start trigger of master channel</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>INTTAUDnIm of upper channel (m – 1) is the start trigger regardless of master setting</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Dead time output signal of TAUDTTOUTm generating unit</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Up/down output trigger signal of master channel</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>TAUDnCOS[1:0]</td><td>Specifies the timing for updating capture register TAUDnCDRm and overflow flag TAUDnCSRm.TAUDnOVF of channel m.
These bits are valid only when channel m is in capture mode or capture one-count mode.
<table>
<tr><th>TAUDnCOS1</th><th>TAUDnCOS0</th><th>TAUDnCDRm</th><th>TAUDnCSRm.TAUDnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of an effective edge of TAUDTTINm input.</td><td>Updated (cleared or set) by detecting an effective edge of TAUDTTINm input:
• If a counter overflow has occurred since the last detection of an effective edge, set TAUDnCSRm.TAUDnOVF.
• If no counter overflow has occurred since the last detection of an effective edge, clear TAUDnCSRm.TAUDnOVF.</td></tr>
<tr><td>0</td><td>1</td><td>Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated upon detection of valid edge of TAUDTTINm input and at the occurrence of counter overflow:
• Detection of valid edge of TAUDTTINm input: Counter value is written into TAUDnCDRm.
• Occurrence of overflow: $FFFF_H$ is loaded into TAUDnCDRm. The next detection of valid edge of TAUDTTINm input is ignored.</td><td>Not set</td></tr>
<tr><td>1</td><td>1</td><td>Set when a counter overflow occurs, and cleared when TAUDnCSCm.TAUDnCLOV is set to 1.</td></tr>
</table></td></tr>
<tr><td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 25.19 TAUDnCMORm Register Contents (3/3)

Bit Position	Bit Name	Function
4 to 0	TAUDnMD[4:0]	Specifies an operating mode.

TAUDnMD4	TAUDnMD3	TAUDnMD2	TAUDnMD1	TAUDnMD0	Functional Description
0	0	0	0	1/0	Interval timer mode
0	0	0	1	1/0	Judge mode
0	0	1	0	1/0	Capture mode
0	0	1	1	0	Event count mode
0	1	0	0	1/0	One-count mode
0	1	0	1	1/0	Setting prohibited
0	1	1	0	0	Capture and one-count mode
0	1	1	1	1/0	Judge and one-count mode
1	0	0	0	0	Setting prohibited
1	0	0	1	0	Count-up/-down mode
1	0	1	0	1/0	Pulse one-count mode
1	0	1	1	1/0	Count capture mode
1	1	0	0	0	Gate count mode
1	1	0	1	0	Capture and gate count mode

Mode	Role of TAUDnMD0 Bit
Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUDnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUDnIm is not generated. 1: INTTAUDnIm is generated.
Event count mode Count-up/-down mode	This bit should be set to 0 (the INTTAUDnIm signal is not output at the beginning of count operation).
One-count mode Pulse one-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. **CAUTION** • INTTAUDnIm signal is not output at the beginning of count operation in one-count mode. • INTTAUDnIm signal is output at the beginning of count operation in pulse one-count mode.
Gate count mode	This bit should be set to 0 (disables start trigger detection during counting).
Capture and one-count mode Capture and gate count mode	This bit should be set to 0. **CAUTION** INTTAUDnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.
Judge mode Judge and one-count mode	Specifies INTTAUDnIm output timing. 0: When TAUDnCNTm ≤ TAUDnCDRm 1: When TAUDnCNTm > TAUDnCDRm

25.3.3.4 TAUDnCMURm — TAUDn Channel Mode User Register

This register specifies a type of valid edge detection used for TAUDTTINm input.

Access: This register can be read or written in 8-bit units.

Address: <TAUDn_base> + $C0_H$ + m × 4_H

Value after reset: 00_H

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

Table 25.20 TAUDnCMURm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	Specifies a valid edge of TAUDTTINm input signal.

TAUDn TIS1	TAUDn TIS0	Functional Description
0	0	Falling edge
0	1	Rising edge
1	0	Detection of rising and falling edges (selects low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge
1	1	Detection of rising and falling edges (selects high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge

- Edge detection of TAUDTTINm input signal is based on the operation clock selected by TAUDnCMORm.TAUDnCKS[1:0].

25.3.3.5 TAUDnCSRm — TAUDn channel status register

This register indicates the count direction and overflow status of channel m counter.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUDn_base> + 140_H + m × 4_H

Value after reset: 00_H

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

Table 25.21 TAUDnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	TAUDnCSF	Indicates a count direction. 0: Count-up 1: Count-down The read value of this bit is valid only in the following mode: • Count-up/-down mode
0	TAUDnOVF	Indicates counter overflow status. 0: No overflow occurs. 1: Overflow occurs. This bit is used only in the following modes: • Capture mode • Capture and one-count mode The function of this bit depends on the setting of control bit TAUDnCMORm.TAUDnCOS[1:0].

25.3.3.6 TAUDnCSCm — TAUDn Channel Status Clear Register

This is a trigger register for clearing the overflow flag TAUDnCSRm.TAUDnOVF of channel m.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUDn_base> + 180_H + m × 4_H

Value after reset: 00_H

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

Table 25.22 TAUDnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAUDnCLOV	0: No function 1: Clears overflow flag TAUDnCSRm.TAUDnOVF.

25.3.3.7 TAUDnTS — TAUDn Channel Start Trigger Register

This register enables the counter operation of each channel.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H.

Address: <TAUDn_base> + $1C4_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTS15	TAUDnTS14	TAUDnTS13	TAUDnTS12	TAUDnTS11	TAUDnTS10	TAUDnTS09	TAUDnTS08	TAUDnTS07	TAUDnTS06	TAUDnTS05	TAUDnTS04	TAUDnTS03	TAUDnTS02	TAUDnTS01	TAUDnTS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 25.23 TAUDnTS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTSm	Enables the counter operation of channel m. 0: No function 1: Enables the counter operation and sets TAUDnTE.TAUDnTEm to 1. The counter operation is only enabled when TAUDnTE.TAUDnTEm is set to 1. Whether counting is started or not depends on a selected operating mode.

25.3.3.8 TAUDnTE — TAUDn Channel Enable Status Register

This register enables/disables a counter operation.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUDn_base> + $1C0_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTE15	TAUDnTE14	TAUDnTE13	TAUDnTE12	TAUDnTE11	TAUDnTE10	TAUDnTE09	TAUDnTE08	TAUDnTE07	TAUDnTE06	TAUDnTE05	TAUDnTE04	TAUDnTE03	TAUDnTE02	TAUDnTE01	TAUDnTE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.24 TAUDnTE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTEm	Enables/disables the counter operation of channel m. 0: Disables counter operation. 1: Enables counter operation. This bit is set to 1 when trigger input of TAUDnTSSTm (synchronous channel start trigger signal) is detected or when TAUDnTS.TAUDnTSm is set to 1. This bit is set to 0 when TAUDnTT.TAUDnTTm is set to 1.

25.3.3.9 TAUDnTT — TAUDn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H.

Address: <TAUDn_base> + $1C8_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTT15	TAUDnTT14	TAUDnTT13	TAUDnTT12	TAUDnTT11	TAUDnTT10	TAUDnTT09	TAUDnTT08	TAUDnTT07	TAUDnTT06	TAUDnTT05	TAUDnTT04	TAUDnTT03	TAUDnTT02	TAUDnTT01	TAUDnTT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 25.25 TAUDnTT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTTm	Stops the counter operation of channel m. 0: No function 1: Stops the counter operation and resets TAUDnTE.TAUDnTEm. TAUDnCNTm, TAUDnTO.TAUDnTOm, and TAUDTTOUTm retain the values provided before the counter is stopped.

25.3.4 Details of TAUDn Simultaneous Rewrite Registers

25.3.4.1 TAUDnRDE — TAUDn Channel Reload Data Enable Register

This register enables/disables simultaneous rewrite of TAUDnCDRm/TAUDnTOLm data register.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 260_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDE15	TAUDn RDE14	TAUDn RDE13	TAUDn RDE12	TAUDn RDE11	TAUDn RDE10	TAUDn RDE09	TAUDn RDE08	TAUDn RDE07	TAUDn RDE06	TAUDn RDE05	TAUDn RDE04	TAUDn RDE03	TAUDn RDE02	TAUDn RDE01	TAUDn RDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.26 TAUDnRDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDEm	Enables/disables simultaneous rewrite of the data register of channel m. 0: Disables simultaneous rewrite 1: Enables simultaneous rewrite

25.3.4.2 TAUDnRDS — TAUDn Channel Reload Data Control Channel Select Register

This register selects a channel that controls simultaneous rewrite.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 268_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDS15	TAUDn RDS14	TAUDn RDS13	TAUDn RDS12	TAUDn RDS11	TAUDn RDS10	TAUDn RDS09	TAUDn RDS08	TAUDn RDS07	TAUDn RDS06	TAUDn RDS05	TAUDn RDS04	TAUDn RDS03	TAUDn RDS02	TAUDn RDS01	TAUDn RDS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.27 TAUDnRDS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDSm	Selects a channel that controls a simultaneous rewrite trigger. 0: Master channel 1: Another upper channel

25.3.4.3 TAUDnRDM — TAUDn Channel Reload Data Mode Register

This register selects the timing for generating a simultaneous rewrite control signal.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 264_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDM15	TAUDn RDM14	TAUDn RDM13	TAUDn RDM12	TAUDn RDM11	TAUDn RDM10	TAUDn RDM09	TAUDn RDM08	TAUDn RDM07	TAUDn RDM06	TAUDn RDM05	TAUDn RDM04	TAUDn RDM03	TAUDn RDM02	TAUDn RDM01	TAUDn RDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.28 TAUDnRDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDMm	Selects the timing for generating a simultaneous rewrite trigger signal. 0: When the master channel counter starts to count 1: At the peak of cycle of triangular wave These bit settings are applied only when TAUDnRDE.TAUDnRDEm = 1 and TAUDnRDS.TAUDnRDSm = 0.

25.3.4.4 TAUDnRDC — TAUDn Channel Reload Data Control Register

This register specifies a channel which generates an INTTAUDnIm signal to trigger simultaneous rewrite.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + $26C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDC15	TAUDn RDC14	TAUDn RDC13	TAUDn RDC12	TAUDn RDC11	TAUDn RDC10	TAUDn RDC09	TAUDn RDC08	TAUDn RDC07	TAUDn RDC06	TAUDn RDC05	TAUDn RDC04	TAUDn RDC03	TAUDn RDC02	TAUDn RDC01	TAUDn RDC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.29 TAUDnRDC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDCm	Specifies whether the channel generates a simultaneous rewrite trigger signal or not. 0: Does not operate as a simultaneous rewrite trigger channel. 1: Operates as a simultaneous rewrite trigger channel. These bit settings are applied only when TAUDnRDE.TAUDnRDEm = 1 and TAUDnRDS.TAUDnRDSm = 1.

25.3.4.5 TAUDnRDT — TAUDn Channel Reload Data Trigger Register

This register triggers a simultaneous rewrite enabling state.

Access: This register is a write-only register that can be written in 16-bit units. It is always read as 0000_H.

Address: <TAUDn_base> + 044_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RDT15	TAUDn RDT14	TAUDn RDT13	TAUDn RDT12	TAUDn RDT11	TAUDn RDT10	TAUDn RDT09	TAUDn RDT08	TAUDn RDT07	TAUDn RDT06	TAUDn RDT05	TAUDn RDT04	TAUDn RDT03	TAUDn RDT02	TAUDn RDT01	TAUDn RDT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 25.30 TAUDnRDT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRDTm	Triggers a simultaneous rewrite enabling state. 0: No function 1: The simultaneous rewrite enabling flag (TAUDnRSFm) is set to 1. The system waits for a simultaneous rewrite trigger. These bits only apply when: • TAUDnRDE.TAUDnRDEm = 1

25.3.4.6 TAUDnRSF — TAUDn Channel Reload Status Register

This flag register indicates simultaneous rewrite status.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAUDn_base> + 048_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn RSF15	TAUDn RSF14	TAUDn RSF13	TAUDn RSF12	TAUDn RSF11	TAUDn RSF10	TAUDn RSF09	TAUDn RSF08	TAUDn RSF07	TAUDn RSF06	TAUDn RSF05	TAUDn RSF04	TAUDn RSF03	TAUDn RSF02	TAUDn RSF01	TAUDn RSF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25.31 TAUDnRSF Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnRSFm	Indicates simultaneous rewrite status. 0: Indicates that simultaneous rewrite has been completed due to the generation of simultaneous rewrite trigger. 1: Indicates that the system waits for a simultaneous rewrite trigger in the simultaneous rewrite enabling state (TAUDnRDTm = 1).

25.3.5 Details of TAUDn Output Registers

25.3.5.1 TAUDnTOE — TAUDn Channel Output Enable Register

This register enables/disables the independent channel output mode controlled by software.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + $5C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTOE15	TAUDnTOE14	TAUDnTOE13	TAUDnTOE12	TAUDnTOE11	TAUDnTOE10	TAUDnTOE09	TAUDnTOE08	TAUDnTOE07	TAUDnTOE06	TAUDnTOE05	TAUDnTOE04	TAUDnTOE03	TAUDnTOE02	TAUDnTOE01	TAUDnTOE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.32 TAUDnTOE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOEm	Enables/disables the independent channel output function. 0: Disables the independent timer output function (controlled by software). 1: Enables the independent timer output function.

25.3.5.2 TAUDnTO — TAUDn Channel Output Register

This register specifies and reads a TAUDTTOUTm level.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 58_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTO15	TAUDnTO14	TAUDnTO13	TAUDnTO12	TAUDnTO11	TAUDnTO10	TAUDnTO09	TAUDnTO08	TAUDnTO07	TAUDnTO06	TAUDnTO05	TAUDnTO04	TAUDnTO03	TAUDnTO02	TAUDnTO01	TAUDnTO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.33 TAUDnTO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOm	Specifies and reads a TAUDTTOUTm level. 0: Low level 1: High level Only TAUDnTOm bits for which Independent Channel Output function is disabled (TAUDnTOEm = 0) can be written.

25.3.5.3 TAUDnTOM — TAUDn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 248_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOM15	TAUDn TOM14	TAUDn TOM13	TAUDn TOM12	TAUDn TOM11	TAUDn TOM10	TAUDn TOM09	TAUDn TOM08	TAUDn TOM07	TAUDn TOM06	TAUDn TOM05	TAUDn TOM04	TAUDn TOM03	TAUDn TOM02	TAUDn TOM01	TAUDn TOM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.34 TAUDnTOM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOMm	Specifies an output mode. 0: Independent channel operation 1: Synchronous channel operation

25.3.5.4 TAUDnTOC — TAUDn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUDnTOMm.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + $24C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOC15	TAUDn TOC14	TAUDn TOC13	TAUDn TOC12	TAUDn TOC11	TAUDn TOC10	TAUDn TOC09	TAUDn TOC08	TAUDn TOC07	TAUDn TOC06	TAUDn TOC05	TAUDn TOC04	TAUDn TOC03	TAUDn TOC02	TAUDn TOC01	TAUDn TOC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.35 TAUDnTOC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOCm	Specifies an output mode. 0: Operating mode 1 1: Operating mode 2 As listed below, the output mode depends on the setting of TAUDnTOM.TAUDnTOMm.

TAUDn TOMm	TAUDn TOCm	Functional Description
0	0	Toggle mode: Toggle operation is conducted when INTTAUDnIm occurs.
0	1	Set/reset mode: Set when INTTAUDnIm occurs at the beginning of count operation, and reset when INTTAUDnIm is caused by detection of a match between TAUDnCNTm and TAUDnCDRm.
1	0	Synchronous channel operating mode 1: Set when INT occurs on master channels, and reset when INT occurs on slave channels.
1	1	Synchronous channel operating mode 2: Set when INTTAUDnIm occurs in count-down status, and reset when INTTAUDnIm occurs in count-up status.

25.3.5.5 TAUDnTOL — TAUDn Channel Output Level Register

This register specifies the output logic of channel output bit (TAUDnTO.TAUDnTOm).

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 040_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TOL15	TAUDn TOL14	TAUDn TOL13	TAUDn TOL12	TAUDn TOL11	TAUDn TOL10	TAUDn TOL09	TAUDn TOL08	TAUDn TOL07	TAUDn TOL06	TAUDn TOL05	TAUDn TOL04	TAUDn TOL03	TAUDn TOL02	TAUDn TOL01	TAUDn TOL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.36 TAUDnTOL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOLm	Specifies the output logic of channel m output bit (TAUDnTO.TAUDnTOm). 0: Positive logic (active high) 1: Negative logic (active low) The setting of these bits applies to all channel output modes other than independent channel output mode controlled by software and independent channel output mode 1.

25.3.6 Details of TAUDn Dead Time Output Registers

25.3.6.1 TAUDnTDE — TAUDn Channel Dead Time Output Enable Register

This register enables/disables the dead time operation of every channel.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 250_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTDE15	TAUDnTDE14	TAUDnTDE13	TAUDnTDE12	TAUDnTDE11	TAUDnTDE10	TAUDnTDE09	TAUDnTDE08	TAUDnTDE07	TAUDnTDE06	TAUDnTDE05	TAUDnTDE04	TAUDnTDE03	TAUDnTDE02	TAUDnTDE01	TAUDnTDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.37 TAUDnTDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDEm	Enables/disables the dead time control operation of channel m. 0: Disables dead time operation 1: Enables dead time operation. The same setting should be made for both even and odd slave channels in pairs. These bit settings are applied when: • TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm, TAUDnTOC.TAUDnTOCm = 1

25.3.6.2 TAUDnTDM — TAUDn Channel Dead Time Output Mode Register

This register specifies the timing to add dead time during dead time output.

Access: This register can be read or written in 16-bit units. Writable only while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

Address: <TAUDn_base> + 254_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTDM15	TAUDnTDM14	TAUDnTDM13	TAUDnTDM12	TAUDnTDM11	TAUDnTDM10	TAUDnTDM09	TAUDnTDM08	TAUDnTDM07	TAUDnTDM06	TAUDnTDM05	TAUDnTDM04	TAUDnTDM03	TAUDnTDM02	TAUDnTDM01	TAUDnTDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.38 TAUDnTDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDMm	Specifies the timing to add dead time during dead time output. 0: When detecting the duty cycle of an upper even channel (duty dead time output). 1: When detecting the TIN input edge of a lower odd channel (one-phase dead time output). The same setting should be made for both even and odd slave channels in pairs. These bit settings are applied when: • TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm, TAUDnTOC.TAUDnTOCm, TAUDnTDE.TAUDnTDEm = 1

25.3.6.3 TAUDnTDL — TAUDn Channel Dead Time Output Level Register

This register selects a phase in which dead time is added.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 54_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDn TDL15	TAUDn TDL14	TAUDn TDL13	TAUDn TDL12	TAUDn TDL11	TAUDn TDL10	TAUDn TDL09	TAUDn TDL08	TAUDn TDL07	TAUDn TDL06	TAUDn TDL05	TAUDn TDL04	TAUDn TDL03	TAUDn TDL02	TAUDn TDL01	TAUDn TDL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.39 TAUDnTDL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTDLm	Selects a phase in which dead time is added. 0: Normal phase 1: Reverse phase These bit settings are applied when: • TAUDnTOE.TAUDnTOEm, TAUDnTOM.TAUDnTOMm, TAUDnTOC.TAUDnTOCm, TAUDnTDE.TAUDnTDEm = 1

25.3.7 Details of TAUDn Real-time/Modulation Output Registers

25.3.7.1 TAUDnTRE — TAUDn Channel Real-time Output Enable Register

This register enables/disables real-time output.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + 258_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTRE15	TAUDnTRE14	TAUDnTRE13	TAUDnTRE12	TAUDnTRE11	TAUDnTRE10	TAUDnTRE09	TAUDnTRE08	TAUDnTRE07	TAUDnTRE06	TAUDnTRE05	TAUDnTRE04	TAUDnTRE03	TAUDnTRE02	TAUDnTRE01	TAUDnTRE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.40 TAUDnTRE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTREm	Enables or disables real-time output of channel m. 0: Disables real-time output 1: Enables real-time output. These bit settings are applied only when TAUDnTOE.TAUDnTOEm = 1. When TAUDnTRE.TAUDnTREm = 0, TAUDTTOUTm is not affected by real-time output. When TAUDnTRE.TAUDnTREm = 1, TAUDTTOUTm outputs the value of real-time output bit TAUDnTRO.TAUDnTROm in response to a timer operation.

25.3.7.2 TAUDnTRC — TAUDn Channel Real-time Output Control Register

This register controls the real-time output trigger of each channel.

Access: This register can be read or written in 16-bit units. Writable only while TAUDnTE.TAUDnTEm = 0.

Address: <TAUDn_base> + $25C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTRC15	TAUDnTRC14	TAUDnTRC13	TAUDnTRC12	TAUDnTRC11	TAUDnTRC10	TAUDnTRC09	TAUDnTRC08	TAUDnTRC07	TAUDnTRC06	TAUDnTRC05	TAUDnTRC04	TAUDnTRC03	TAUDnTRC02	TAUDnTRC01	TAUDnTRC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.41 TAUDnTRC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTRCm	Specifies a channel on which the real-time output trigger for channel m is generated. 0: Next upper channel with this bit set to 1 1: Channel m These bit settings are applied only when TAUDnTRE.TAUDnTREm = 1.

25.3.7.3 TAUDnTRO — TAUDn Channel Real-time Output Register

This register sets a value which is output to TAUDTTOUTm.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + $04C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTRO15	TAUDnTRO14	TAUDnTRO13	TAUDnTRO12	TAUDnTRO11	TAUDnTRO10	TAUDnTRO09	TAUDnTRO08	TAUDnTRO07	TAUDnTRO06	TAUDnTRO05	TAUDnTRO04	TAUDnTRO03	TAUDnTRO02	TAUDnTRO01	TAUDnTRO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.42 TAUDnTRO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTROm	Sets a value which is output to TAUDTTOUTm. 0: Low 1: High TAUDnTROm value is not output to TAUDTTOUTm when TAUDnTRE.TAUDnTREm = 0, even if a real-time output trigger occurs.

25.3.7.4 TAUDnTME — TAUDn Channel Modulation Output Enable Register

This register enables/disables modulation output for timer output and real-time output.

Access: This register can be read or written in 16-bit units.

Address: <TAUDn_base> + 050_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnTME15	TAUDnTME14	TAUDnTME13	TAUDnTME12	TAUDnTME11	TAUDnTME10	TAUDnTME09	TAUDnTME08	TAUDnTME07	TAUDnTME06	TAUDnTME05	TAUDnTME04	TAUDnTME03	TAUDnTME02	TAUDnTME01	TAUDnTME00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25.43 TAUDnTME Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTMEm	Enables/disables modulation output for timer output and real-time output of channel m. 0: Disables modulation 1: Enables modulation These bit settings are applied only when TAUDnTOE.TAUDnTOEm and TAUDnTRE.TAUDnTREm = 1.

25.3.8 TAUDn Emulation Register

25.3.8.1 TAUDnEMU — TAUDn Emulation Register

This register controls SVSTOP operations.

Access: This register can be read or written in 8-bit units.
Perform write operations when the counter is stopped (TAUDnTE.TAUDnTEm = 0) and EPC.SVSTOP = 0.

Address: <TAUDn_base> + 290_H

Value after reset: 00_H

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

Table 25.44 TAUDnEMU Register Contents

Bit Position	Bit Name	Function
7	TAUDnSVSDIS	When EPC.SVSTOP bit = 0: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint), regardless of the value of this bit (1 or 0). When EPC.SVSTOP bit = 1: 0: The count clock is stopped when the debugger takes control of the microcontroller (as in the breakpoint). 1: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

25.4 Operating Procedure

The following lists the general operation procedure for the TAUDn.

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUDTTOUTm is also initialized and outputs a low level.

(1) Set the TAUDnTPS and TAUDnBRS registers to specify the clock frequency of CK0 to CK3.

(2) Configure the desired TAUDn function:
- Set the operation mode
- Set the channel output mode
- Set any other control bits

(3) Enable the counter by setting the TAUDnTS.TAUDnTSm bit to 1.
The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.

(4) If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUDnTT.TAUDnTTm bit to 1. The counter can be forcibly restarted by setting the TAUDnTS.TAUDnTSm bit to 1.

(5) Stop the function by setting the TAUDnTT.TAUDnTTm bit to 1.

NOTES

1. A detailed description of the required control bits and the operation of the individual functions are given in **Section 25.12, Independent Channel Operation Functions** and **Section 25.15, Synchronous Channel Operation Functions**.
2. The function can be changed while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

25.5 Concepts of Synchronous Channel Operation

The synchronous channel operation function is implemented using a combination of channel groups (consisted of master and slave channels). Several rules apply to the settings of channels. These rules are detailed in **Section 25.5.1, Rules of Synchronous Channel Operation**.

Two special features for synchronous channel operation are detailed in the following:

- **Section 25.5.2, Simultaneous Start and Stop of Synchronous Channel Counters**
- **Section 25.6, Simultaneous Rewrite**

25.5.1 Rules of Synchronous Channel Operation

Number of master and slave channels

- Only even channels (CH0, CH2, CH4, …) can be set as master channels. Any channel apart from CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and several slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 and the lower channels (CH3, CH4, CH5, …) can be set as slave channels.
- If multiple master channels are used, slave channels cannot cross the master channels.
  Example: If CH0 and CH4 are master channels, CH1 to CH3 can be set as slave channels for CH0, but CH5 to CH15 cannot.

Operation clock

- The same operation clock must be set for the slave channel and the master channel. This is achieved by setting the TAUDnCMORm.TAUDnCKS[1:0] bits of the slave and master channel.

The basic concepts of master/slave usage and operation clocks are illustrated in **Figure 25.4**.

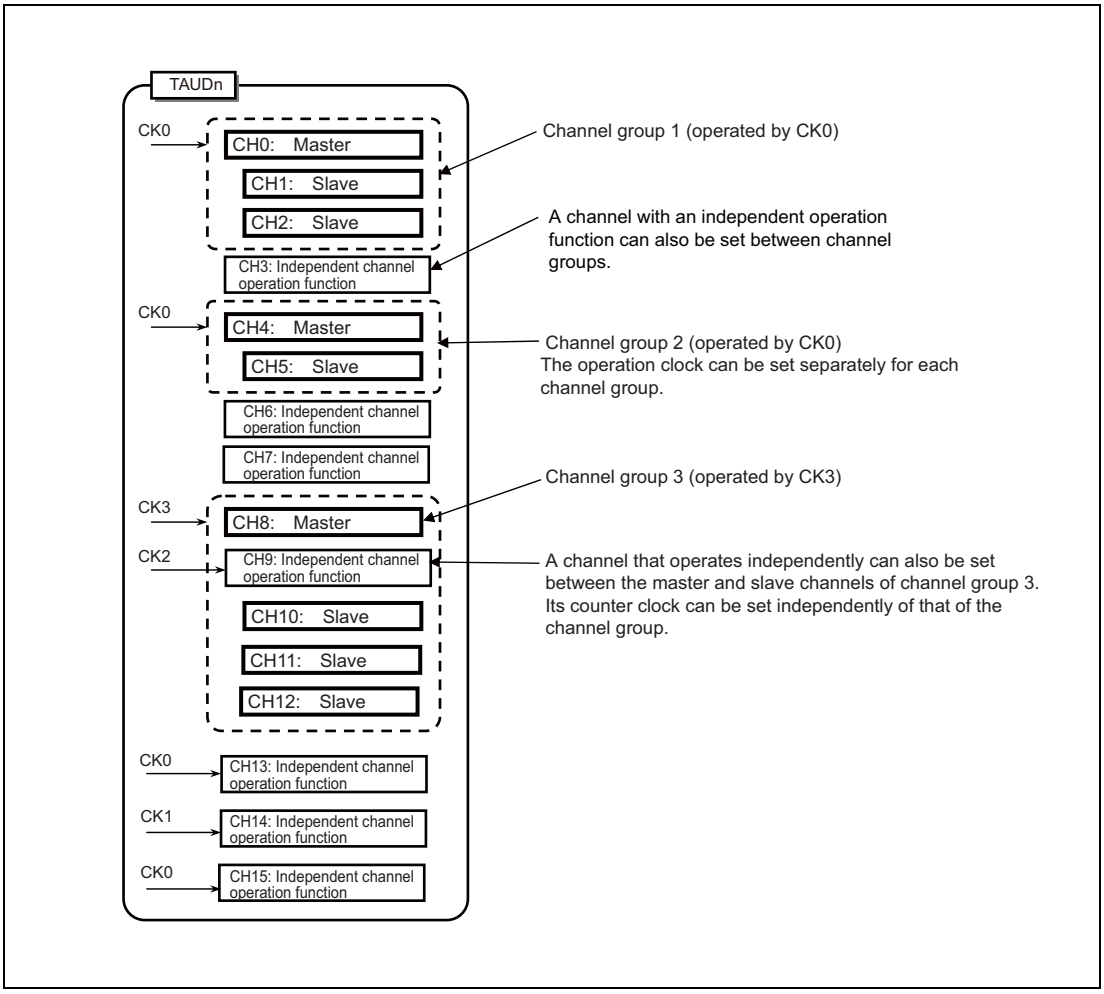


Figure 25.4 Grouping of Channels and Assignment of Count Clocks

Control trigger signal for master/slave channels

- Master channels can output control trigger signals to slave channels.
- Slave channels can use control trigger signals from master channels but cannot output control trigger signals of their own to lower channels.
- Master channels cannot use control trigger signals from upper master channels.

25.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously within the same unit and between the units.

25.5.2.1 Simultaneous Start and Stop within the Same Unit

- To simultaneously start synchronized channels, the TAUDnTS.TAUDnTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUDnTT.TAUDnTTm bits of the channels should be set at the same time.

Setting to the TAUDnTS.TAUDnTSm bits to 1 also sets the corresponding TAUDnTE.TAUDnTEm bits to 1, enabling counting. The count start timing depends on operating mode.

25.5.2.2 Simultaneous Start between the Units

Counters in different units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

For details about how to perform simultaneous start between the units, see **Section 29.8, Simultaneous Start Trigger Function**.

25.6 Simultaneous Rewrite

25.6.1 Overview of Operations

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUDnCDRm and TAUDnTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered by:

- The counter on the master channel or upper channel (depending on the selected operation mode) reaching a certain value
- INTTAUDnIm being issued on the upper channel specified by TAUDnRDC.TAUDnRDCm

There are four methods for simultaneous rewrite. These are listed in **Table 25.45**, along with how to specify them and when they cause simultaneous rewrite to be triggered.

Table 25.45 Simultaneous Rewrite Methods and when They are Triggered

Method	Simultaneous Rewrite Triggered when	TAUDnRDE. TAUDnRDEm	TAUDnRDS. TAUDnRDSm	TAUDnRDM. TAUDnRDMm
—	There is no simultaneous rewrite	0	0	0
A	The master channel (re)starts counting	1	0	0
B	Counting is started in the master channel. The master channel starts counting down at the peak of triangular cycle of the corresponding slave channel.	1	0	1
C1	INTTAUDnIm is generated on an upper channel specified by TAUDnRDC.TAUDnRDCm	1	1	0/1
C2	INTTAUDnIm is generated on an upper channel specified by TAUDnRDC.TAUDnRDCm that in turn is triggered by an external signal	1	1	0/1

Table 25.46 lists which of these four methods is available for each channel operation function. For more information about the individual channel operation functions, see the corresponding sections in **Section 25.14, Independent Channel Simultaneous Rewrite Functions**, **Section 25.15, Synchronous Channel Operation Functions** and **Section 25.16, Synchronous Non-Complementary and Complementary Modulation Output Functions**.

Table 25.46 Channel Functions and the Methods They Use for Simultaneous Rewrite

Function	A	B	C1	C2	TAUDnTOL. TAUDnTOLm
Simultaneous Rewrite Trigger Output Function Type 1			√		
PWM Output Function	√		√		√
One-Shot Pulse Output Function	√				
Trigger Start PWM Output Function	√			√	
Delay Pulse Output Function	√				
Triangle PWM Output Function		√	√		√
Triangle PWM Output Function with Dead Time		√	√		
Interrupt Request Signals Culling Function	√	√	√		
AD Conversion Trigger Output Function Type 1	√		√		
AD Conversion Trigger Output Function Type 2		√	√		
Non-Complementary Modulation Output Function Type 1	√		√		
Non-Complementary Modulation Output Function Type 2		√	√		
Complementary Modulation Output Function		√	√		

Note: √: Available, (Blank): Unavailable

25.6.2 How to Control Simultaneous Rewrite

Figure 25.5 shows the general procedure for simultaneous rewrite. The three main blocks (initial settings, start and counter count operation, and simultaneous rewrite) are explained afterwards.

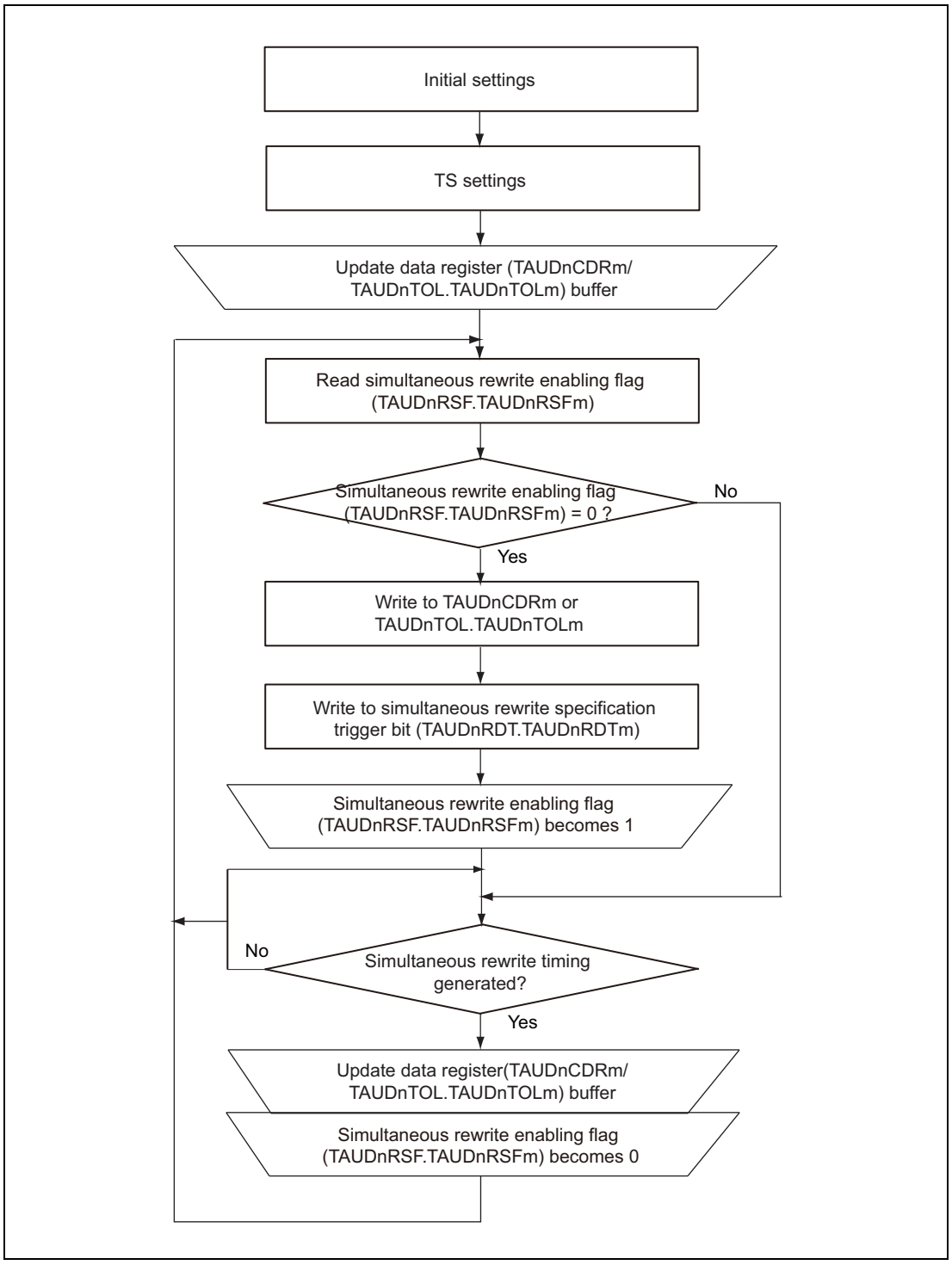


Figure 25.5 General Procedure for Simultaneous Rewrite

25.6.2.1 Initial Settings

- To enable simultaneous rewrite in channel m, set TAUDnRDE.TAUDnRDEm = 1
- To select the type of simultaneous rewrite, set TAUDnRDM.TAUDnRDMm and TAUDnRDS.TAUDnRDSm according to the values listed in **Table 25.45, Simultaneous Rewrite Methods and when They are Triggered**.
- Specify a simultaneous rewrite trigger channel by using TAUDnRDC.TAUDnRDCm. (Prerequisite: TAUDnRDS.TAUDnRDSm has been set to the upper channel.)

25.6.2.2 Start Counter and Count Operation

- To start all the TAUDnCNTm counters of the channel group, set the corresponding TAUDnTS.TAUDnTSm bits to 1. The values of TAUDnTOL.TAUDnTOLm and the data registers (TAUDnCDRm) are loaded into the corresponding TAUDnTOL.TAUDnTOLm buffer (TAUDnTOL.TAUDnTOLm buf) and data buffer registers (TAUDnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1 sets the reload flag (TAUDnRSF.TAUDnRSFm) to 1, enabling simultaneous rewrite. TAUDnRSF.TAUDnRSFm remains set to 1 until simultaneous rewrite is completed.
- When the specified trigger for simultaneous rewrite is detected, the TAUDnRSF.TAUDnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUDnRSF.TAUDnRSFm = 1). If it is, simultaneous rewrite is carried out. Otherwise the simultaneous rewrite is not carried out and waits for the next trigger detection.

25.6.2.3 Simultaneous Rewrite

- When simultaneous rewrite is enabled (TAUDnRSF.TAUDnRSFm = 1) and the simultaneous rewrite trigger is detected, the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and are applied the next time the counter starts or restarts.
- When simultaneous rewrite is complete, the TAUDnRSF.TAUDnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

25.6.3 Other General Rules of Simultaneous Rewrite

The following rules also apply:

- TAUDnRDE.TAUDnRDEm, TAUDnRDS.TAUDnRDSm, TAUDnRDM.TAUDnRDMm, and TAUDnRDC.TAUDnRDCm cannot be changed while the counter is in operation (TAUDnTE.TAUDnTEm = 1).
- TAUDnTOL.TAUDnTOLm can only be rewritten during operation with PWM output function or triangle PWM output function. For all other output functions, TAUDnTOL.TAUDnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUDTTOUTm outputs an invalid wave.
- When an upper channel is used as a channel issuing the simultaneous rewrite trigger (TAUDnRDS.TAUDnRDSm = 1), the TAUDnRDC.TAUDnRDCm bit controls all the lower channels. This means that if the TAUDnRDC.TAUDnRDCm bits of CH2 and CH7 are set to 1 and the TAUDnRDC.TAUDnRDCm bits of other channels are set to 0, CH2 and CH7 serve as simultaneous rewrite trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.
- If simultaneous rewrite is enabled and an upper channel is selected as the channel to generate the simultaneous rewrite trigger (TAUDnRDE.TAUDnRDEm and TAUDnRDS.TAUDnRDSm = 1) but no upper channel is set (TAUDnRDC.TAUDnRDC[15:0] = 0), simultaneous rewrite cannot take place.

25.6.4 Types of Simultaneous Rewrite

In the following section, the four simultaneous rewrite methods are explained using timing diagrams.

25.6.4.1 Simultaneous Rewrite when the Master Channel (Re)starts Counting (Method A)

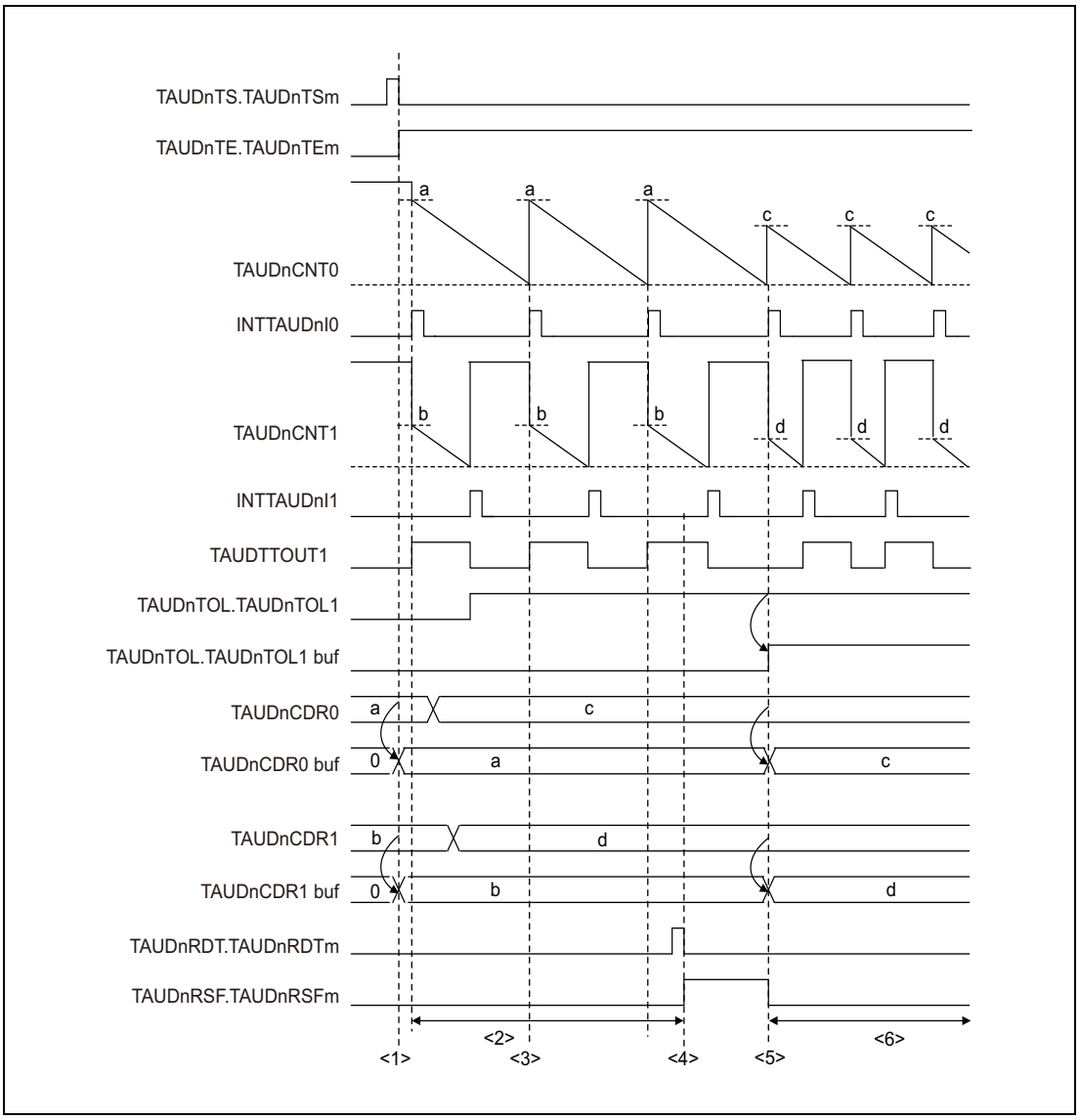


Figure 25.6 Simultaneous Rewrite when the Master Channel (Re)starts Counting

Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method A is applied.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer and TAUDnTOL.TAUDnTOLm value is copied to the TAUDnTOL.TAUDnTOLm buffer.

(2) The TAUDnCDRm and TAUDnTOL.TAUDnTOLm registers can be written at any time.

(3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUDnRSF.TAUDnRSFm = 0)

(4) The reload data trigger bit (TAUDnRDT.TAUDnRDTm) is set to 1 which sets the status flag (TAUDnRSF.TAUDnRSFm = 1), enabling simultaneous rewrite.

(5) Because simultaneous rewrite is enabled, it is triggered when CH0 restarts counting. The TAUDnCDRm value is loaded into the TAUDnCDRm buffer and the TAUDnTOL.TAUDnTOLm value is loaded into the TAUDnTOL.TAUDnTOLm buffer.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUDnCDRm and TAUDnTOL.TAUDnTOLm can be changed again.

25.6.4.2 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel (Method B)

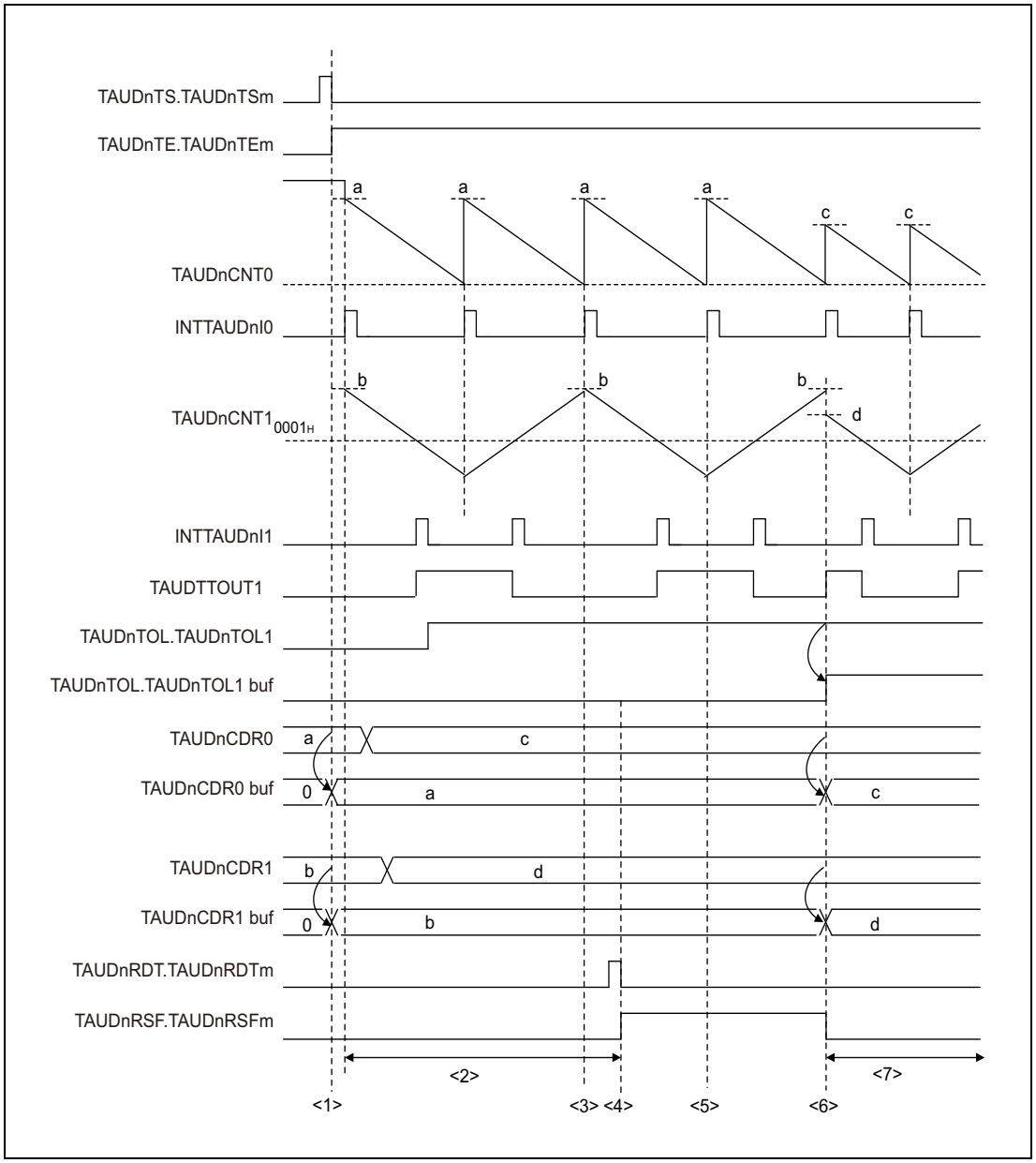


Figure 25.7 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel

Setting:

CH0 is the master channel which performs counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method B is applied.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer.

(2) The TAUDnCDRm and TAUDnTOL registers can be written at any time.

(3) Simultaneous rewrite does not occur because it is disabled (TAUDnRSF.TAUDnRSFm = 0).

(4) The reload data trigger bit (TAUDnRDT.TAUDnRDTm) is set to 1 which sets the status flag (TAUDnRSF.TAUDnRSFm = 1), enabling simultaneous rewrite.

(5) Simultaneous rewrite does not take place at the bottom of the triangular cycle.

(6) Simultaneous rewrite takes place at the top of the triangular cycle. The TAUDnCDRm value is loaded into the TAUDnCDRm buffer, the TAUDnTOL.TAUDnTOLm value is loaded into the TAUDnTOL.TAUDnTOLm buffer.

(7) The counters count down and await the next simultaneous rewrite trigger. The values of TAUDnCDRm and TAUDnTOL.TAUDnTOLm can be changed again.

25.6.4.3 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm (Method C1)

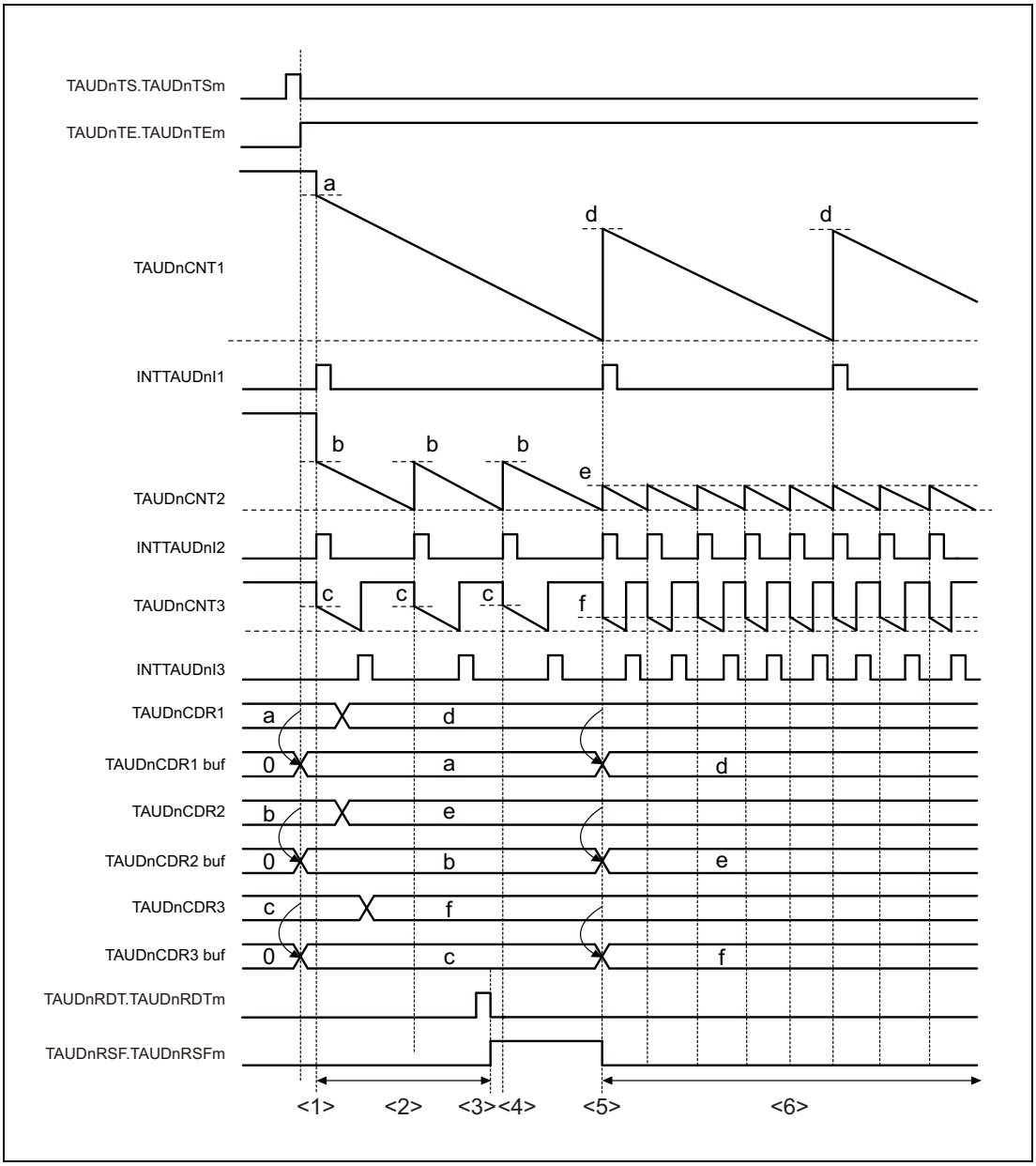


Figure 25.8 Simultaneous Rewrite When INTTAUDnIm Is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm

Setting:

CH1 is an upper channel which performs counting down, CH2 is a master channel, and CH3 is the slave channel. The simultaneous rewrite method C1 is applied. The TAUDnRDC register specifies a channel which generates simultaneous rewrite triggers.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer.

(2) The TAUDnCDRm register is always ready to write.

(3) By setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1, the status flag is set (TAUDnRSF.TAUDnRSFm = 1) to enable simultaneous rewrite.

(4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.

(5) Simultaneous rewrite is triggered by INT1 which is generated when counter 1 reaches 0000_H. The TAUDnCDRm values are loaded into the corresponding TAUDnCDRm buffers.

(6) The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUDnCDRm registers can be changed again.

25.6.4.4 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal (Method C2)

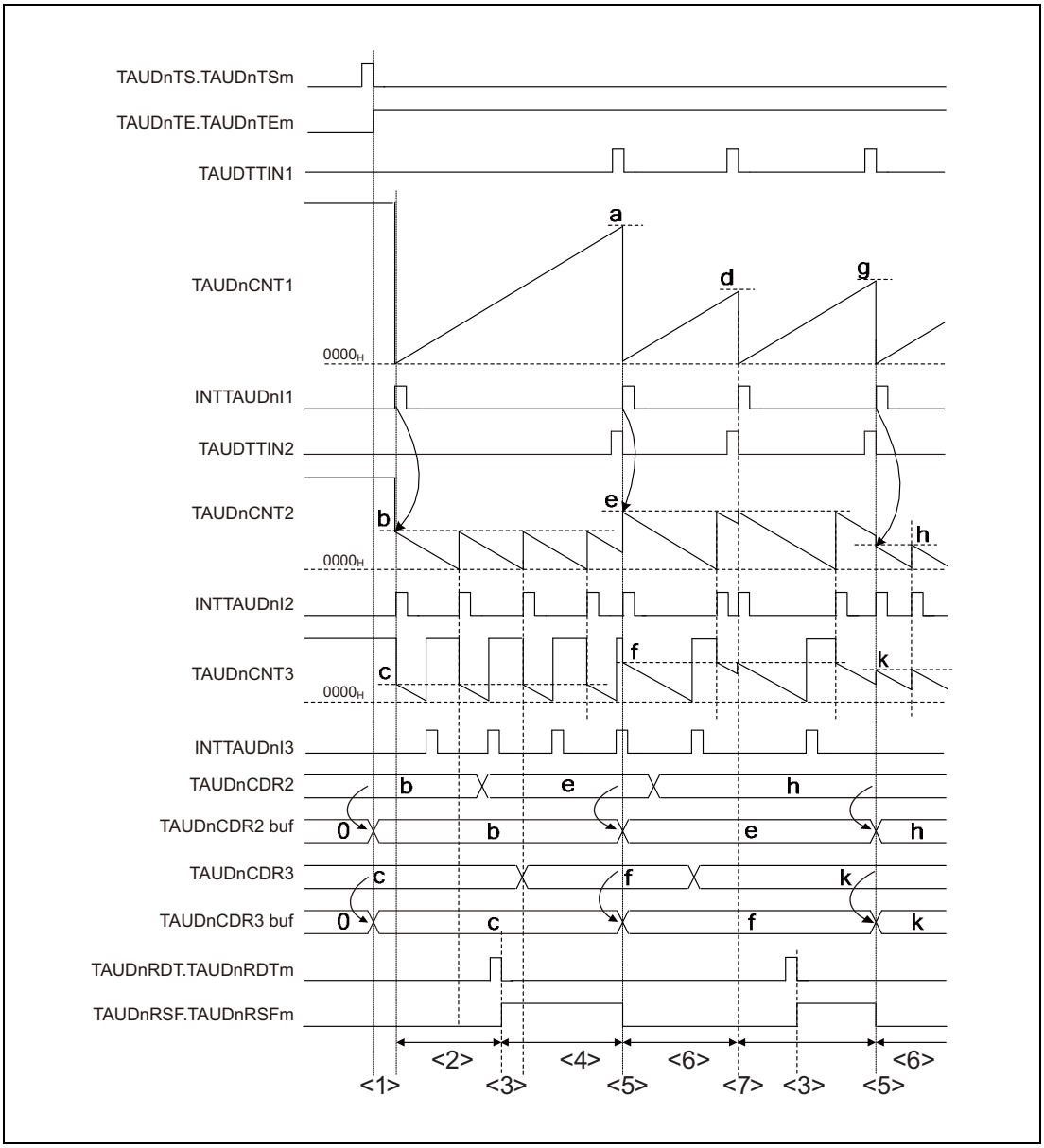


Figure 25.9 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal

Setting:

CH1 is an upper channel which performs counting up, CH2 is a master channel, and CH3 is the slave channel. The synchronous channel operation method C2 is applied. The TAUDnRDC register specifies which upper channel is monitored for an INTTAUDnIm trigger.

Description:

(1) When TAUDnTS.TAUDnTSm is set to 1, TAUDnCDRm value is copied to the TAUDnCDRm buffer. However, as TAUDnCDR1 operates in capture mode, TAUDnCDR1 value is not copied to the TAUDnCDR1 buffer.

(2) The TAUDnCDRm register is always ready to write.

(3) By setting the reload data trigger bit (TAUDnRDT.TAUDnRDTm) to 1, the status flag is set (TAUDnRSF.TAUDnRSFm = 1) to enable simultaneous rewrite.

(4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.

(5) Simultaneous rewrite is triggered by INT1 which is caused by external signal TIN1. The TAUDnCDRm values are written to the corresponding TAUDnCDRm buffers.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of the TAUDnCDRm registers can be changed again.

(7) An external signal occurs at TIN2 but simultaneous rewrite does not take place because it is disabled (TAUDnRSF.TAUDnRSFm = 0).

25.7 Channel Output Modes

The output of the TAUDTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUDnTOE.TAUDnTOEm = 0)
  When controlled by software, the value written in the output register bit (TAUDnTO.TAUDnTOm) is sent to the output pin (TAUDTTOUTm).
- By TAUD signals (TAUDnTOE.TAUDnTOEm = 1)
  When controlled by TAUD signals, the output level of TAUDTTOUTm is set or reset or toggled by internal signals. The value of TAUDnTO.TAUDnTOm is updated accordingly to reflect the value of TAUDTTOUTm.
  - Independently (TAUDnTOM.TAUDnTOMm = 0)
    In case of independent operation, the output of the TAUDTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUDnTOM.TAUDnTOMm = 0).
  - Synchronously (TAUDnTOM.TAUDnTOMm = 1)
    In case of synchronous operation, the output of the TAUDTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUDnTOM.TAUDnTOMm = 1).

The TAUDnTO.TAUDnTOm bit can always be read to determine the current value of TAUDTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 25.47, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 25.7.2, Channel Output Modes Controlled Independently by TAUDn Signals**
- **Section 25.7.3, Channel Output Modes Controlled Synchronously by TAUDn Signals**

Batch operation of TAUDnTOm bit

Whether a set value is reflected to the TAUDnTOm bit or not is controlled by the TAUDnTOE.TAUDnTOEm bit.

The TAUDnTOm setting is written only to the bit (channel) set with TAUDnTOE.TAUDnTOEm bit = 0 when a write to the TAUDnTO register is attempted. No TAUDnTOm setting is reflected to the bit (channel) set with TAUDnTOE.TAUDnTOEm bit = 1.

NOTE

TAUDnTO.TAUDnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or negative logic of the output is specified by control bit TAUDnTOL.TAUDnTOLm.

The value of TAUDnTOL.TAUDnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function or triangle PWM output function. Otherwise, changes to TAUDnTOL.TAUDnTOLm result in an invalid TAUDTTOUTm signal output.

See **Section 25.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 25.47**.

Table 25.47 Channel Output Modes

Channel Output Mode	TAUDn TOE. TAUDn TOEm	TAUDn TOM. TAUDn TOMm	TAUDn TOC. TAUDn TOCm	TAUDn TDE. TAUDn TDEm	TAUDn TRE. TAUDn TREm	TAUDn TME. TAUDn TMEm	TAUDn TDM. TAUDn TDMm
By software							
Independent channel output mode controlled by software	0	X					
By TAUD signals, independently							
Independent channel output mode 1	1	0	0	0	0	0	0
with real-time output					1		
Independent channel output mode 2			1		0		
By TAUD signals, synchronously							
Synchronous channel output mode 1	1	1	0	0	0	0	0
with non-complementary modulation output					1	X	
Synchronous channel output mode 2			1	0	0	0	0
with dead time output				1			
with one-phase PWM output							1
with complementary modulation output					1	1	0
with non-complementary modulation output			1	0			

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

NOTES

1. The following bits cannot be changed during count operation (TAUDnTE.TAUDnTEm = 1):
   - TAUDnTOM.TAUDnTOMm
   - TAUDnTOC.TAUDnTOCm
   - TAUDnTDE.TAUDnTDEm
   - TAUDnTRE.TAUDnTREm
   - TAUDnTDM.TAUDnTDMm
2. The following bits cannot be changed during count operation (TAUDnTE.TAUDnTEm = 1) except in channel output modes with modulation output:
   - TAUDnTME.TAUDnTMEm
   - TAUDnTDL.TAUDnTDLm

25.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUDTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUDnTOE.TAUDnTOEm = 0).

(1) Set TAUDnTO.TAUDnTOm to specify the initial level of the TAUDTTOUTm output.

(2) Set channel output mode according to **Table 25.47, Channel Output Modes**, and the output logic using the TAUDnTOL.TAUDnTOLm bit.

(3) Start the counter (TAUDnTS.TAUDnTSm = 1).

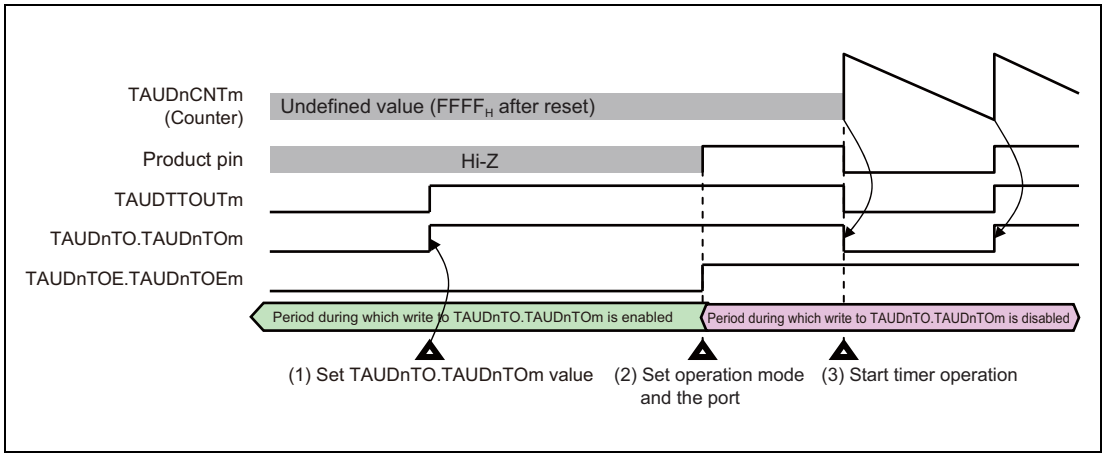


Figure 25.10 General Procedure for Specifying a TAUDTTOUTm Channel Output Mode

25.7.2 Channel Output Modes Controlled Independently by TAUDn Signals

This section lists the channel output modes that are controlled independently by TAUDn signals. The control bits used to specify a mode are listed in **Table 25.47, Channel Output Modes**.

25.7.2.1 Independent Channel Output Mode 1

Set/reset conditions

In this output mode, TAUDTTOUTm toggles when INTTAUDnIm is detected. The value of TAUDnTOL.TAUDnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 25.47, Channel Output Modes**.

25.7.2.2 Independent Channel Output Mode 1 with Real-Time Output

In this output mode, the value of TAUDnTRO.TAUDnTROm bit of the trigger channel is output to TAUDTTOUTm. The trigger channel is specified by setting the corresponding TAUDnTRC.TAUDnTRCm bit to 1. It controls all lower channels for which TAUDnTRC.TAUDnTRCm = 0.

Set/reset conditions

The value of TAUDnTRO.TAUDnTROm bit is sent to TAUDTTOUTm only when an INTTAUDnIm interrupt occurs on the trigger channel. The interrupt is generated either:

- at certain specified intervals or
- on detection of a valid TAUDTTINm input edge/counter start

The type of trigger is set using the TAUDnCMORm.TAUDnMD[4:1] bits.

Prerequisites

Both the master and slave channels can be set as a trigger generation channel. A channel for which TAUDnTRC.TAUDnTRCm is set to 1 serves as a trigger generation channel regardless of the value of TAUDnTRE.TAUDnTREm.

If there is no channel for which TAUDnTRC.TAUDnTRCm is set to 1 or if TAUDnTRC.TAUDnTRC0 = 0, real-time output cannot take place.

This can be seen in **Figure 25.11**.

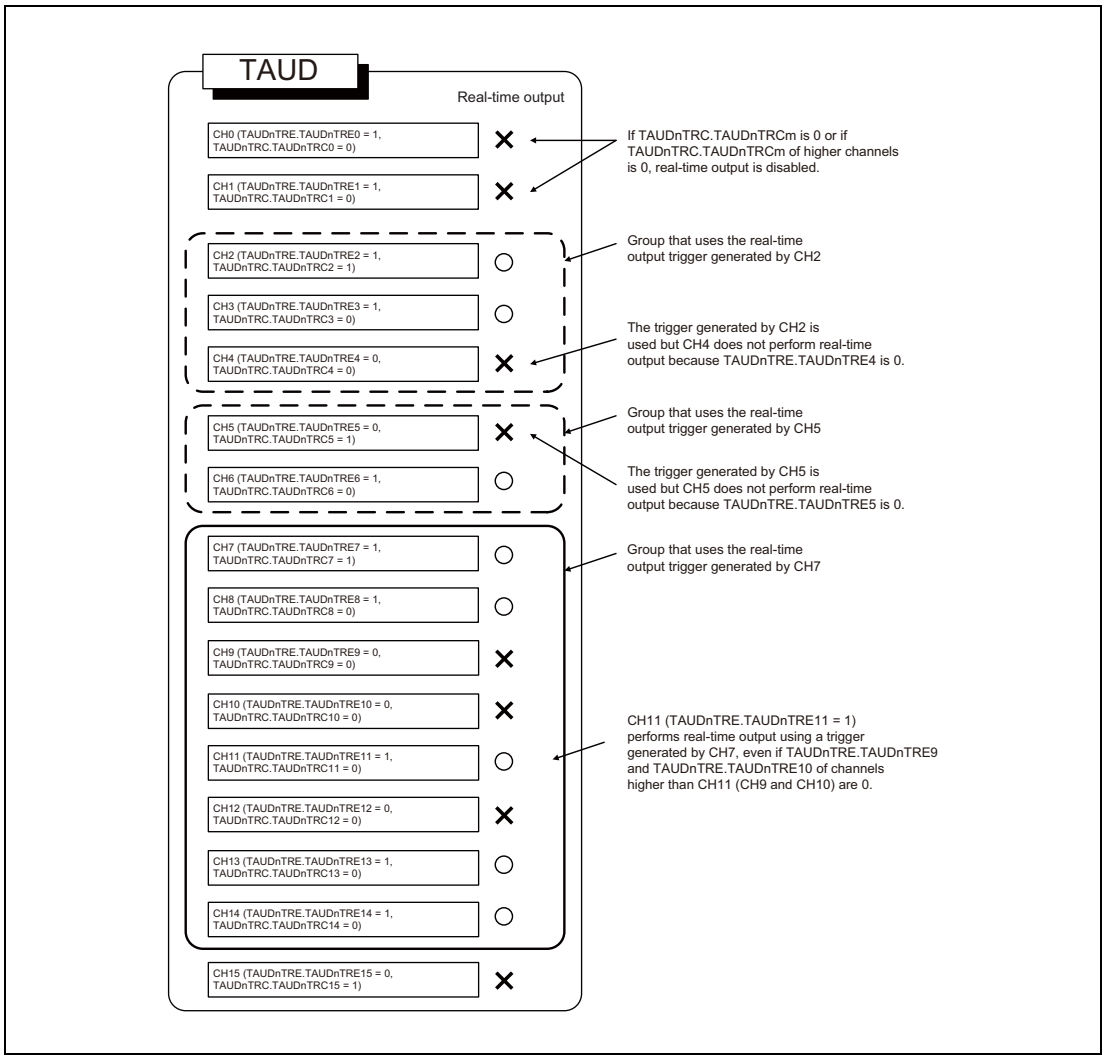


Figure 25.11 Real-Time Output

25.7.2.3 Independent Channel Output Mode 2

Set/reset conditions

In this output mode, TAUDTTOUTm is set when INTTAUDnIm occurs at the time of count start, and reset when INTTAUDnIm occurs due to a match between TAUDnCNTm and TAUDnCDRm.

Prerequisites

There are no prerequisites other than those shown in **Table 25.47, Channel Output Modes**.

25.7.3 Channel Output Modes Controlled Synchronously by TAUDn Signals

This section lists the channel output modes that are controlled synchronously by TAUDn signals. The control bits used to specify a mode are listed in **Table 25.47, Channel Output Modes**.

25.7.3.1 Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUDnIm of master channel serves as a set signal and INTTAUDnIm of the slave channel as a reset signal. If INTTAUDnIm of master channel and INTTAUDnIm of the slave channel are generated at the same time, INTTAUDnIm of the slave channel (reset signal) has priority over INTTAUDnIm (set signal) of master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 25.47, Channel Output Modes**.

25.7.3.2 Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Set/reset conditions

In this output mode, TAUDTTOUTm outputs the result of an AND operation between the PWM output and the real-time output bit (TAUDnTRO.TAUDnTROm) of a channel.

The phase period to which the dead time is added is specified using the TAUDnTDL.TAUDnTDLm bit; for positive phase set TAUDnTDL.TAUDnTDLm = 0 and for negative phase set TAUDnTDL.TAUDnTDLm = 1.

Prerequisites

A set of at least three channels is required to generate the PWM output. The master channel and slave channel 1 generate a period, and slave channel 2 generates the duty cycle. In typical applications, five more slave channels are also used that operate in the same manner as slave channel 2.

Only the PWM output and the real-time output bit of the same channel can be combined.

TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can only be changed during count operation.

- If TAUDnTME.TAUDnTMEm is changed, its new value is applied upon detection of INTTAUDnIm on the specified channel.
- If TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm are changed, their new values are applied upon detection of INTTAUDnIm on the master channel.

25.7.3.3 Synchronous Channel Output Mode 2

In this output mode, the operating mode should be set to count-up/-down mode. The result is a triangle PWM output at TAUDTTOUTm. For details, see **Section 25.15.7, Triangle PWM Output Function**.

Set/reset conditions

TAUDnCNTm of the slave channel counts down and up alternatively. When it passes 0001_H it generates an interrupt, causing TAUDTTOUTm to toggle.

Prerequisites

A set of two channels is required to generate the triangle PWM output. TAUDTTOUTm should be set to 0 before the function starts.

25.7.3.4 Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUDTTOUTm. The set/reset conditions are shown in **Figure 25.12**.

Set/reset conditions

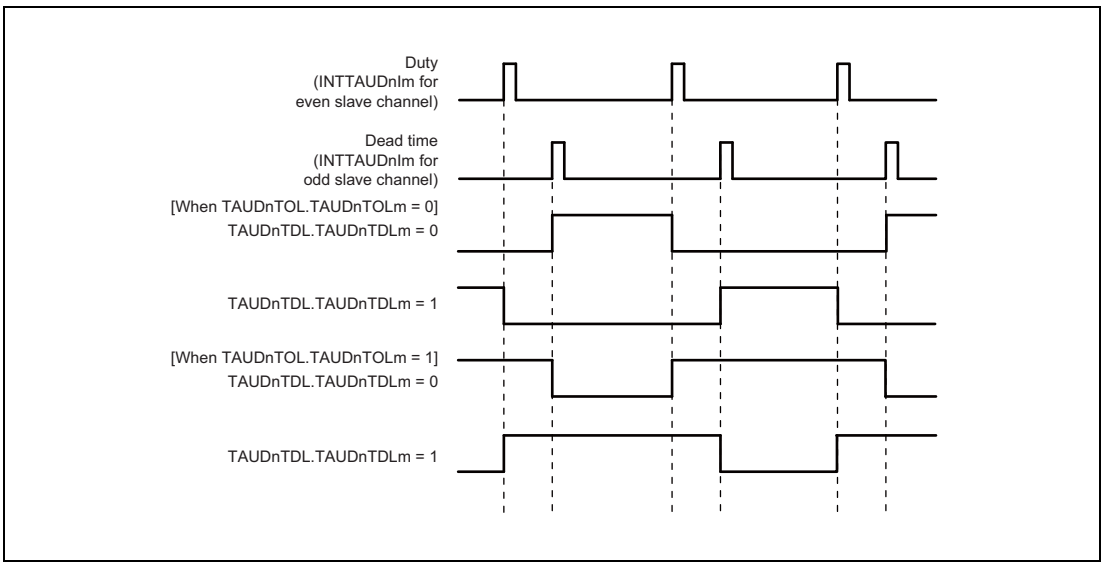


Figure 25.12 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output

With regard to the edge to which dead time is added, set TAUDnTDL.TAUDnTDLm = 0 for rising edges and TAUDnTDL.TAUDnTDLm = 1 for falling edges.

Prerequisites

Dead time control requires a set of three channels, each operating in the following modes:

- One master channel
  The master channel should be set to interval timer mode.
- One even slave channel
  The even slave channel should be set to count-up/-down mode.
- One odd slave channel (even channel + 1)
  The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUDnTOE.TAUDnTOEm
- TAUDnTME.TAUDnTMEm
- TAUDnTRE.TAUDnTREm
- TAUDnTOM.TAUDnTOMm
- TAUDnTOC.TAUDnTOCm

- TAUDnTDE.TAUDnTDEm
- TAUDnTDM.TAUDnTDMm

25.7.3.5 Synchronous Channel Output Mode 2 with One-Phase PWM Output

In this output mode, a dead time delay is added to TAUDTTOUTm. The set/reset conditions are shown in **Figure 25.13**.

Set/reset conditions

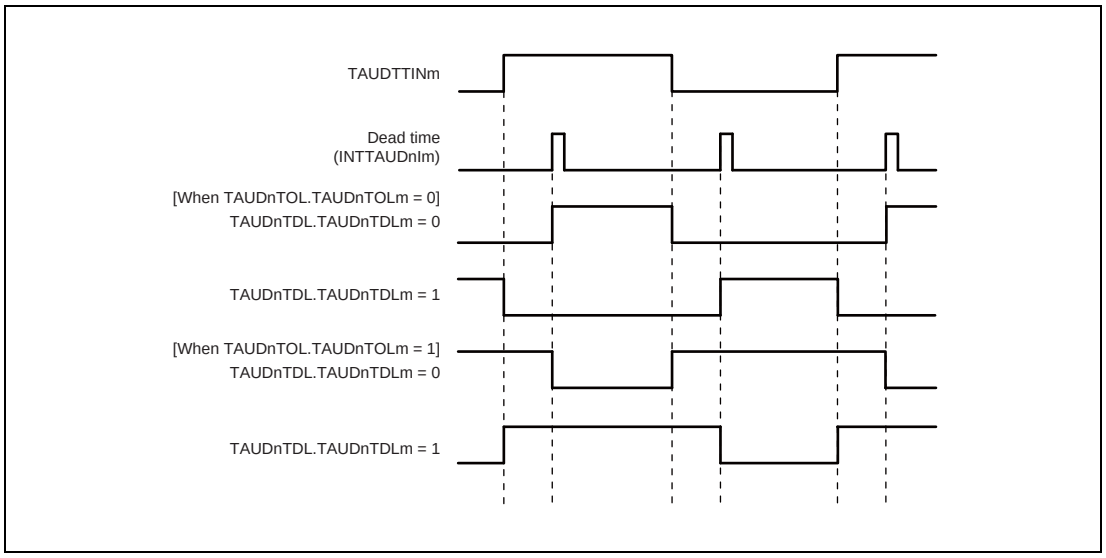


Figure 25.13 Set/Reset Conditions for Synchronous Channel Output Mode 2 with One-Phase PWM Output

With regard to the edge to which dead time is added, set TAUDnTDL.TAUDnTDLm = 0 for rising edges and TAUDnTDL.TAUDnTDLm = 1 for falling edges.

Prerequisites

One-phase PWM output control requires a set of two channels:

- One even slave channel
- One odd slave channel (even channel + 1)
  The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUDnTOE.TAUDnTOEm
- TAUDnTME.TAUDnTMEm
- TAUDnTRE.TAUDnTREm
- TAUDnTOM.TAUDnTOMm
- TAUDnTOC.TAUDnTOCm
- TAUDnTDE.TAUDnTDEm
- TAUDnTDM.TAUDnTDMm

25.7.3.6 Synchronous Channel Output Mode 2 with Complementary Modulation Output

Set/reset conditions

In this output mode, TAUDTTOUTm outputs a PWM signal, a high signal, or a low signal depending on the value of real-time output bit (TAUDnTRO.TAUDnTROm), the modulation output bit (TAUDnTME.TAUDnTMEm), and the output level bit (TAUDnTOL.TAUDnTOLm) of a pair of slave channels.

For details, see **Section 25.16.3, Complementary Modulation Output Function**.

Prerequisites

A set of at least four channels is required for this mode. The master channel and slave channel 1 generate a period, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time. Slave channels 2 and 3 are a pair. In typical applications, four more channels are also used, which operates in the same manner as slave channels 2 and 3 respectively.

TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can only be changed during count operation.

- If TAUDnTME.TAUDnTMEm is changed during operation, its new value is applied upon detection of INTTAUDnIm at the specified channel.
- If TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm are changed, their new values are applied upon detection of INTTAUDnIm on an even slave channel.

25.7.3.7 Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

The difference from synchronous channel output mode 1 with non-complementary modulation output is the PWM wave shape.

Mode 1 has a square wave while mode 2 has a triangular wave.

25.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUDnTS.TAUDnTSm is set to 1 in each operating mode.

In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

25.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-up/-down Mode, and Count Capture Mode

The counter starts operating with the next count clock cycle after TAUDnTS.TAUDnTSm is set to 1. The value of data register is also loaded when the counter starts.

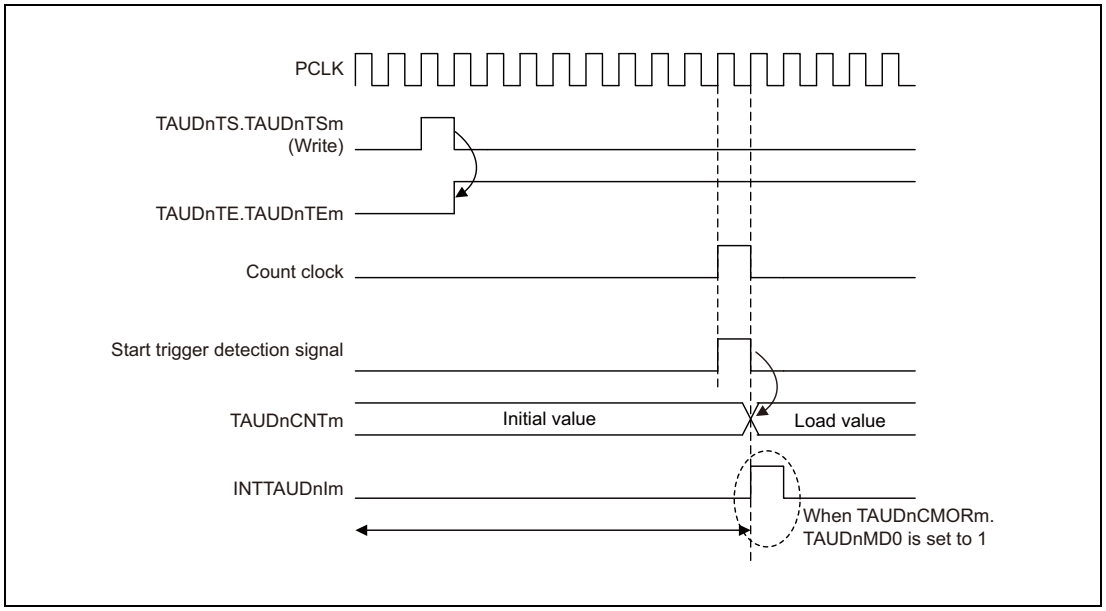


Figure 25.14 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, Count-up/-down Mode, and Count Capture Mode

NOTE

Make sure to set TAUDnCMORm.TAUDnMD0 to 0 when using the count-up/-down mode.

25.8.2 Event Count Mode

The value of data register is loaded as soon as TAUDnTS.TAUDnTSm is set to 1. The counter also starts immediately. The value of data register decrements when the subsequent count clock cycle starts.

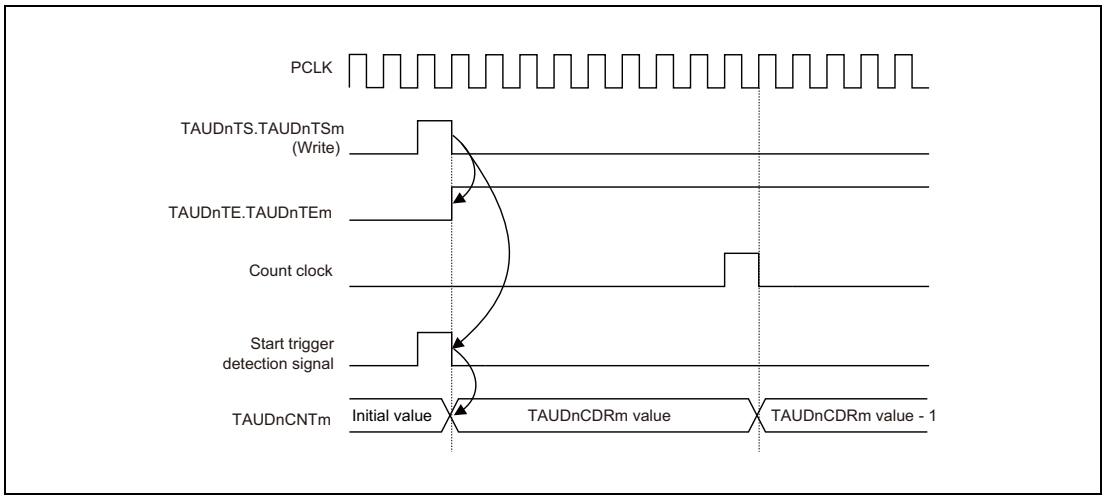


Figure 25.15 Start Timing in Event Count Mode

25.8.3 Other Operating Modes

In other operating modes, the counter operation start timing is triggered only upon detection of a valid edge of TAUDTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which are irrelevant to start of counter operation, determine the frequency with which all operations take place.

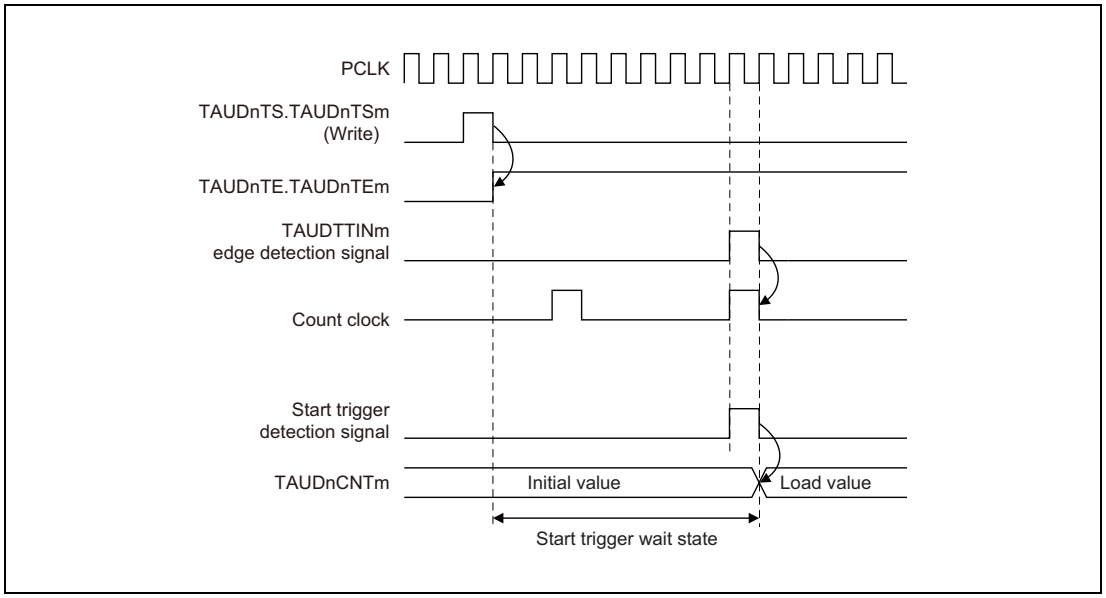


Figure 25.16 Count Start Timing in Other Operating Modes

25.9 TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUDnIm is generated using the TAUDnCMORm.TAUDnMD0 bit. The generation of INTTAUDnIm when the TAUDnCMORm.TAUDnMD0 bit starts counting and the effect to TAUDTTOUTm depend on the selected function. For details, refer to the description of TAUDnCMORm.TAUDnMD0 of each function.

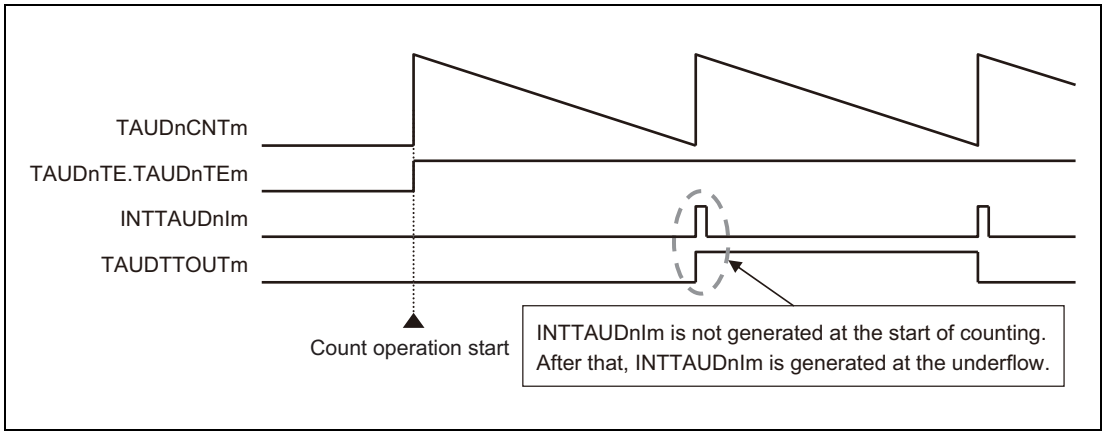


Figure 25.17 INTTAUDnIm Generation Timing (when TAUDnCMORm.TAUDnMD0 = 0)

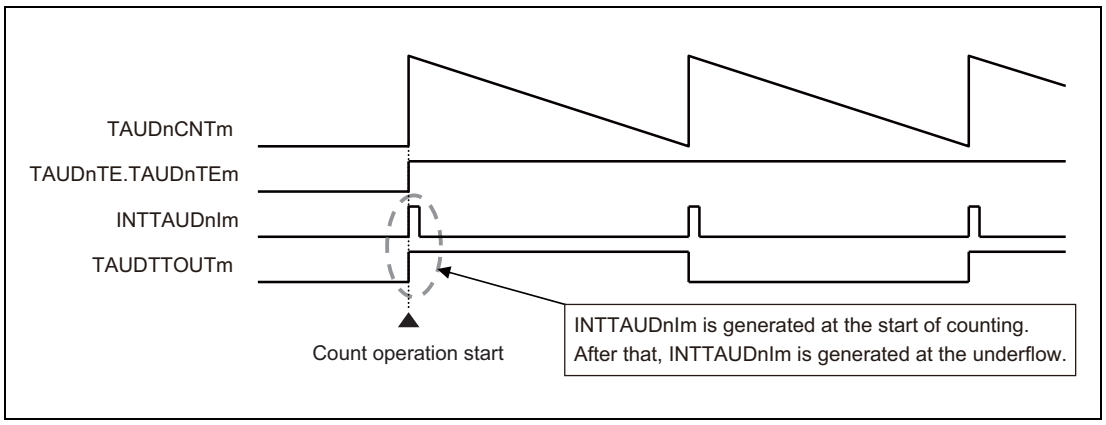


Figure 25.18 INTTAUDnIm Generation Timing (when TAUDnCMORm.TAUDnMD0 = 1)

25.10 Interrupt Generation upon Overflow

In certain independent functions, an interrupt is not generated when the counter value reaches $FFFF_H$ and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operation in a mode that counts up and in a mode that counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000_H at the same time as the first channel overflows (TAUDnCNTm = $FFFF_H$).
- Set TAUDnCDRm of the second channel to $FFFF_H$.
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same TAUDTTINm input.
- The trigger detection settings (TAUDnCMORm.TAUDnSTS[2:0] and TAUDnCMURm.TAUDnTIS[1:0]) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000_H at exactly the same time as the up-counter of the first channel overflows (TAUDnCNTm = $FFFF_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

25.10.1 Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDnIm of the TAUDTTINm input interval timer function can detect the overflow when TAUDnCNTm of the TAUDTTINm input pulse interval measurement function exceeds $FFFF_H$.

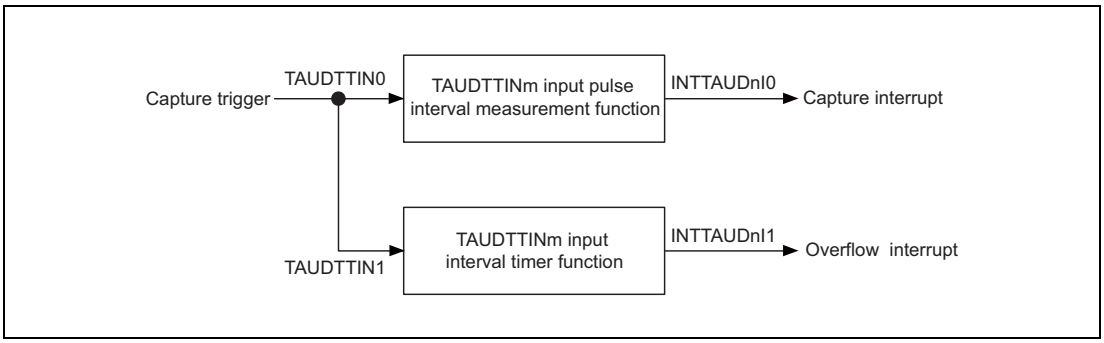


Figure 25.19 Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function

Timing diagram

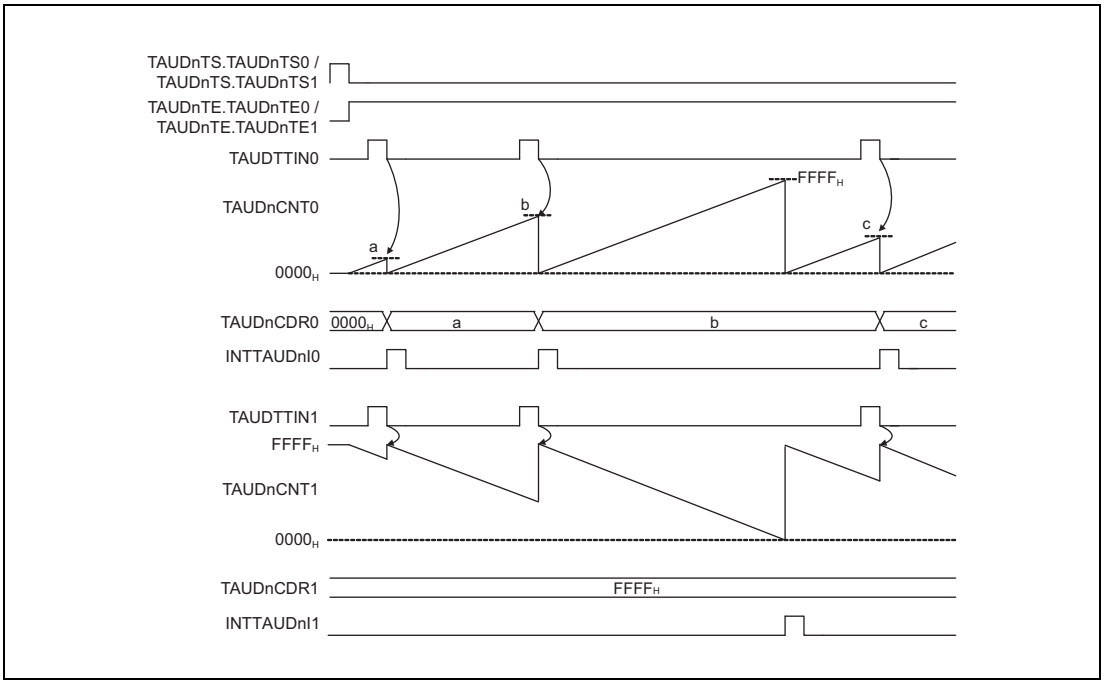


Figure 25.20 Interrupt Generation via Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function

25.10.2 Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width)

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDnIm of the overflow interrupt output function (at measuring the TAUDTTINm width) can detect the overflow when TAUDnCNTm of the TAUDTTINm input signal width measurement function exceeds $FFFF_H$.

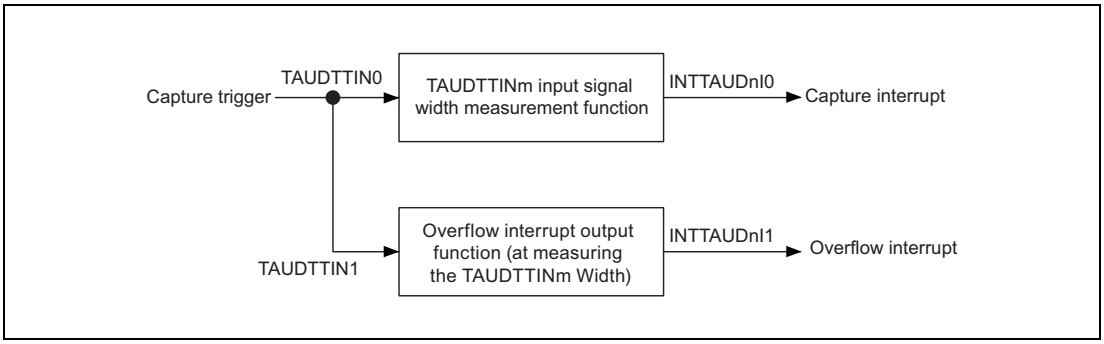


Figure 25.21 Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width)

Timing diagram

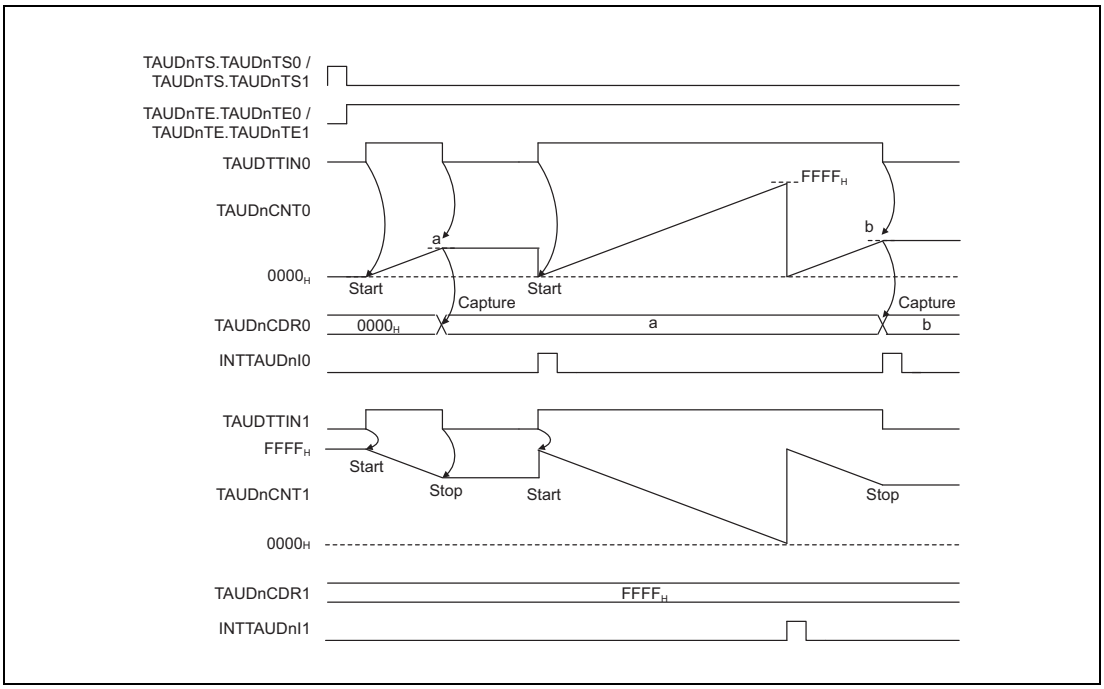


Figure 25.22 Interrupt Generation via Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width)

25.10.3 Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function

When the counters of both channels are enabled simultaneously, INTTAUDnIm of the interval timer function can detect the overflow when TAUDnCNTm of the TAUDTTINm input position detection function exceeds $FFFF_H$.

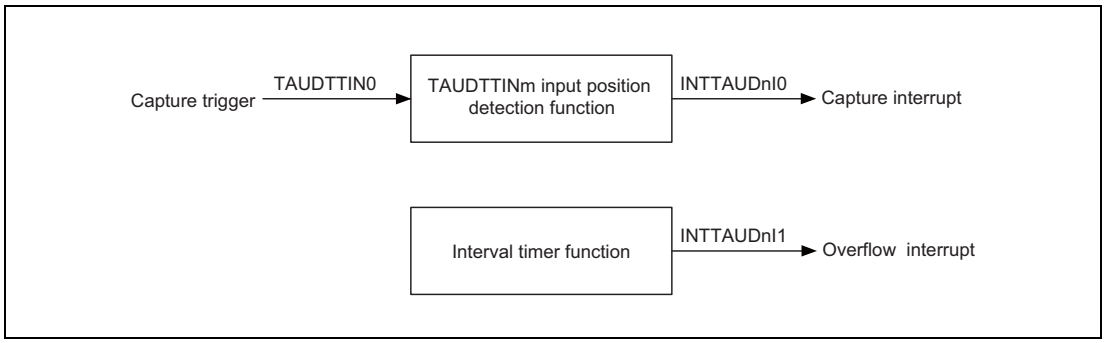


Figure 25.23 Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function

Timing diagram

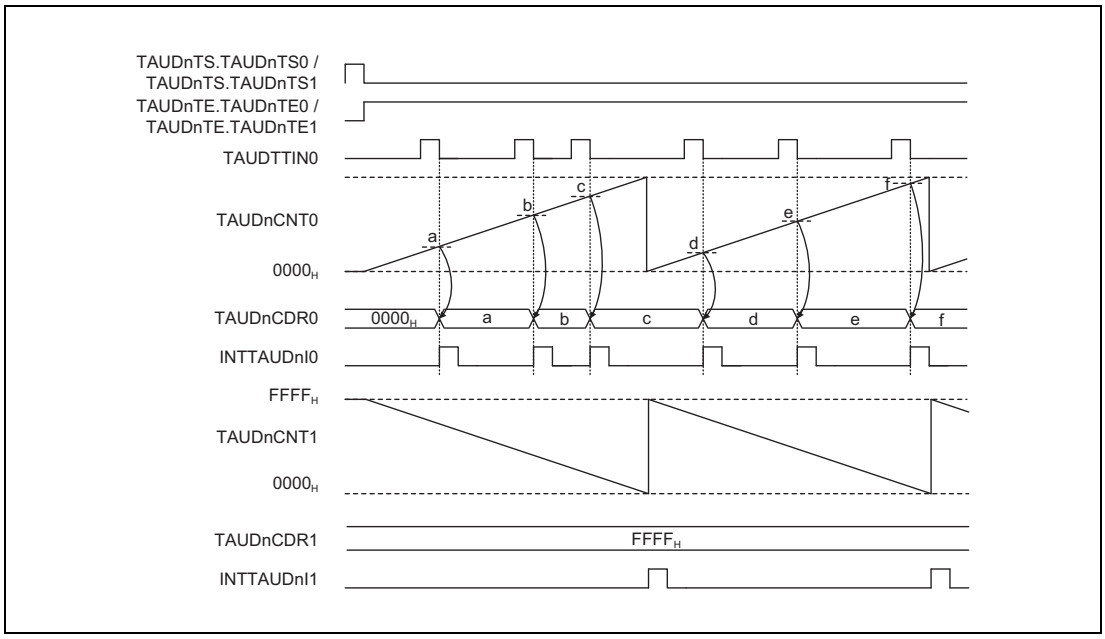


Figure 25.24 Interrupt Generation via Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function

25.10.4 Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count)

When the capture trigger is input simultaneously to TAUDTTINm of both channels, INTTAUDnIm of the overflow interrupt output function (at detecting the TAUDTTINm input period count) can detect the overflow when TAUDnCNTm of the TAUDTTINm input period count detection function exceeds $FFFF_H$.

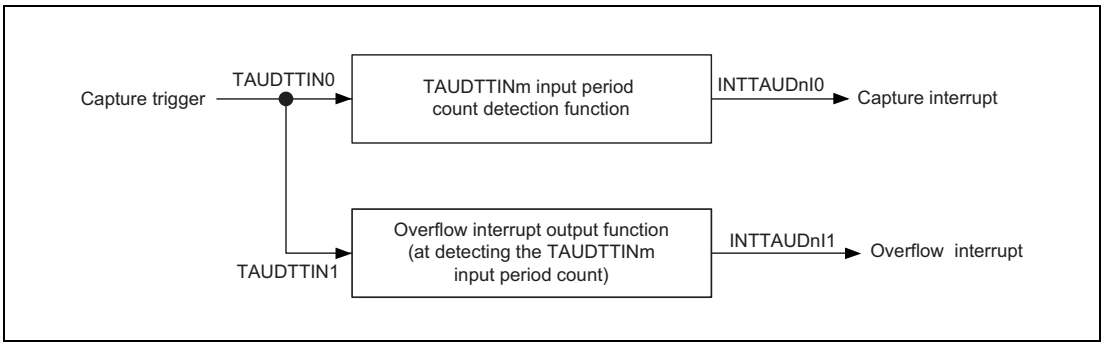


Figure 25.25 Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count)

Timing diagram

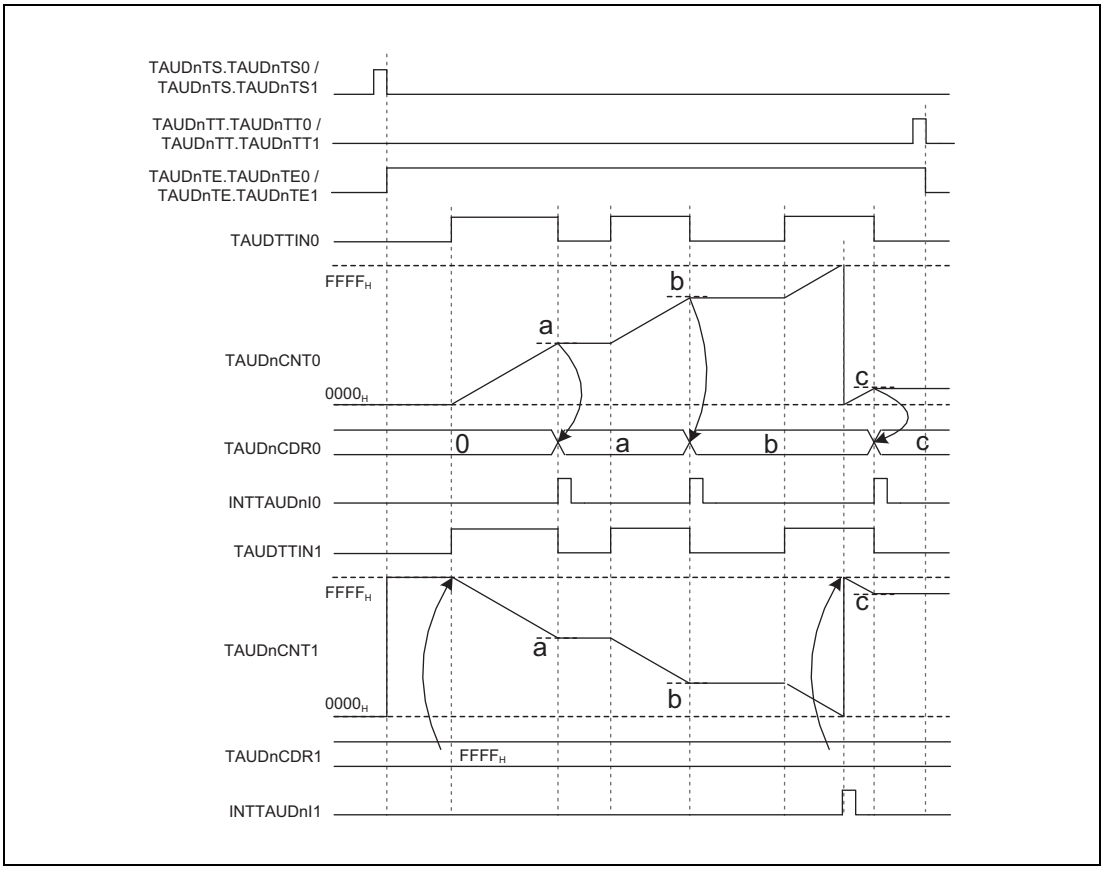


Figure 25.26 Interrupt Generation via Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count)

25.11 TAUDTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

Figure 25.27 shows when edge detection takes place.

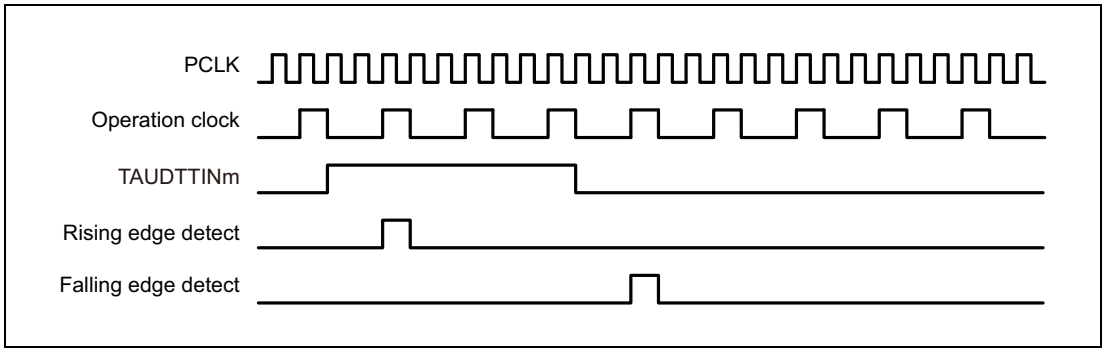


Figure 25.27 Basic Edge Detection Timing

Figure 25.27 shows an operation timing image. Actually, a noise filter or synchronization circuit which is located between the TAUDnIm pin and TAUDn causes a delay time.

25.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the Timer Array Unit D. For a general overview of independent channel operation, see **Section 25.2, Overview**.

This section describes functions that generate interrupts at regular intervals or with a specified delay.

25.12.1 Interval Timer Function

25.12.1.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDnIm) at regular intervals. When an interrupt is generated, the TAUDTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode must be set to Interval Timer Mode, see **Table 25.48, Contents of the TAUDnCMORm Register for Interval Timer Function**.
- The channel output mode must be set to Independent Channel Output Mode 1, see **Section 25.7, Channel Output Modes**.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is written to TAUDnCNTm and the counter starts to count down from this value.

When the counter reaches 0000_H, INTTAUDnIm is generated and the TAUDTTOUTm signal toggles. TAUDnCNTm then reloads the TAUDnCDRm value and subsequently continues operation.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1, which in turn sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm stop but retain their values. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm to 1 during operation.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUDTTOUTm does not toggle. This results in a negative TAUDTTOUTm signal compared to when TAUDnCMORm.TAUDnMD0 is set to 1. For details see **Section 25.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

25.12.1.2 Equations

INTTAUDnIm cycle = count clock cycle × (TAUDnCDRm + 1)

TAUDTTOUTm square wave cycle = count clock cycle × (TAUDnCDRm + 1) × 2

25.12.1.3 Block Diagram and General Timing Diagram

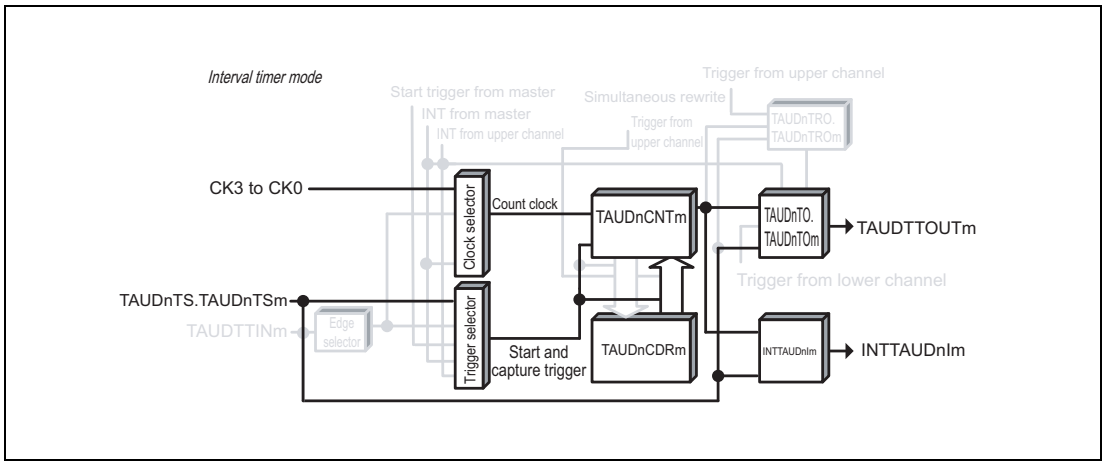


Figure 25.28 Block Diagram of Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 1).

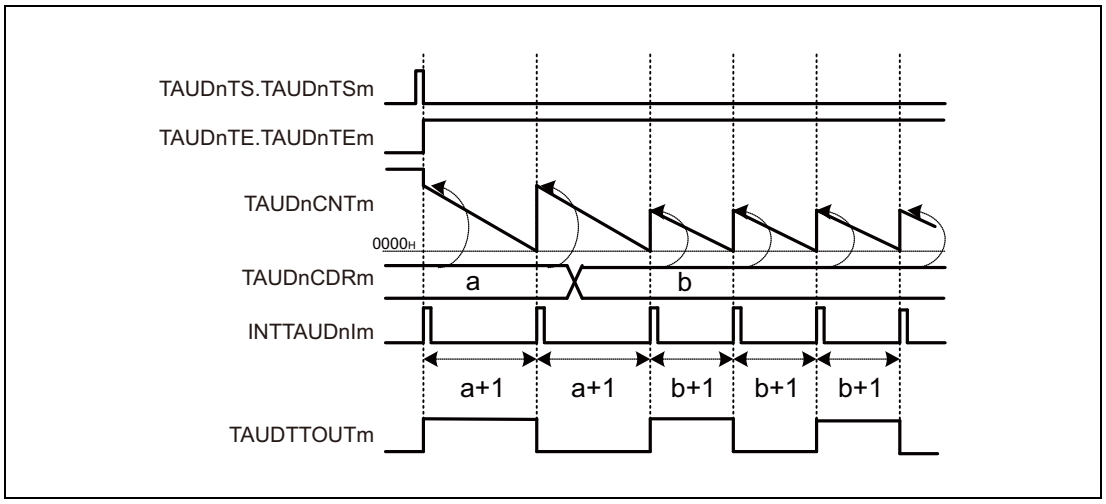


Figure 25.29 General Timing Diagram of Interval Timer Function

25.12.1.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.48 Contents of the TAUDnCMORm Register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS [1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, set to 0.
10 to 8	TAUDnSTS[2:0]	000: Triggers the counter by software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated to toggle TAUDTTOUTm at the beginning of an operation.

(2) TAUDnCMURm

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

Table 25.49 Contents of the TAUDnCMURm Register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode

Table 25.50 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDTTOUTm can then be controlled independently of the interrupts. For details, see **Section 25.7, Channel Output Modes**.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the interval timer function. Therefore, these registers should be set to 0.

Table 25.51 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.1.5 Operating Procedure for Interval Timer Function

Table 25.52 Operating Procedure for Interval Timer Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.48, Contents of the TAUDnCMORm Register for Interval Timer Function**, and **Table 25.49, Contents of the TAUDnCMURm Register for Interval Timer Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 25.50, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. When TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated and TAUDTTOUTm toggles.
During Operation	The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down. When the counter reaches 0000_H: • The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. • INTTAUDnIm is generated and TAUDTTOUTm toggles.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart operation

25.12.1.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H, count clock = PCLK/2

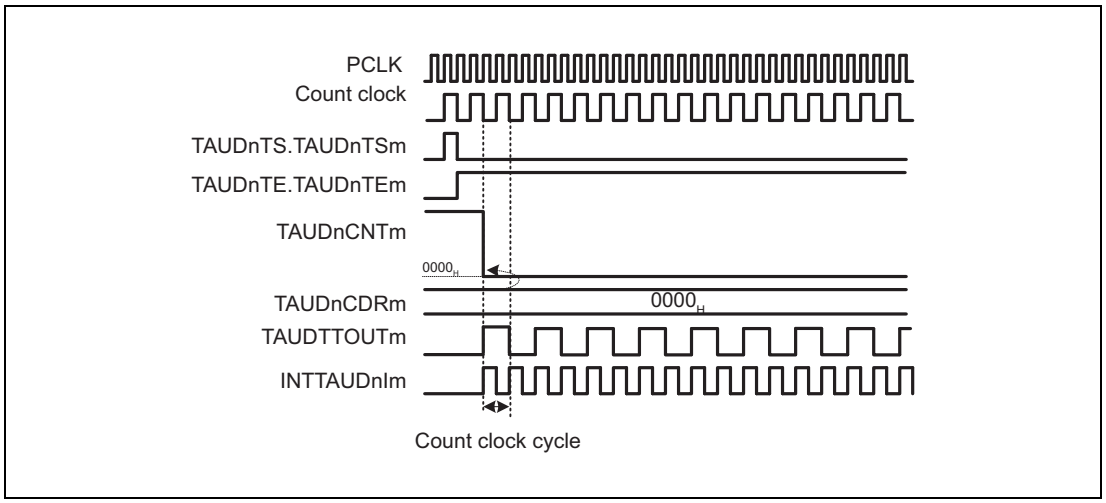


Figure 25.30 TAUDnCDRm = 0000_H, Count Clock = PCLK/2

- If TAUDnCDRm = 0000_H and the count clock = PCLK/2, the TAUDnCDRm value is loaded into TAUDnCNTm every count clock, meaning that TAUDnCNTm is always 0000_H.
- INTTAUDnIm is generated every count clock, resulting in TAUDTTOUTm toggling every count clock.

(2) TAUDnCDRm = 0000_H, count clock = PCLK

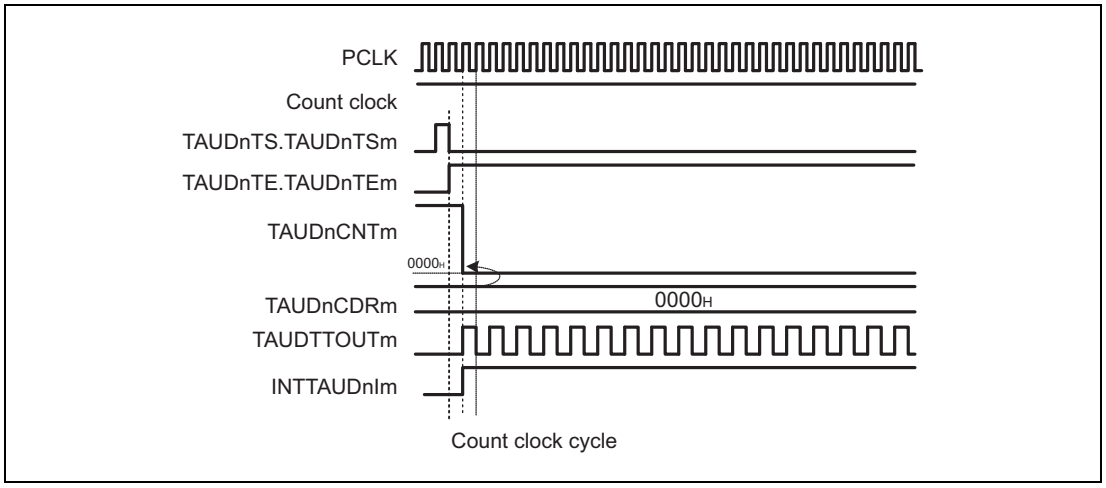


Figure 25.31 TAUDnCDRm = 0000_H, Count Clock = PCLK

- If TAUDnCDRm = 0000_H and the count clock = PCLK, the TAUDnCDRm value is loaded into TAUDnCNTm every PCLK clock, meaning that TAUDnCNTm is always 0000_H.
- INTTAUDnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated.
  TAUDTTOUTm is toggled every PCLK clock.

(3) Operation stop and restart

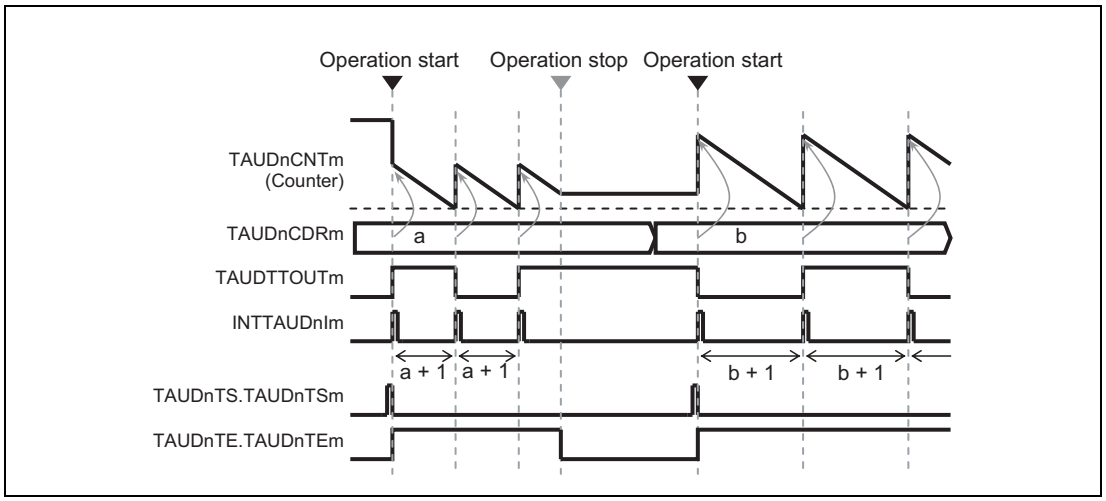


Figure 25.32 Operation Stop and Restart (TAUDnCMORm.TAUDnMD0 = 1)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm and TAUDTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

(4) Forced restart (TAUDnCMORm.TAUDnMD0 = 1)

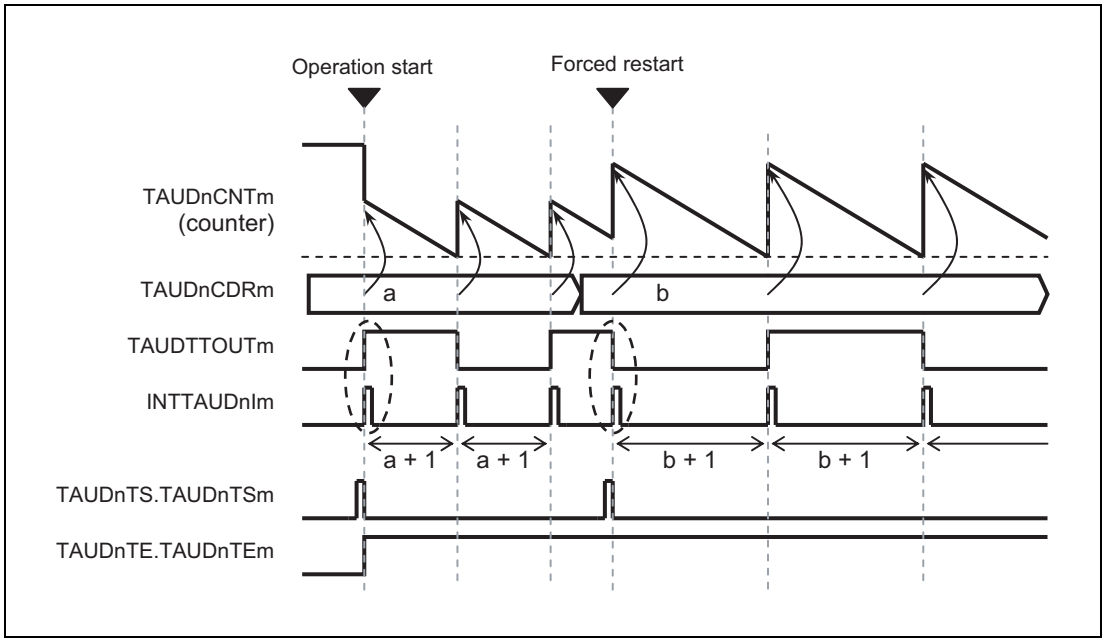


Figure 25.33 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm to 1 during operation.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- When a forced restart is made, the TAUDnCDRm value is reflected to TAUDnCNTm and counting starts. Execute a forced restart to reflect the changed TAUDnCDRm value immediately.
- When a forced restart is made, an interrupt (INTTAUDnIm) is generated and TAUDTTOUTm is inverted.

(5) Forced restart (TAUDnCMORm.TAUDnMD0 = 0)

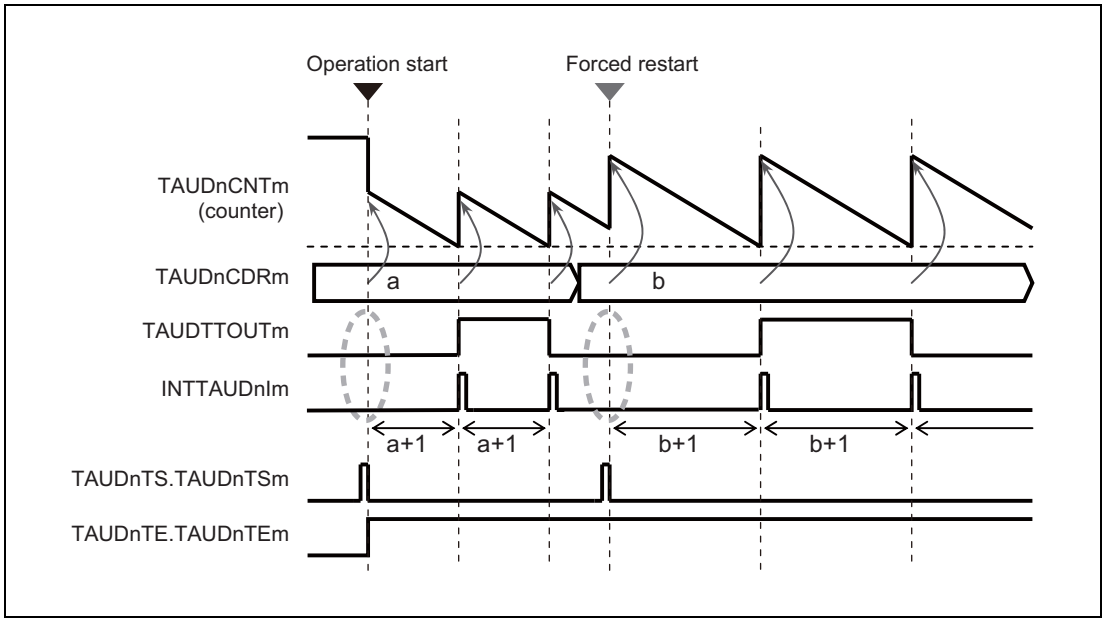


Figure 25.34 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 0)

- When a forced restart is made, an interrupt (INTTAUDnIm) is not generated and TAUDTTOUTm is not inverted.

25.12.2 TAUDTTINm Input Interval Timer Function

25.12.2.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUDnIm) at regular intervals or when a valid TAUDTTINm input edge is detected. When an interrupt is generated, the TAUDTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operating mode should be set to interval timer mode. See **Table 25.53, Contents of the TAUDnCMORm Register for TAUDTTINm Input Interval Timer Function**.
- The channel output mode should be set to independent channel output mode 1. See **Section 25.7, Channel Output Modes**.

Functional description

This function operates in an identical manner to the interval timer function (see **Section 25.12.1, Interval Timer Function**) except that this function is restarted by a valid TAUDTTINm input edge. The type of edge used as a trigger is specified using the TAUDnCMURm.TAUDnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

25.12.2.2 Equations

INTTAUDnIm cycle = count clock cycle × (TAUDnCDRm + 1)

TAUDTTOUTm square wave cycle = count clock cycle × (TAUDnCDRm + 1) × 2

25.12.2.3 Block Diagram and General Timing Diagram

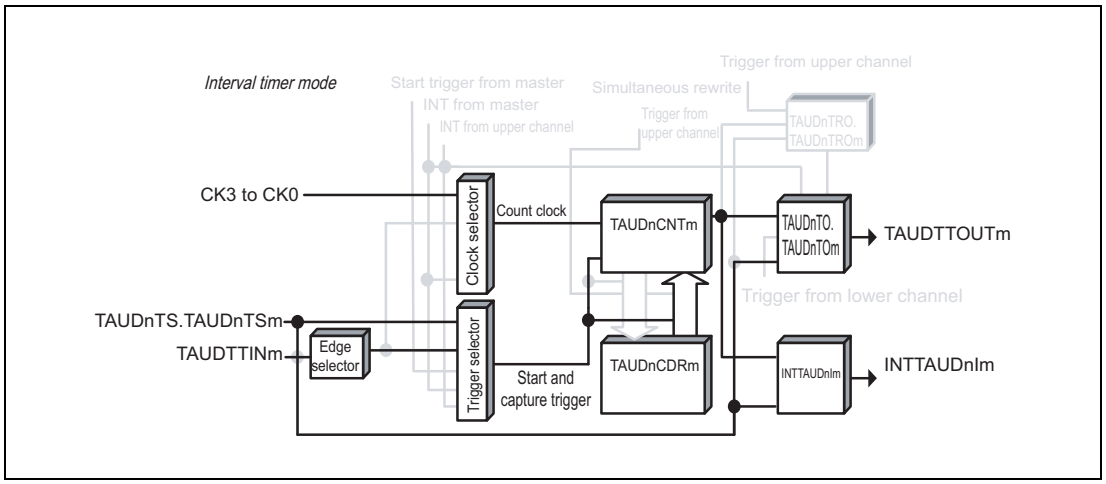


Figure 25.35 Block Diagram of TAUDTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 1).
- Rising edge detection (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

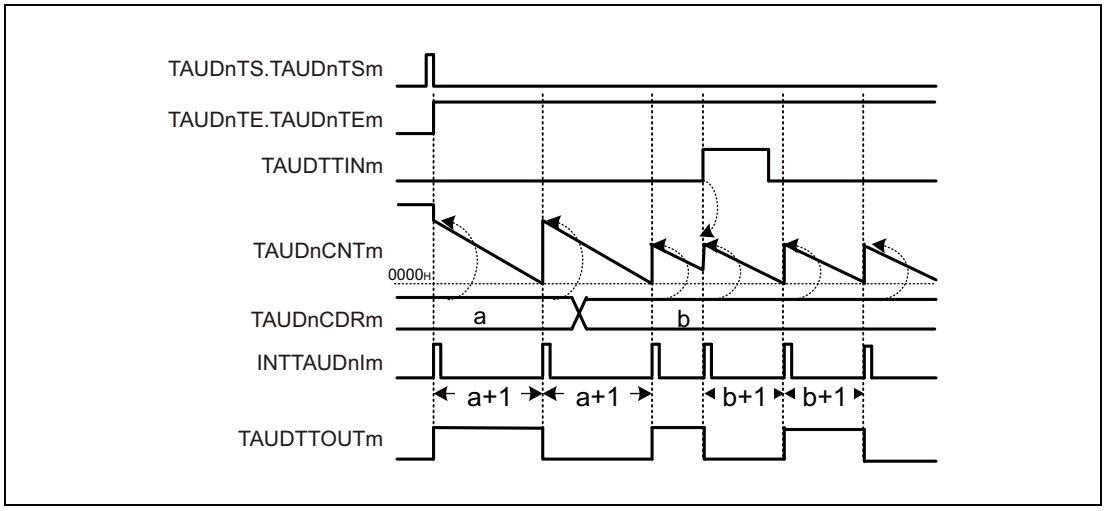


Figure 25.36 General Timing Diagram of TAUDTTINm Input Interval Timer Function

25.12.2.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.53 Contents of the TAUDnCMORm Register for TAUDTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS [1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated to toggle TAUDTTOUTm at the beginning of an operation.

(2) TAUDnCMURm

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

Table 25.54 Contents of the TAUDnCMURm Register for TAUDTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode

Table 25.55 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDTTOUTm can then be controlled independently of the interrupts. For details, see **Section 25.7, Channel Output Modes**.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm Input Interval Timer Function. Therefore, these registers should be set to 0.

Table 25.56 Simultaneous Rewrite Settings for TAUDTTINm Input Interval Timer Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.2.5 Operating Procedure for TAUDTTINm Input Interval Timer Function

Table 25.57 Operating Procedure for TAUDTTINm Input Interval Timer Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.53, Contents of the TAUDnCMORm Register for TAUDTTINm Input Interval Timer Function**, and **Table 25.54, Contents of the TAUDnCMURm Register for TAUDTTINm Input Interval Timer Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 25.55, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. When TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated and TAUDTTOUTm toggles.
During Operation	The values of the TAUDnCMURm.TAUDnTIS[1:0] and the TAUDnCDRm register are changeable at any time. The TAUDnCNTm register can be read at all times. Detection of TAUDTTINm edge	TAUDnCNTm counts down. When the counter reaches 0000_H: • The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. • INTTAUDnIm is generated and TAUDTTOUTm toggles. When a TAUDTTINm input valid edge is detected during count operation, the TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

25.12.2.6 Specific Timing Diagrams

The timing diagrams in **Section 25.12.1, Interval Timer Function** apply, and in addition the counter can also be restarted by an effective TAUDTTINm input edge.

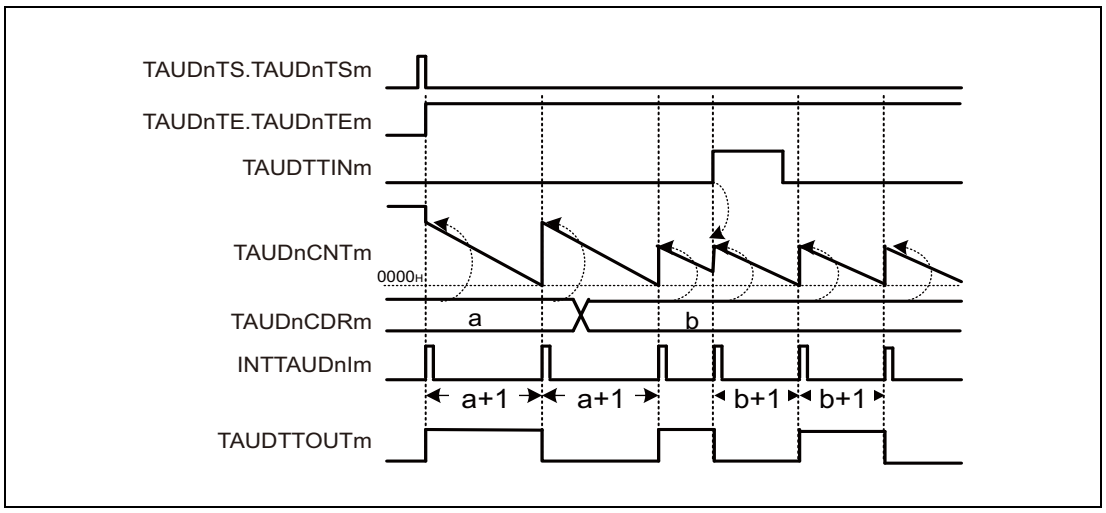


Figure 25.37 Counter Triggered by Rising TAUDTTINm Input Edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B), TAUDnCMORm.TAUDnMD0 = 1

- If an effective TAUDTTINm input edge is detected, an interrupt is generated which causes TAUDTTOUTm to toggle. In this example, the effective edge is a rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

25.12.3 Clock Divide Function

25.12.3.1 Overview

Summary

This function is used as a frequency divider. The frequency of the input signal TAUDTTINm is divided by a factor related to TAUDnCDRm, and the resulting signal is output to TAUDTTOUTm.

Prerequisites

- TAUDTTINm should have a fixed frequency.
- The operating mode should be set to interval timer mode. (See **Table 25.58, Contents of the TAUDnCMORm Register for Clock Divide Function**.)
- The channel output mode should be set to independent channel output mode 1. (See **Section 25.7, Channel Output Modes**.)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value, using TAUDTTINm as a count clock.

When the counter value reaches 0000_H, INTTAUDnIm occurs and TAUDTTOUTm signal is toggled. Then, TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.

The value of TAUDnCDRm can be rewritten at any time. The changed value of TAUDnCDRm is applied when the counter starts to count down next time.

The counter can be stopped by setting TAUDnTT.TAUDnTTm = 1. This sets TAUDnTE.TAUDnTEm = 0. TAUDnCNTm and TAUDTTOUTm stop but retain their values. The function can be restarted by setting TAUDnTS.TAUDnTSm = 1. The counter can also be forcibly restarted without making a stop by setting TAUDnTS.TAUDnTSm = 1 during operation (forced restart).

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUDTTOUTm does not toggle. This results in a negative TAUDTTOUTm signal compared to when TAUDnCMORm.TAUDnMD0 is set to 1. For details, see **Section 25.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

NOTE

TAUDTTINm input signals are sampled at the frequency of the operation clock set by TAUDnCMORm.TAUDnCKS[1:0] bits. Therefore, the TAUDTTOUTm output clock cycle has an error of ± 1 operation clock cycle.

25.12.3.2 Equations

- When rising edge detection is selected:
  TAUDTTOUTm frequency = TAUDTTINm frequency/[(TAUDnCDRm + 1) × 2]
- When falling edge detection is selected:
  TAUDTTOUTm frequency = TAUDTTINm frequency/[(TAUDnCDRm + 1) × 2]
- When falling and rising edge detection is selected:
  TAUDTTOUTm frequency = TAUDTTINm frequency/(TAUDnCDRm + 1)

25.12.3.3 Block Diagram and General Timing Diagram

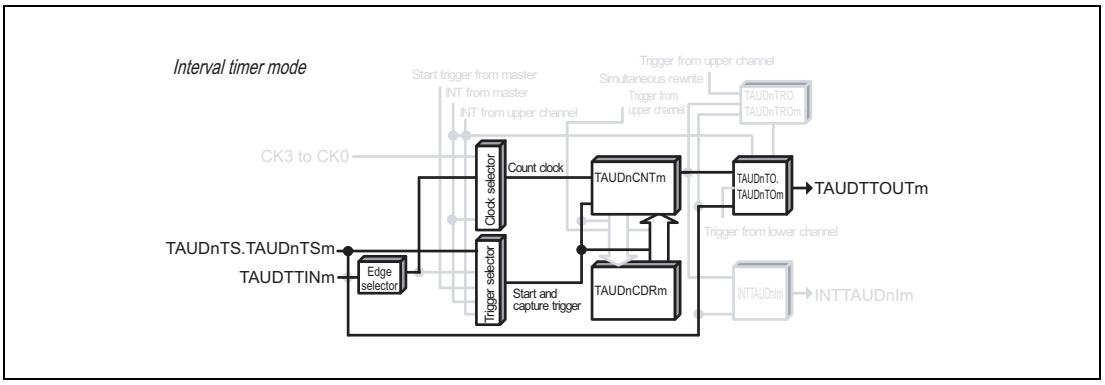


Figure 25.38 Block Diagram of Clock Divide Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

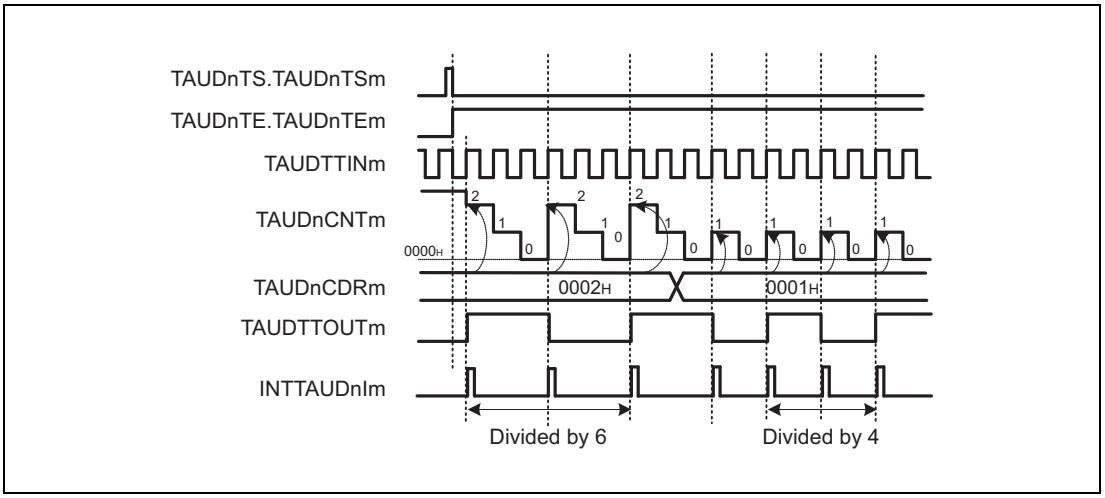


Figure 25.39 General Timing Diagram of Clock Divide Function

25.12.3.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.58 Contents of the TAUDnCMORm Register for Clock Divide Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	01: Valid TAUDTTINm input edge is used as a count clock.
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated and TAUDTTOUTm is toggled at the beginning of operation.

(2) TAUDnCMURm

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

Table 25.59 Contents of the TAUDnCMURm Register for Clock Divide Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode

Table 25.60 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the clock divide function. Therefore, these registers should be set to 0.

Table 25.61 Simultaneous Rewrite Settings for Clock Divide Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.3.5 Operating Procedure for Clock Divide Function

Table 25.62 Operating Procedure for Clock Divide Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.58, Contents of the TAUDnCMORm Register for Clock Divide Function**, and **Table 25.59, Contents of the TAUDnCMURm Register for Clock Divide Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 25.60, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm loads TAUDnCDRm value. If TAUDnCMORm.TAUDnMD0 is set to 1, INTTAUDnIm is generated and TAUDTTOUTm is toggled.
During Operation	The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down each time TAUDTTINm input edge is detected. When the counter reaches 0000_H: • TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. • INTTAUDnIm is generated. • TAUDTTOUTm is toggled. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm and TAUDTTOUTm retain their current values.

Restart Operation

25.12.3.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H

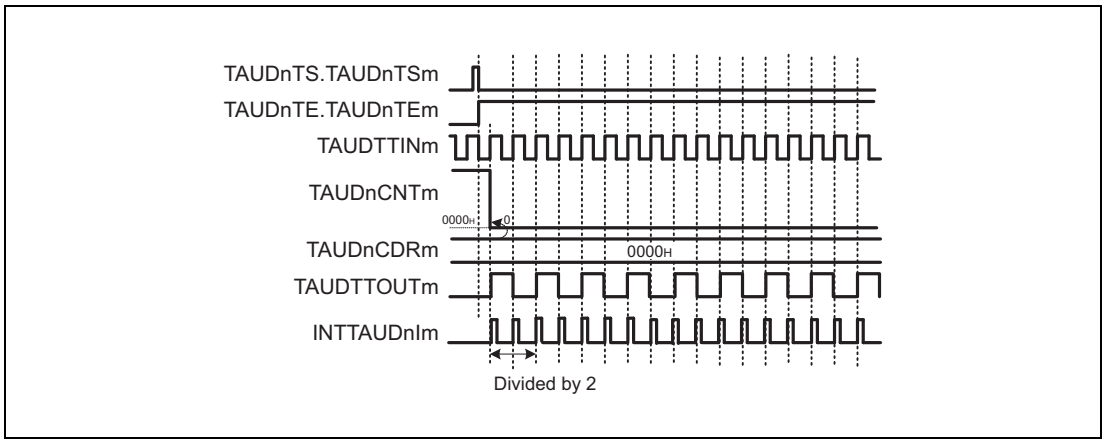


Figure 25.40 TAUDnCDRm = 0000_H, TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B

- If TAUDnCDRm is 0000_H, TAUDnCNTm is always 0000_H.
- INTTAUDnIm is generated every count clock, resulting in TAUDTTOUTm toggling every count clock.

Figure 25.40 shows an operation timing example. Actually, there is a delay from TINm detection until TOUTm output because of the delay time of a noise filter or synchronization circuit placed between the TAUDnIm pin and TAUDn.

(2) Operation restart

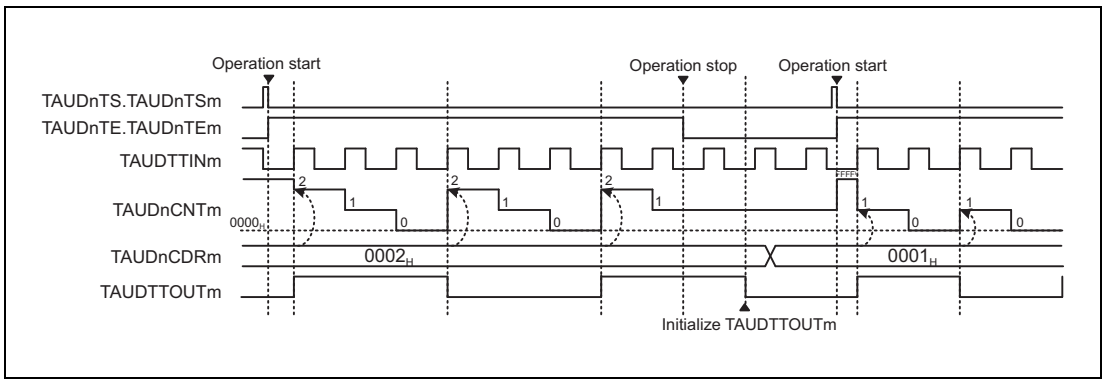


Figure 25.41 Operation Restart (TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

To reset the value of TAUDTTOUTm:

- Set TAUDnTOE.TAUDnTOEm = 0 when the counter is stopped (TAUDnTE.TAUDnTEm = 0).
- Then, write either 0 or 1 to TAUDnTO.TAUDnTOm to set the new start value of TAUDTTOUTm.

(3) Forced restart

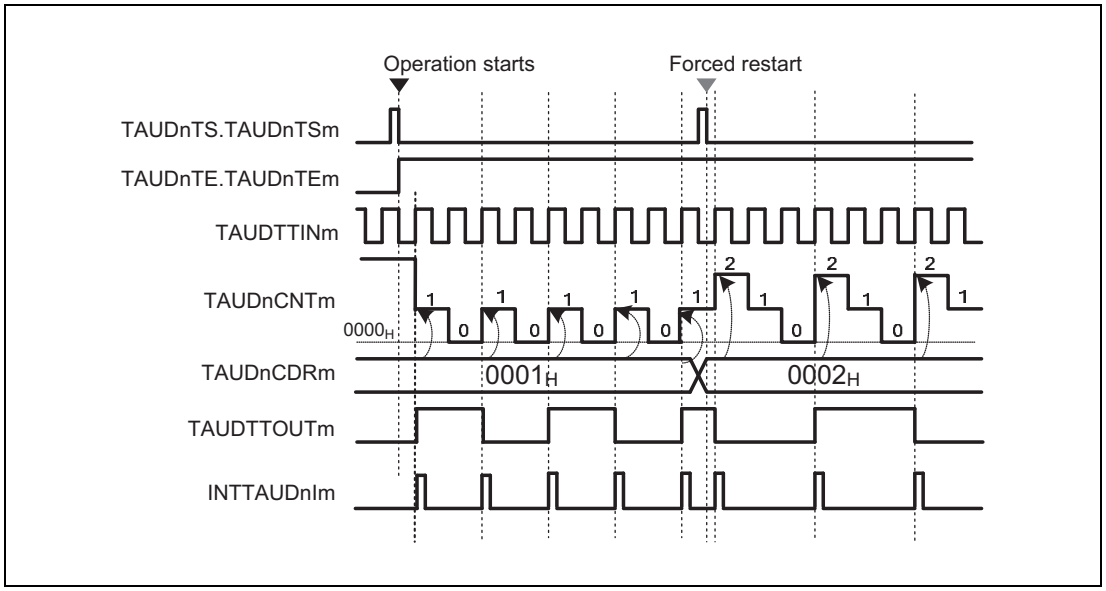


Figure 25.42 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

- The counter can be forcibly restarted (without stopping it first) by setting TAUDnTS.TAUDnTSm = 1 during operation.
- The value of TAUDnCDRm is written to TAUDnCNTm and the count operation restarts.
- TAUDTTOUTm restarts at the same level as before the forced restart.

25.12.4 External Event Count Function

25.12.4.1 Overview

Summary

This function is used as an event timer, which generates an interrupt (INTTAUDnIm) when a specific number of valid TAUDTTINm input edges are detected.

Prerequisites

- The operating mode should be set to the event count mode. (See **Table 25.63, Contents of the TAUDnCMORm Register for External Event Count Function**.)
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. When the counter starts, the current value of TAUDnCDRm is loaded into TAUDnCNTm.

When a valid TAUDTTINm input edge is detected, the value of TAUDnCNTm decrements by 1. TAUDnCNTm retains this value until a valid TAUDTTINm input edge is detected or the counter is restarted.

When the valid edge is detected for the (TAUDnCDRm + 1) times, INTTAUDnIm is generated. Then, TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUDnTS.TAUDnTSm to 1 during operation.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

An edge type used as a trigger is specified by TAUDnCMURm.TAUDnTIS[1:0] bits.

- When TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges are counted.
- When TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges are counted.
- When TAUDnCMURm.TAUDnTIS[1:0] = 10_B, both edges are counted.

25.12.4.2 Equations

Number of valid edges detected before INTTAUDnIm generation = TAUDnCDRm + 1

25.12.4.3 Block Diagram and General Timing Diagram

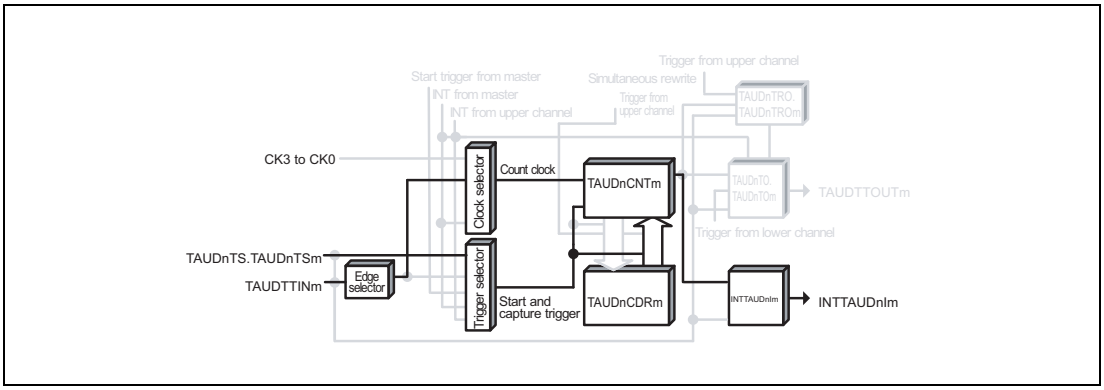


Figure 25.43 Block Diagram of External Event Count Function

The following settings apply to the general timing diagram.

- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

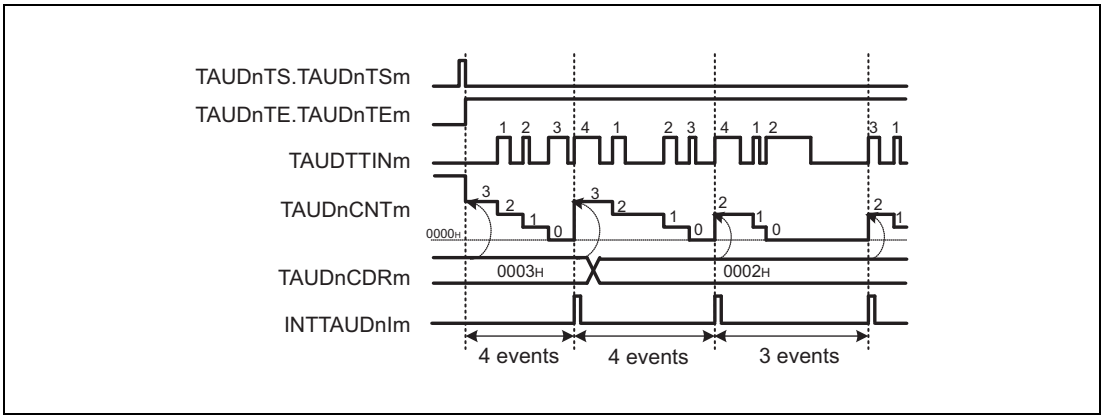


Figure 25.44 General Timing Diagram of External Event Count Function

25.12.4.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS [1:0]		TAUDnCCS [1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS [1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.63 Contents of the TAUDnCMORm Register for External Event Count Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	01: Valid TAUDTTINm input edge is used as a count clock.
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm

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

Table 25.64 Contents of the TAUDnCMURm Register for External Event Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Falling edge is detected. 01: Rising edge is detected. 10: Both edges are detected. 11: Setting prohibited

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the external event count function. Therefore, these registers should be set to 0.

Table 25.65 Simultaneous Rewrite Settings for External Event Count Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.4.5 Operating Procedure for External Event Count Function

Table 25.66 Operating Procedure for External Event Count Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.63, Contents of the TAUDnCMORm Register for External Event Count Function**, and **Table 25.64, Contents of the TAUDnCMURm Register for External Event Count Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm loads TAUDnCDRm value and waits for TAUDTTINm input edge detection.
During Operation	Detection of TAUDTTINm edge The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at any time.	TAUDnCNTm counts down each time TAUDTTINm input edge is detected. When effective edges are detected (TAUDnCDRm + 1) times: • TAUDnCDRm value is loaded in TAUDnCNTm and count operation continues. • INTTAUDnIm is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

25.12.4.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H

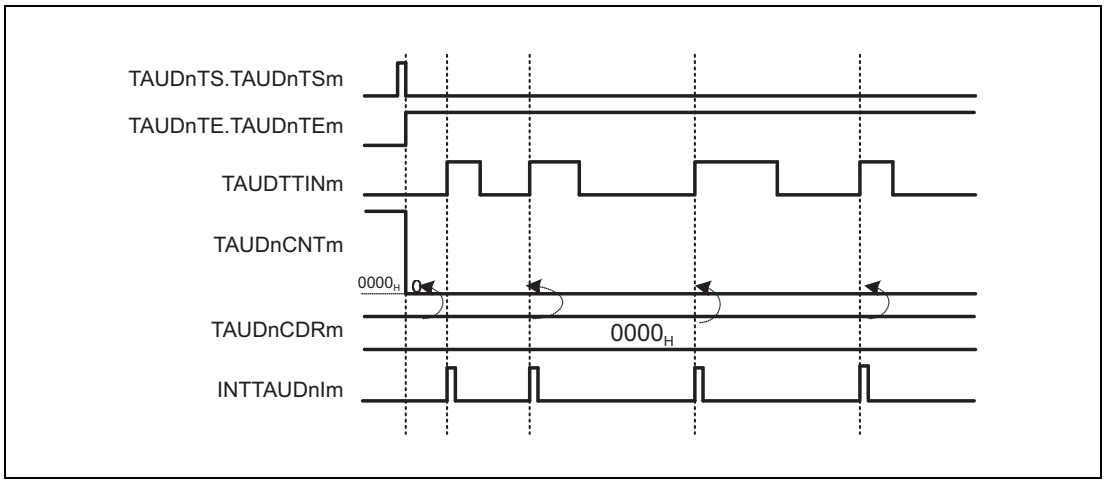


Figure 25.45 TAUDnCDRm = 0000_H, TAUDnCMURm.TAUDnTIS[1:0] = 01_B

- If 0000_H = TAUDnCDRm, 0000_H is loaded into TAUDnCNTm each time a valid TAUDTTINm input edge is detected.
  In other words, INTTAUDnIm is generated each time a valid TAUDTTINm input edge is detected.

(2) Operation stop and restart

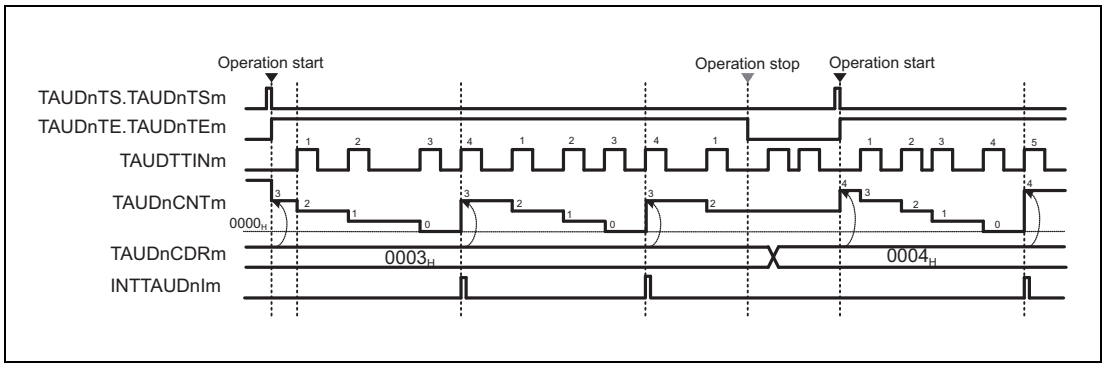


Figure 25.46 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm stops and retains its current value. TAUDTTINm continues and TAUDnCNTm ignores the valid edge.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm loads the TAUDnCDRm value and restarts count operation.

(3) Forced restart

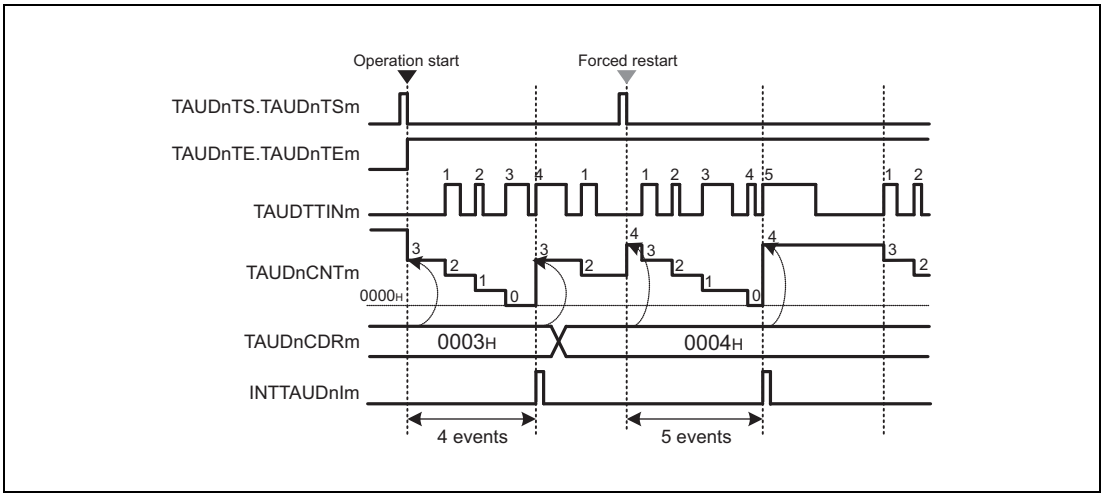


Figure 25.47 Forced Restart Operation (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

Once a forced restart is made, the changed TAUDnCDRm value is applied to TAUDnCNTm immediately.

- The counter can be restarted without making a stop by setting TAUDnTS.TAUDnTSm to 1 during operation.
- The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter awaits the next valid TAUDTTINm input edge.

25.12.5 Delay Count Function

25.12.5.1 Overview

Summary

This function generates interrupts (INTTAUDnIm), which have a defined delay to the TAUDTTINm input signal. TAUDTTINm input signal pulses that occur within the delay period are ignored.

Prerequisites

- The operating mode should be set to one-count mode. See **Table 25.67, Contents of the TAUDnCMORm Register for Delay Count Function**.
- TAUDTTOUTm is not used with this function.
- Trigger detection should be disabled during counting (TAUDnCMORn.TAUDnMD0 = 0).

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value.

When the counter reaches 0000_H, an interrupt is generated. The counter returns to $FFFF_H$ and awaits the next valid TAUDTTINm input edge.

When the counter is counting down, further TAUDTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as a trigger is specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

25.12.5.2 Equations

Delay between TAUDTTINm and INTTAUDnIm = count clock cycle × (TAUDnCDRm + 1)

25.12.5.3 Block Diagram and General Timing Diagram

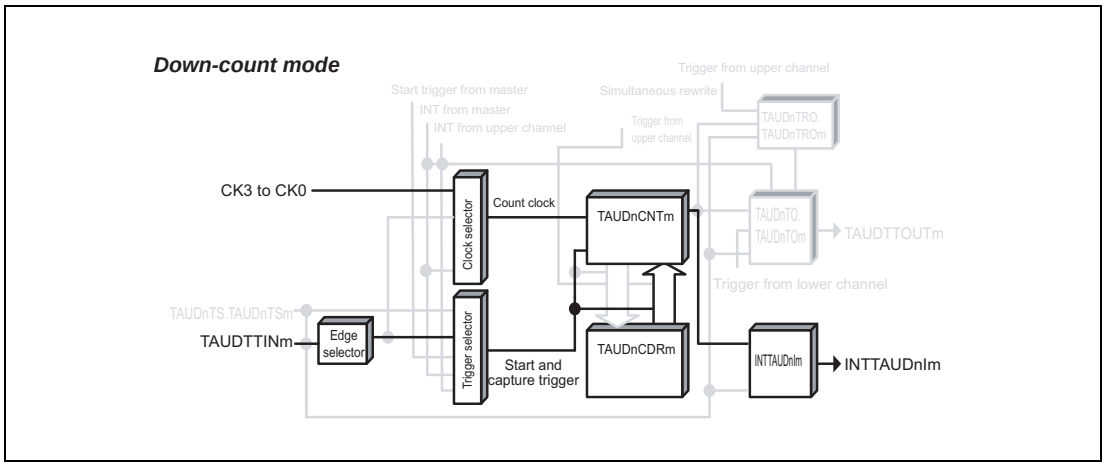


Figure 25.48 Block Diagram of Delay Count Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

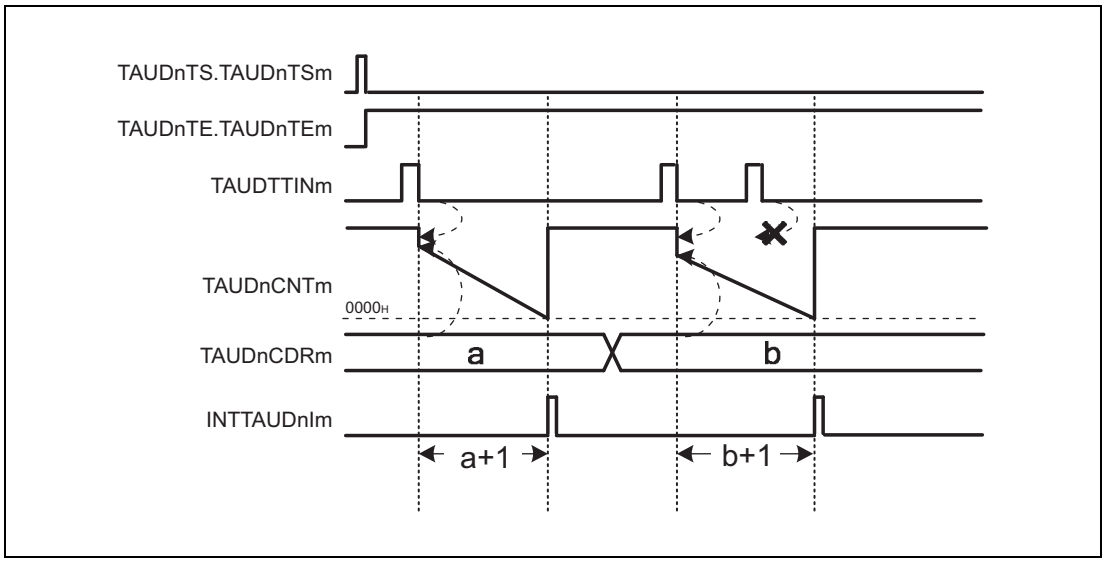


Figure 25.49 General Timing Diagram of Delay Count Function

25.12.5.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.67 Contents of the TAUDnCMORm Register for Delay Count Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	0: Disables a start trigger during operation

(2) TAUDnCMURm

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

Table 25.68 Contents of the TAUDnCMURm Register for Delay Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the delay count function. Therefore, these registers should be set to 0.

Table 25.69 Simultaneous Rewrite Settings for Delay Count Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.5.5 Operating Procedure for Delay Count Function

Table 25.70 Operating Procedure for Delay Count Function

		Operation	TAUDn Status
	Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.67, Contents of the TAUDnCMORm Register for Delay Count Function**, and **Table 25.68, Contents of the TAUDnCMURm Register for Delay Count Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Restart Operation	Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge. When a start edge is detected, the TAUDnCDRm value is loaded in TAUDnCNTm.
	During Operation	The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down. When the counter reaches 0000_H. INTTAUDnIm is generated. TAUDnCNTm stops counting, returns $FFFF_H$, and waits for a trigger. If a trigger occurs while TAUDnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
	Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its value.

25.12.6 One-Pulse Output Function

25.12.6.1 Overview

Summary

This function generates an interrupt (INTTAUDnIm) when a valid TAUDTTINm input edge is detected and at a defined interval afterward. TAUDTTINm input signal pulses that occur within the defined interval are ignored. When an interrupt is generated, the TAUDTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode should be set to pulse one-count mode. (See **Table 25.71, Contents of the TAUDnCMORm Register for One-Pulse Output Function**.)
- The channel output mode should be set to independent channel output mode 2. (See **Section 25.7, Channel Output Modes**.)
- Trigger detection should be disabled during counting (TAUDnCMORn.TAUDnMD0 = 0).

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value. An interrupt is generated and TAUDTTOUTm is set to active level.

When the counter reaches 0001_H, an interrupt is generated and TAUDTTOUTm is set to the inactive level. The counter stops at 0000_H and awaits the next valid TAUDTTINm input edge.

When the counter is counting down, further TAUDTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as a trigger is specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

25.12.6.2 Equations

Interval between TAUDTTINm and INTTAUDnIm = TAUDTTOUTm (timer output) width = count clock cycle × TAUDnCDRm

25.12.6.3 Block Diagram and General Timing Diagram

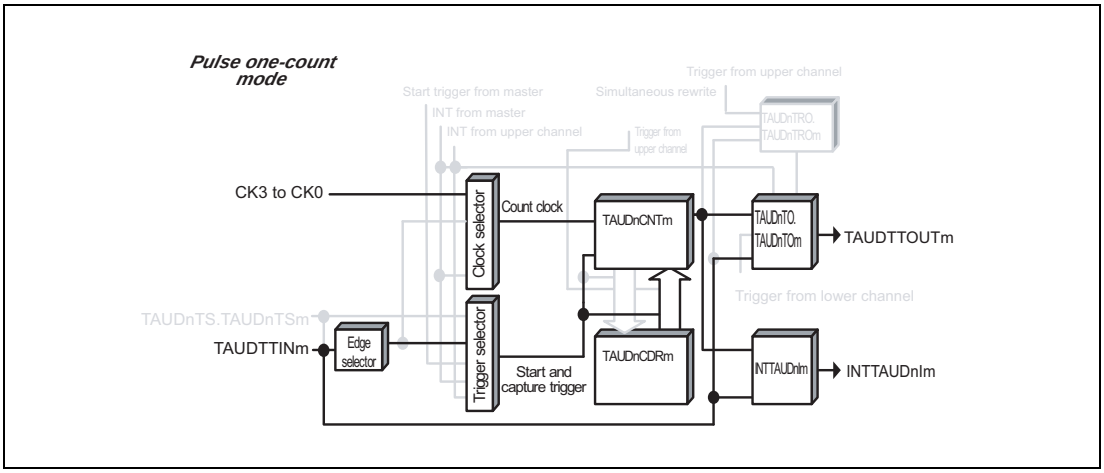


Figure 25.50 Block Diagram of One-Pulse Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

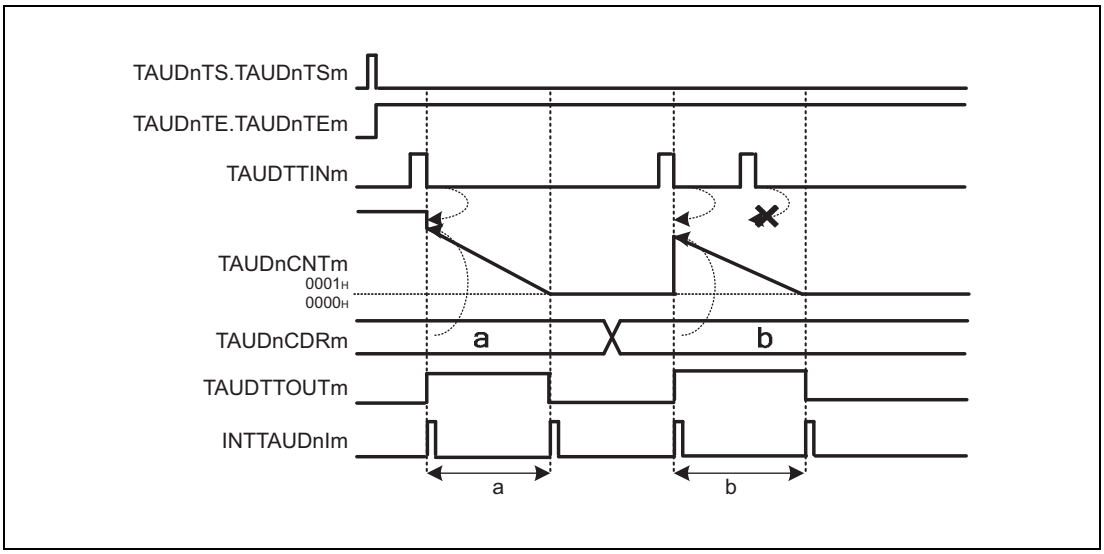


Figure 25.51 General Timing Diagram of One-Pulse Output Function

25.12.6.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.71 Contents of the TAUDnCMORm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMD0	0: Disables a start trigger during operation.

(2) TAUDnCMURm

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

Table 25.72 Contents of the TAUDnCMURm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode

Table 25.73 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode controlled by software.
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operation mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUDnTOE.TAUDnTOEm = 0. TAUDTTOUTm can then be controlled independently of the interrupts. For details, see **Table 25.47, Channel Output Modes**.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the One-Pulse Output Function. Therefore, these registers should be set to 0.

Table 25.74 Simultaneous Rewrite Settings for One-Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.6.5 Operating Procedure for One-Pulse Output Function

Table 25.75 Operating Procedure for One-Pulse Output Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.71, Contents of the TAUDnCMORm Register for One-Pulse Output Function**, and **Table 25.72, Contents of the TAUDnCMURm Register for One-Pulse Output Function**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 25.73, Control Bit Settings in Independent Channel Output Mode 2**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge. When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value.
During Operation	The value of TAUDnCDRm is changeable at any time. The TAUDnCNTm register can be read at all times.	INTTAUDnIm is generated when TAUDnCNTm starts and TAUDTTOUTm is set to its active level. TAUDnCNTm counts down. When the counter reaches 0001_H: • INTTAUDnIm is generated. • TAUDTTOUTm is set to its inactive level. TAUDnCNTm stops counting and waits for a trigger. If a trigger occurs while TAUDnCNTm is counting, the trigger is ignored.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

25.12.7 TAUDTTINm Input Pulse Interval Measurement Function

25.12.7.1 Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUDnCSRm.TAUDnOVF to measure the interval of the TAUDTTINm input signals.

Prerequisites

- The operating mode should be set to capture mode. See **Table 25.77, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Measurement Function**.
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter TAUDnCNTm starts to count up from 0000_H. When a valid TAUDTTINm edge is detected, the value of TAUDnCNTm is captured, transferred to TAUDnCDRm, and an interrupt INTTAUDnIm is generated. The counter resets to 0000_H and subsequently continues operation.

If the counter reaches $FFFF_H$ before a valid TAUDTTINm edge is detected, it overflows to 0000_H. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUDnCDRm and TAUDnCSRm.TAUDnOVF respectively depend on the values of bits TAUDnCMORm.TAUDnCOS[1:0].

Table 25.76 Effects of Overflow

TAUDnCMORm.TAUDnCOS[1:0]	When Overflow Occurs		When a Valid TAUDTTINm Input is Detected	
	TAUDnCDRm	TAUDnCSRm.TAUDnOVF	TAUDnCDRm, TAUDnCNTm	TAUDnCSRm.TAUDnOVF
00	Unchanged	0	TAUDnCNTm loaded into TAUDnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUDnCNTm set to 0, TAUDnCDRm unchanged	Unchanged
11		1		

When TAUDnCMORm.TAUDnCOS[0] = 1, the overflow bit (TAUDnCSRm.TAUDnOVF) can be cleared only by setting TAUDnCSCm.TAUDnCLOV = 1.

The combination of the value of TAUDnCDRm and TAUDnCSRm.TAUDnOVF can be used to deduce the interval of the TAUDTTINm signal. However, if an overflow occurs multiple times before a valid TAUDTTINm input is detected, the overflow bit TAUDnCSRm.TAUDnOVF cannot indicate the occurrence of multiple overflows.

The function can be stopped by setting TAUDnTT.TAUDnTTm = 1. This sets TAUDnTE.TAUDnTEm = 0. TAUDnCNTm stops but retains its value. While the function is stopped, valid TAUDTTINm input edge detection and TAUDnCNTm capture are not performed.

The counter is reset to 0000_H and subsequently continues operation.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 25.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

NOTE

When TAUDnCMORm.TAUDnCOS[1:0] = 10_B or 11_B, the value of TAUDnCNTm is not loaded into TAUDnCDRm when the first valid TAUDTTINm input edge occurs after an overflow. However, an interrupt is generated.

25.12.7.2 Equations

TAUDTTINm input pulse interval = count clock cycle × [(TAUDnCSRm.TAUDnOVF × ($FFFF_H$ + 1)) + TAUDnCDRm capture value + 1]

25.12.7.3 Block Diagram and General Timing Diagram

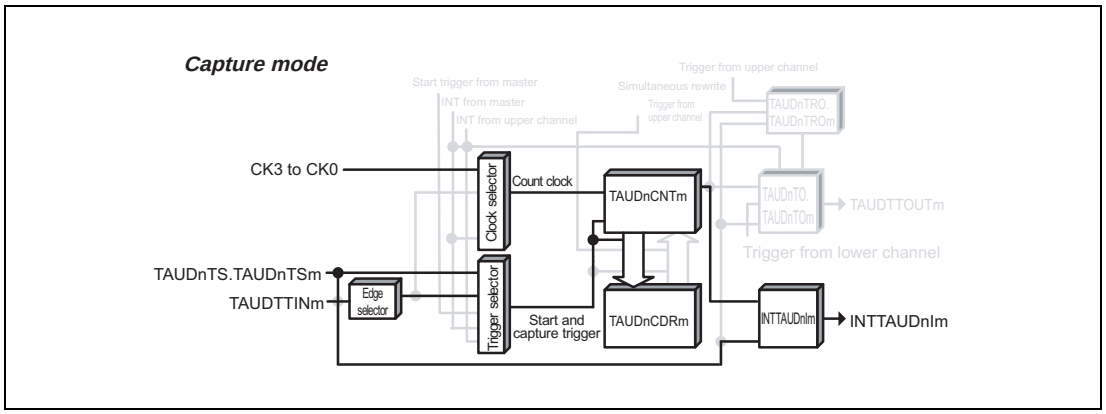


Figure 25.52 Block Diagram of TAUDTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 0).
- Falling edge detection (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)
- When a valid TAUDTTINm input is detected after an overflow, TAUDnCDRm is changed and TAUDnCSRm.TAUDnOVF is set to 1 (TAUDnCMORm.TAUDnCOS[1:0] = 00_B).

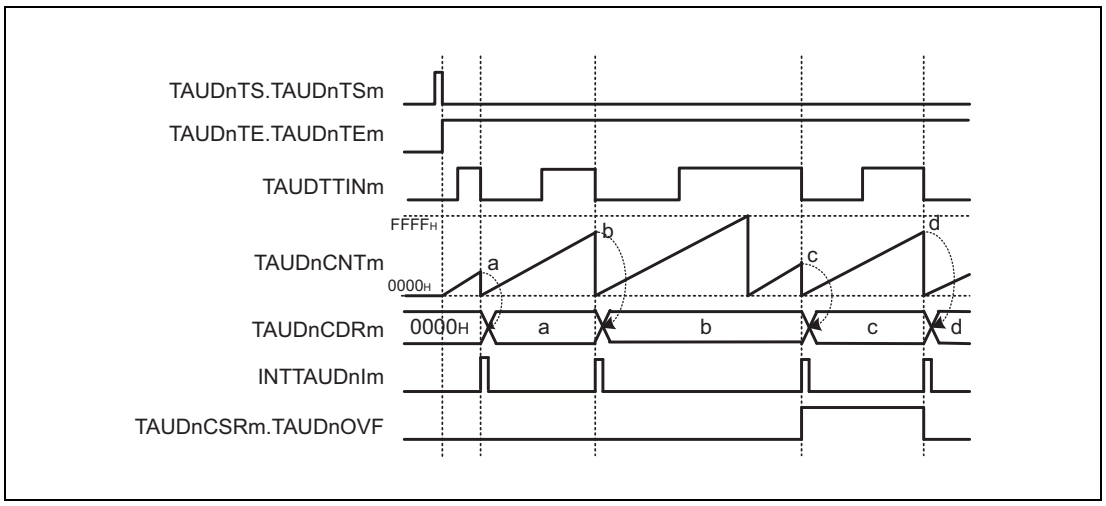


Figure 25.53 General Timing Diagram of TAUDTTINm Input Pulse Interval Measurement Function

25.12.7.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.77 Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as the external capture trigger.
7, 6	TAUDnCOS[1:0]	See **Table 25.76, Effects of Overflow**.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm

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

Table 25.78 Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input pulse interval measurement function. Therefore, these registers should be set to 0.

Table 25.79 Simultaneous Rewrite Settings for TAUDTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.7.5 Operating Procedure for TAUDTTINm Input Pulse Interval Measurement Function

Table 25.80 Operating Procedure for TAUDTTINm Input Pulse Interval Measurement Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.77, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Measurement Function**, and **Table 25.78, Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Measurement Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000_H. INTTAUDnIm is generated when TAUDnCMORm.TAUDnMD0 is set to 1.
During Operation	Detection of TAUDTTINm edge The values of TAUDnCMURm.TAUDnTIS[1:0] bits can be changed at any time. The TAUDnCDRm and TAUDnCSRm registers can be read at any time. TAUDnCSCm.TAUDnCLOV can be written to 1. (TAUDnCSRm.TAUDnOVF bit is cleared to 0.)	TAUDnCNTm starts to count up from 0000_H. When a TAUDTTINm valid edge is detected: • TAUDnCNTm transfers (captures) its value to TAUDnCDRm, and returns to 0000_H. • INTTAUDnIm is then generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and both it and TAUDnCSRm.TAUDnOVF retain their current values.

Restart Operation

25.12.7.6 Specific Timing Diagrams: Overflow Operation

(1) TAUDnCMORm.TAUDnCOS[1:0] = 00_B

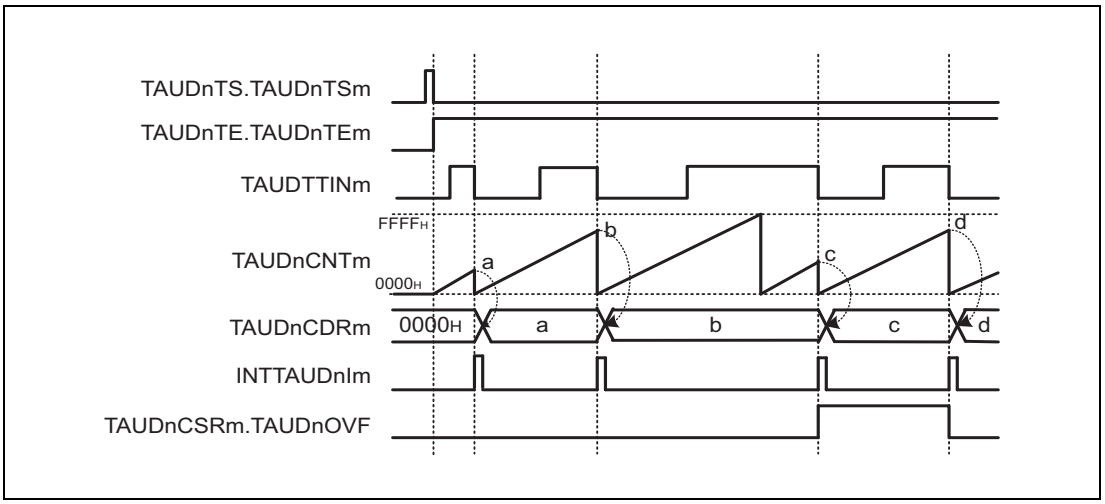


Figure 25.54 TAUDnCMORm.TAUDnCOS[1:0] = 00_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge with no overflow occurring, TAUDnCSRm.TAUDnOVF is cleared to 0.

(2) TAUDnCMORm.TAUDnCOS[1:0] = 01_B

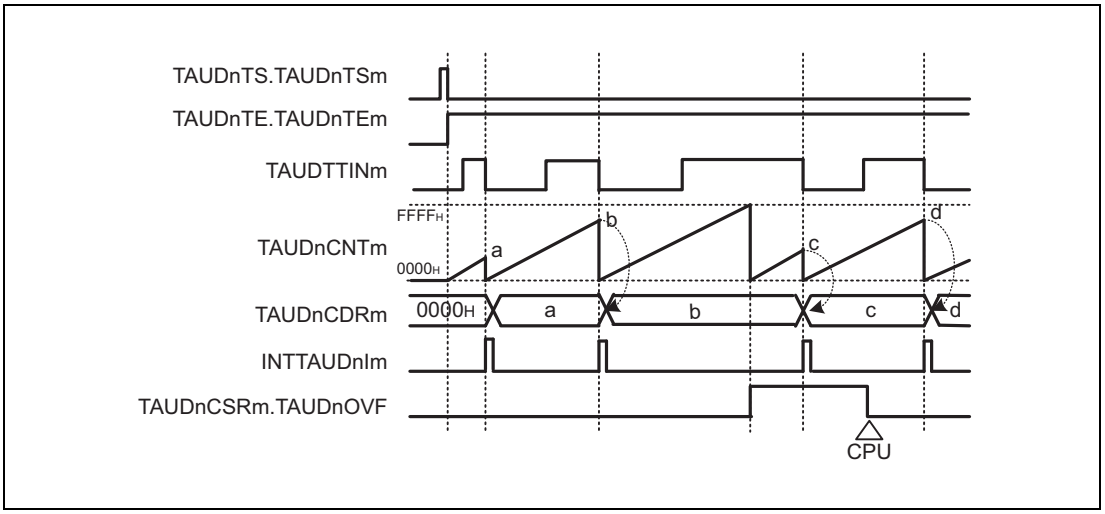


Figure 25.55 TAUDnCMORm.TAUDnCOS[1:0] = 01_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF is set to 1.

- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm.
- TAUDnCSRm.TAUDnOVF is only cleared by a CPU command (by setting TAUDnCSCm.TAUDnCLOV bit to 1).

(3) TAUDnCMORm.TAUDnCOS[1:0] = 10_B

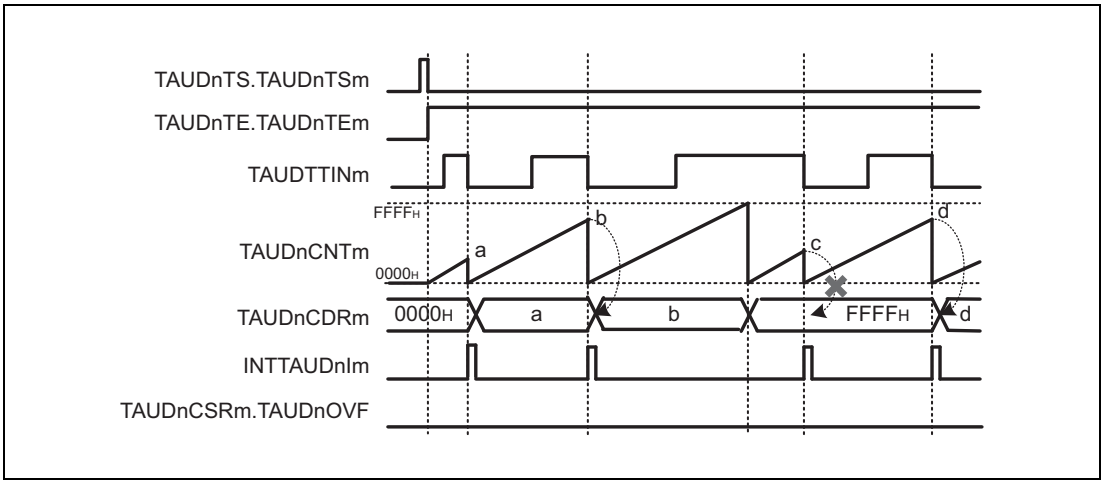


Figure 25.56 TAUDnCMORm.TAUDnCOS[1:0] = 10_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm is reset to 0, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.

(4) TAUDnCMORm.TAUDnCOS[1:0] = 11_B

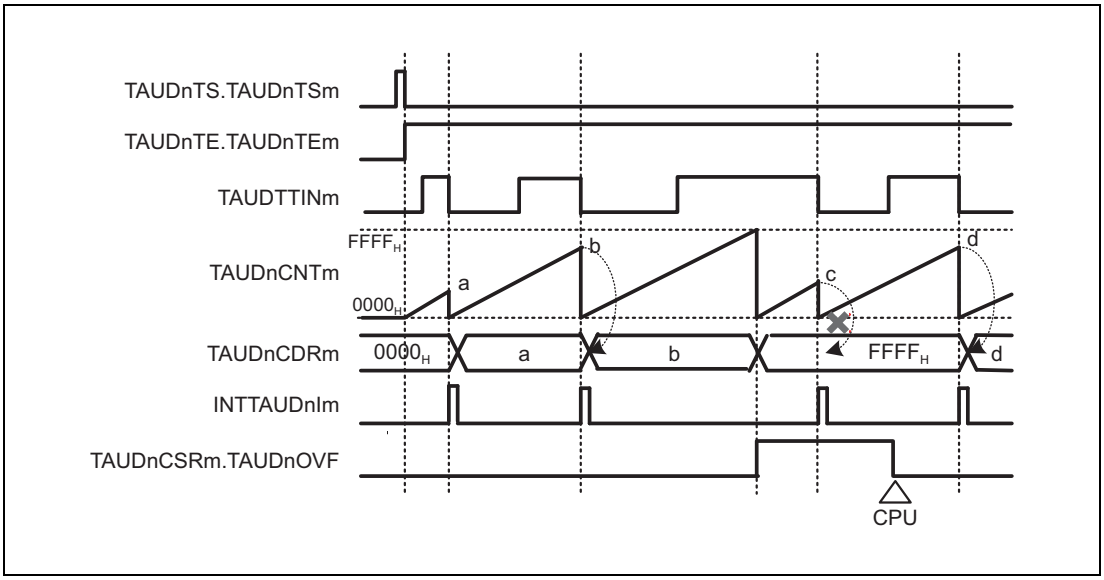


Figure 25.57 TAUDnCMORm.TAUDnCOS[1:0] = 11_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm is reset to 0, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.
- TAUDnCSRm.TAUDnOVF is cleared by setting TAUDnCSCm.TAUDnCLOV to 1.

25.12.8 TAUDTTINm Input Signal Width Measurement Function

25.12.8.1 Overview

Summary

This function measures the width of a TAUDTTINm signal, by starting the count at one edge of TAUDTTINm and capturing the count value at the other edge.

Prerequisites

- The operating mode should be set to capture and one-count mode. See **Table 25.82, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Measurement Function**.
- TAUDTTOUTm is not used with this function.
- TAUDnCMORm.TAUDnMD0 should be set to 0.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. When a valid TAUDTTINm start edge is detected, the counter TAUDnCNTm starts to count up from 0000_H. When a valid TAUDTTINm stop edge is detected, the value of TAUDnCNTm is captured, transferred to TAUDnCDRm, and an interrupt INTTAUDnIm is generated. The counter retains its value (TAUDnCDRm +1) and awaits the next valid TAUDTTINm input start edge.

If the counter reaches $FFFF_H$ before a valid TAUDTTINm stop edge is detected, it overflows. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUDnCDRm and TAUDnCSRm.TAUDnOVF respectively depend on the values of bits TAUDnCMORm.TAUDnCOS[1:0].

Table 25.81 Effects of Overflow

TAUDnCMORm.TAUDnCOS[1:0]	When Overflow Occurs		When a Valid TAUDTTINm Input Stop Edge is Detected	
	TAUDnCDRm	TAUDnCSRm.TAUDnOVF	TAUDnCDRm, TAUDnCNTm	TAUDnCSRm.TAUDnOVF
00	Unchanged	0	TAUDnCNTm loaded into TAUDnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUDnCNTm stops counting TAUDnCDRm unchanged	Unchanged
11		1		

When TAUDnCMORm.TAUDnCOS[0] = 1, overflow bit TAUDnCSRm.TAUDnOVF can be cleared only by setting TAUDnCSCm.TAUDnCLOV to 1.

The combination of the value of TAUDnCDRm and TAUDnCSRm.TAUDnOVF can be used to deduce the width of the TAUDTTINm signal. However, if an overflow occurs multiple times before a valid TAUDTTINm input is detected, overflow bit TAUDnCSRm.TAUDnOVF cannot indicate the occurrence of multiple overflows.

This function cannot be forcibly restarted.

NOTE

When TAUDnCMORm.TAUDnCOS[1] = 1, the value of TAUDnCNTm is not loaded to TAUDnCDRm when the first valid TAUDTTINm input edge occurs after an overflow. However, an interrupt is generated.

25.12.8.2 Equations

TAUDTTINm input signal width = count clock cycle × [(TAUDnCSRm.TAUDnOVF × ($FFFF_H$ + 1)) + TAUDnCDRm capture value + 1]

25.12.8.3 Block Diagram and General Timing Diagram

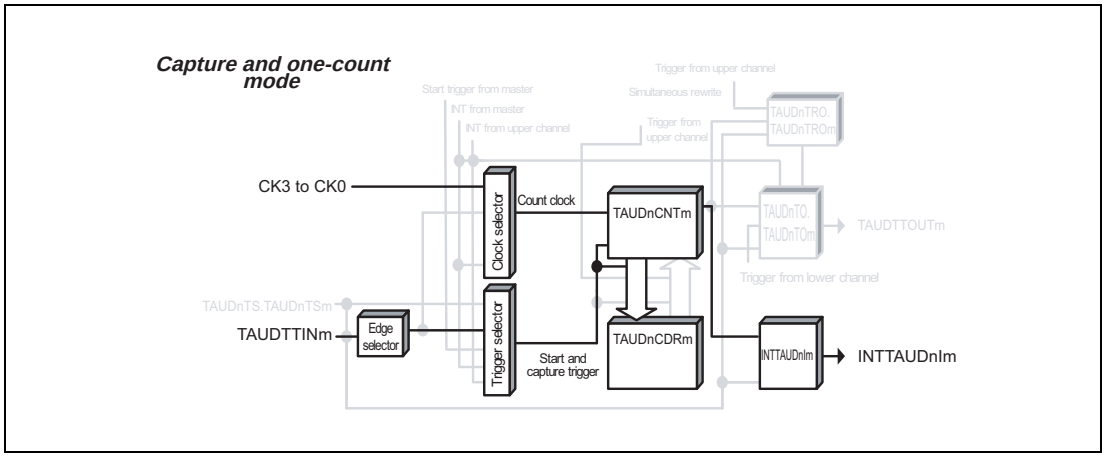


Figure 25.58 Block Diagram of TAUDTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)
- When a valid TAUDTTINm input is detected after an overflow, TAUDnCDRm is changed and TAUDnCSRm.TAUDnOVF is set to 1. (TAUDnCMORm.TAUDnCOS[1:0] = 00_B)

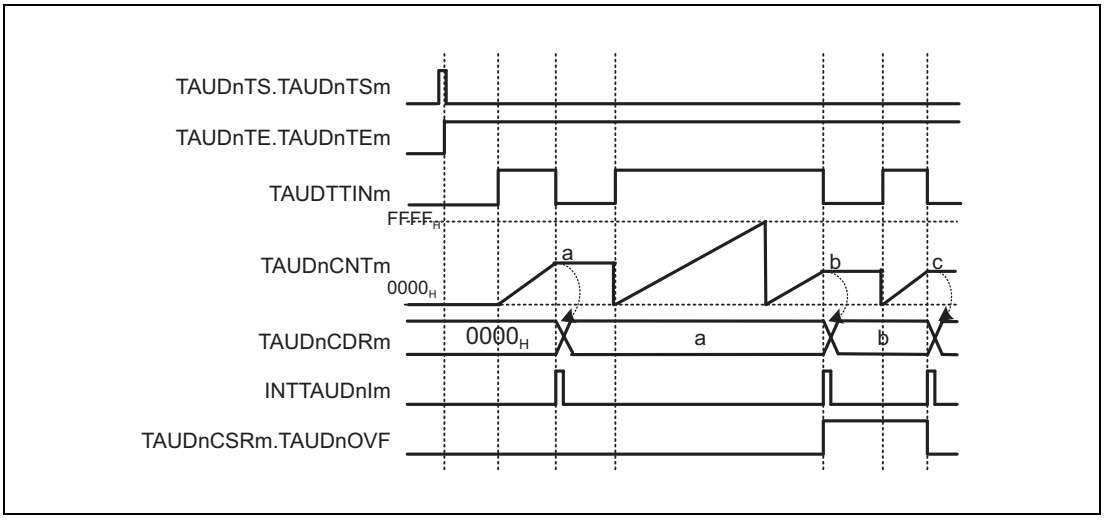


Figure 25.59 General Timing Diagram of TAUDTTINm Input Signal Width Measurement Function

25.12.8.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.82 Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	See **Table 25.81, Effects of Overflow**.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0110: Capture and one-count mode
0	TAUDnMD0	0: Disables the start trigger during operation.

(2) TAUDnCMURm

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

Table 25.83 Contents of the TAUDnCMURm Register For TAUDTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input signal width measurement function. Therefore, these registers should be set to 0.

Table 25.84 Simultaneous Rewrite Settings for TAUDTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.8.5 Operating Procedure for TAUDTTINm Input Signal Width Measurement Function

Table 25.85 Operating Procedure for TAUDTTINm Input Signal Width Measurement Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.82, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Measurement Function**, and **Table 25.83, Contents of the TAUDnCMURm Register For TAUDTTINm Input Signal Width Measurement Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge. When a TAUDTTINm start edge is detected, TAUDnCNTm starts to count up.
During Operation	TAUDnCDRm, TAUDnCNTm, and TAUDnCSRm registers can be read at any time. TAUDnCSCm.TAUDnCLOV bit can be set to 1.	TAUDnCNTm starts to count up from 0000_H. When TAUDTTINm valid edge is detected: • TAUDnCNTm transfers (captures) its value to TAUDnCDRm, and retains its value. INTTAUDnIm is then generated. Counting stops at the “value that transferred to TAUDnCDRm + 1” and TAUDnCNTm waits for detection of the TAUDTTINm start edge. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and both it and TAUDnCSRm.TAUDnOVF retain their current values.

Restart Operation

25.12.8.6 Specific Timing Diagrams: Overflow Operation

(1) TAUDnCMORm.TAUDnCOS[1:0] = 00_B

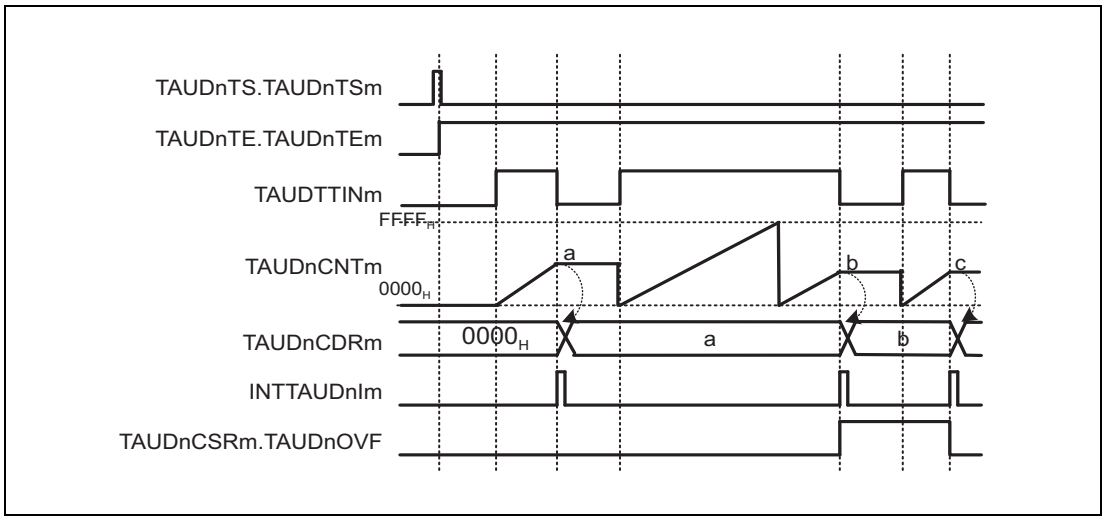


Figure 25.60 TAUDnCMORm.TAUDnCOS[1:0] = 00_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge with no overflow occurring, TAUDnCSRm.TAUDnOVF is cleared to 0.

(2) TAUDnCMORm.TAUDnCOS[1:0] = 01_B

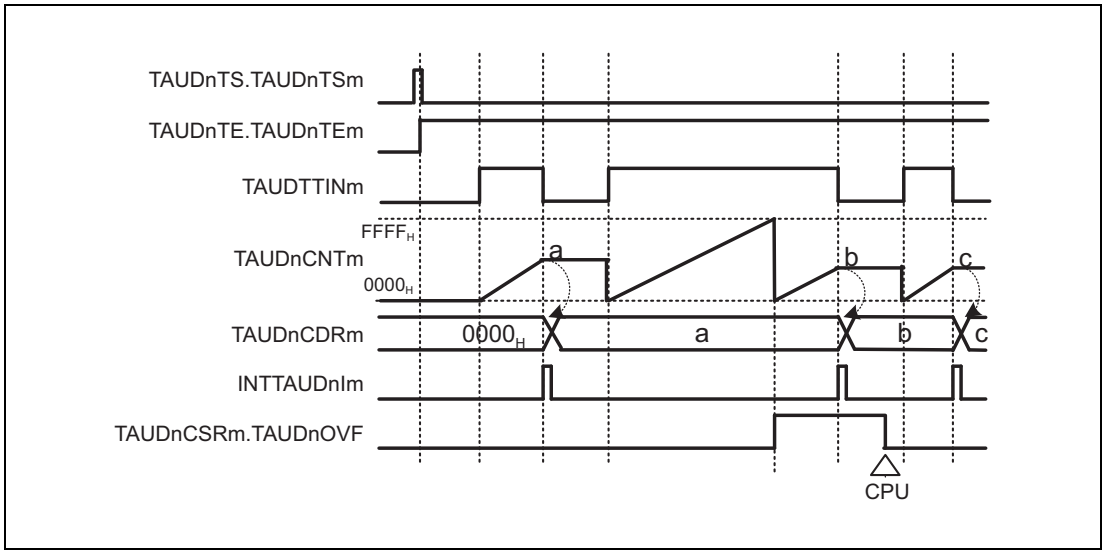


Figure 25.61 TAUDnCMORm.TAUDnCOS[1:0] = 01_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge, the value of TAUDnCNTm is loaded into TAUDnCDRm.
- TAUDnCSRm.TAUDnOVF is only cleared by a CPU command (by setting TAUDnCSCm.TAUDnCLOV bit to 1).

(3) TAUDnCMORm.TAUDnCOS[1:0] = 10_B

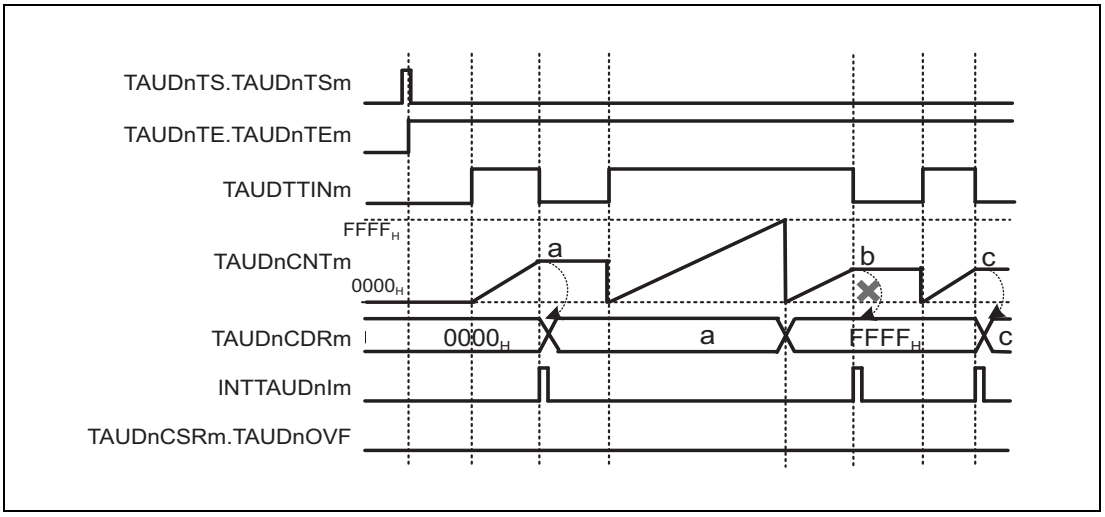


Figure 25.62 TAUDnCMORm.TAUDnCOS[1:0] = 10_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF remains 0.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm stops counting, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.

(4) TAUDnCMORm.TAUDnCOS[1:0] = 11_B

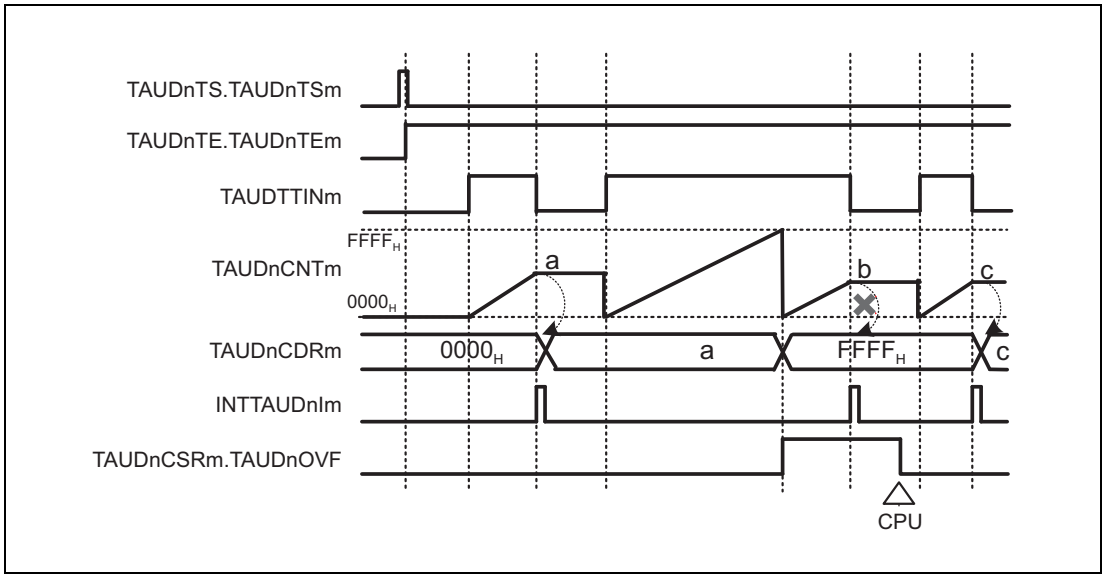


Figure 25.63 TAUDnCMORm.TAUDnCOS[1:0] = 11_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- When an overflow occurs, TAUDnCDRm is set to $FFFF_H$ and TAUDnCSRm.TAUDnOVF is set to 1.
- Upon detection of the next valid TAUDTTINm input edge, TAUDnCNTm stops counting, but TAUDnCDRm and TAUDnCSRm.TAUDnOVF remain unchanged.
- Thus, the next valid TAUDTTINm input edge after the overflow is ignored.
- TAUDnCSRm.TAUDnOVF is cleared by setting TAUDnCSCm.TAUDnCLOV to 1.

25.12.9 TAUDTTINm Input Position Detection Function

25.12.9.1 Overview

Summary

This function measures the input signal duration by capturing the count value at the valid edge of TAUDTTINm.

Prerequisites

- The operating mode should be set to count capture mode. (See **Table 25.86, Contents of the TAUDnCMORm Register for TAUDTTINm Input Position Detection Function**.)
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter starts counting from 0000_H. When a valid TAUDTTINm input edge is detected, the current value of TAUDnCNTm is loaded into TAUDnCDRm and an interrupt (INTTAUDnIm) is generated. The count operation continues.

When the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTE

The TAUDTTINm input signal is sampled at the frequency of the operation clock, specified by TAUDnCMORm.TAUDnCKS[1:0] bits. As a result, the output cycle of TAUDTTOUTm has an error of ± 1 operation clock cycle.

Conditions

If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt does not occur at the beginning of operation or after restart. For details, see **Section 25.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

25.12.9.2 Equations

Functional duration at a TAUDTTINm input pulse =

count clock cycle × (TAUDnCDRm capture value + 1)

25.12.9.3 Block Diagram and General Timing Diagram

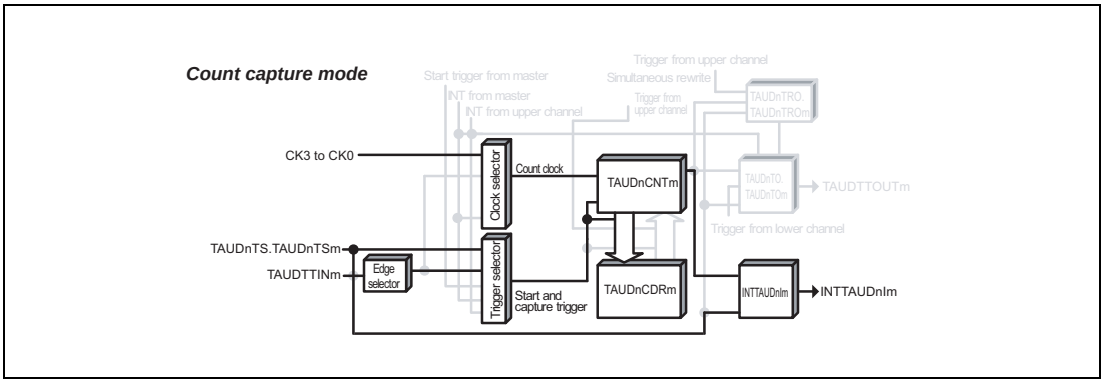


Figure 25.64 Block Diagram of TAUDTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

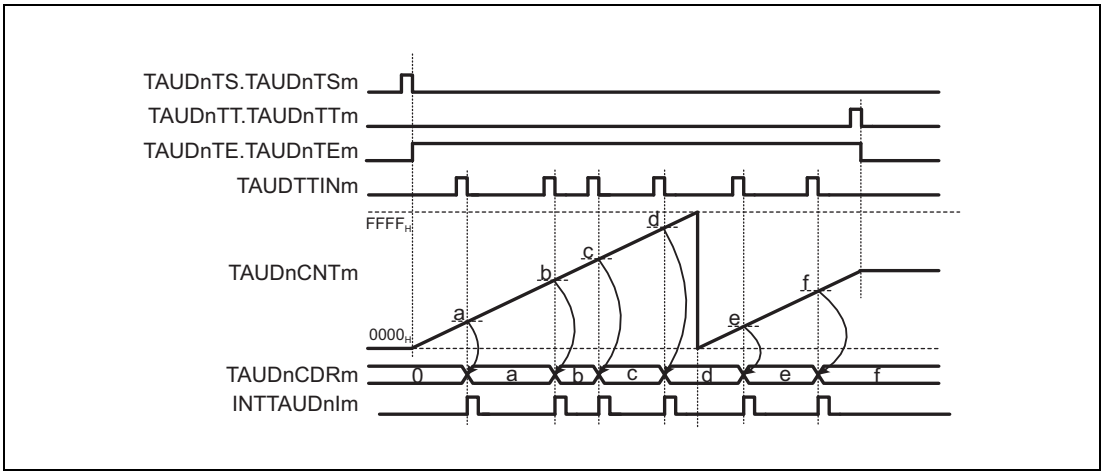


Figure 25.65 General Timing Diagram of TAUDTTINm Input Position Detection Function

25.12.9.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.86 Contents of the TAUDnCMORm Register for TAUDTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as an external capture trigger.
7, 6	TAUDnCOS[1:0]	01: Set to this value.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1011: Count capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm

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

Table 25.87 Contents of the TAUDnCMURm Register for TAUDTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input position detection function. Therefore, these registers should be set to 0.

Table 25.88 Simultaneous Rewrite Settings for TAUDTTINm Input Position Detection Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.9.5 Operating Procedure for TAUDTTINm Input Position Detection Function

Table 25.89 Operating Procedure for TAUDTTINm Input Position Detection Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.86, Contents of the TAUDnCMORm Register for TAUDTTINm Input Position Detection Function**, and **Table 25.87, Contents of the TAUDnCMURm Register for TAUDTTINm Input Position Detection Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm occurs.
During Operation	The values of TAUDnCMURm.TAUDnTIS[1:0] bits can be changed at any time. The TAUDnCDRm and TAUDnCSRm registers can be read at any time.	TAUDnCNTm starts to count up from 0000_H. When a valid TAUDTTINm edge is detected: • TAUDnCNTm transfers (captures) its own value to TAUDnCDRm. • Outputs INTTAUDnIm. • The counter is not cleared to 0000_H and TAUDnCNTm continues counting. Afterwards, this procedure is repeated. When TAUDnCNTm reaches $FFFF_H$, the counter restarts from 0000_H.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart

25.12.9.6 Specific Timing Diagrams

(1) Operation stop and restart

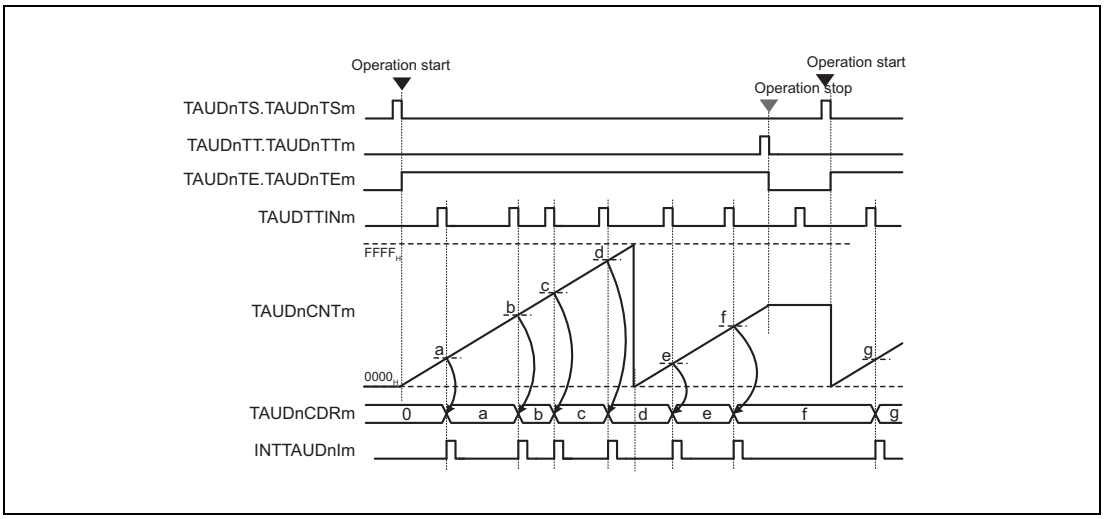


Figure 25.66 Operation Stop and Restart (TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- The counter can stop operating by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm stops and retains its current value.
- If the counter stops operating, valid TAUDTTINm input edges are ignored.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm restarts to count from 0000_H.

25.12.10 TAUDTTINm Input Period Count Detection Function

25.12.10.1 Overview

Summary

This function measures the cumulative width of a TAUDTTINm input signal.

Prerequisites

- The operating mode should be set to capture and gate count mode. (See **Table 25.90, Contents of the TAUDnCMORm Register for TAUDTTINm Input Period Count Detection Function**.)
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter awaits a valid TAUDTTINm input edge.

When a valid TAUDTTINm input start edge is detected, the counter starts to count from 0000_H.

When a valid TAUDTTINm input stop edge is detected, the current TAUDnCNTm value is loaded into TAUDnCDRm and an interrupt (INTTAUDnIm) is generated. The counter stops and retains its value (TAUDnCDRm + 1) until the next valid TAUDTTINm input start edge is detected.

When the next valid TAUDTTINm input start edge is detected, the counter restarts to count from the value retained when stopped.

If the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTES

1. TAUDTTINm input signal is sampled at the frequency of an operation clock set by the TAUDnCMORm.TAUDnCKS[1:0] bits.
2. As this function is to measure the TAUDTTINm input signal width, setting TAUDnTS.TAUDnTSm to 1 is disabled while TAUDnTE.TAUDnTEm = 1.

Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, the TAUDTTINm input low period is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] = 11_B, the TAUDTTINm input high period is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

25.12.10.2 Equations

Cumulative TAUDTTINm input width =
count clock cycle × (TAUDnCDRm capture value + 1)

25.12.10.3 Block Diagram and General Timing Diagram

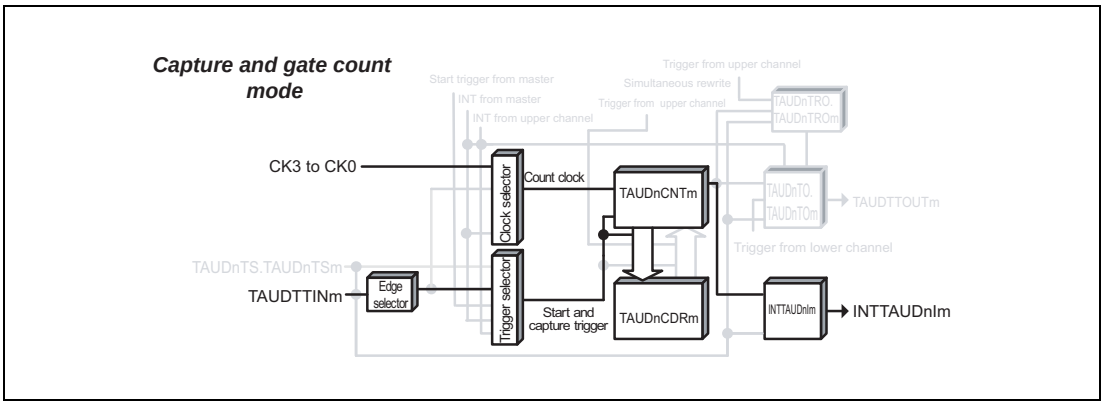


Figure 25.67 Block Diagram of TAUDTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

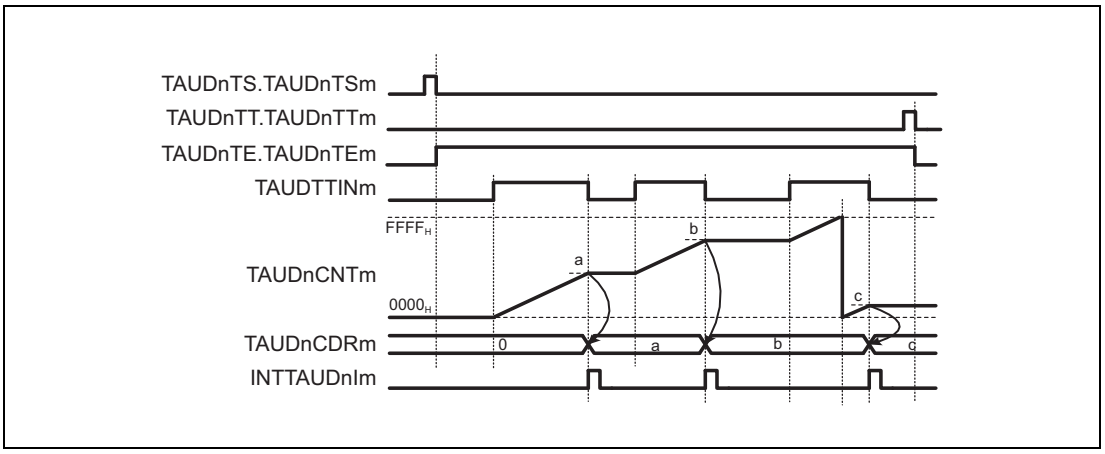


Figure 25.68 General Timing Diagram of TAUDTTINm Input Period Count Detection Function

25.12.10.4 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.90 Contents of the TAUDnCMORm Register for TAUDTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	01: Set to this value.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1101: Capture and gate count mode
0	TAUDnMD0	0: Disables the start trigger during operation.

(2) TAUDnCMURm

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

Table 25.91 Contents of the TAUDnCMURm Register for TAUDTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input period count detection function. Therefore, these registers should be set to 0.

Table 25.92 Simultaneous Rewrite Settings for TAUDTTINm Input Period Count Detection Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.10.5 Operating Procedure for TAUDTTINm Input Period Count Detection Function

Table 25.93 Operating Procedure for TAUDTTINm Input Period Count Detection Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.90, Contents of the TAUDnCMORm Register for TAUDTTINm Input Period Count Detection Function**, and **Table 25.91, Contents of the TAUDnCMURm Register for TAUDTTINm Input Period Count Detection Function**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge.
During Operation	Detection of TAUDTTINm edge The TAUDnCDRm, TAUDnCNTm, and TAUDnCSRm registers can be read at any time.	When a TAUDTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUDnCNTm starts counting up from the stop value. When TAUDnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUDnCDRm and INTTAUDnIm is generated. Counting stops at the “value transferred to TAUDnCDRm + 1” and TAUDnCNTm waits for detection of the TAUDTTINm start edge. When TAUDnCNTm reaches $FFFF_H$, the counter restarts from 0000_H. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

25.12.10.6 Specific Timing Diagrams

(1) Operation stop and restart

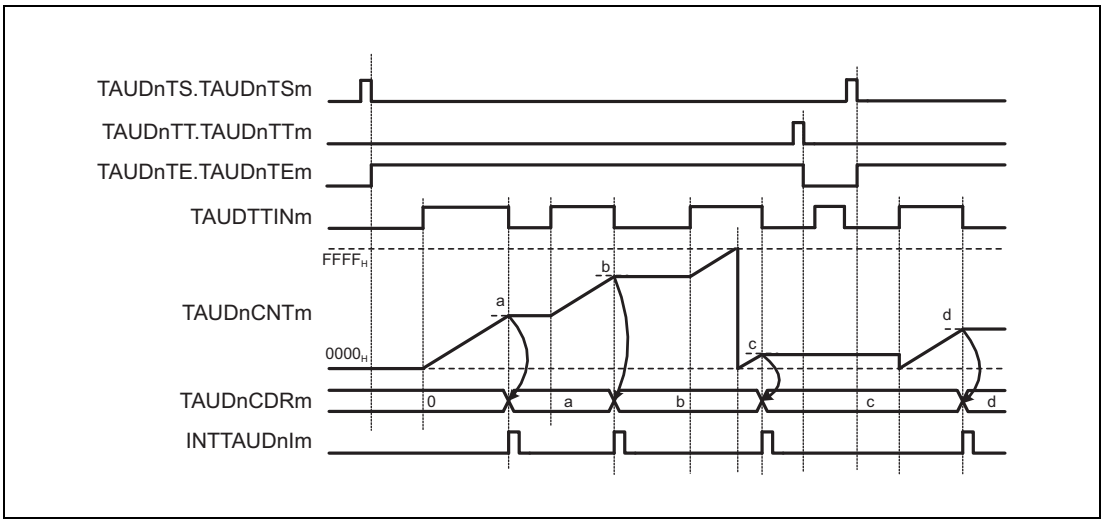


Figure 25.69 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm stops and retains its current value.
- If the counter is stopped, valid TAUDTTINm input edges are ignored.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1. TAUDnCNTm restarts to count from 0000_H.

25.12.11 TAUDTTINm Input Pulse Interval Judgment Function

25.12.11.1 Overview

Summary

This function outputs the result of a comparison between the count value (TAUDnCNTm) and the value in the channel data register (TAUDnCDRm) when a TAUDTTINm input pulse occurs. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true.

Prerequisites

- The operating mode should be set to judge mode. See **Table 25.94, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Judgment Function**.
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value.

When a TAUDTTINm valid edge is detected or TAUDnTS.TAUDnTSm is set to 1, the function compares the current values of TAUDnCNTm and TAUDnCDRm. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true. TAUDnCNTm reloads the value of TAUDnCDRm and subsequently continues operation, regardless of the result of the comparison.

If the counter reaches 0000_H before a TAUDTTINm valid edge is detected, TAUDnCNTm overflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

The TAUDnCMORm.TAUDnMD0 bit specifies the type of comparison:

- If TAUDnCMORm.TAUDnMD0 = 0, INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm.
- If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm.

25.12.11.2 Block Diagram and General Timing Diagram

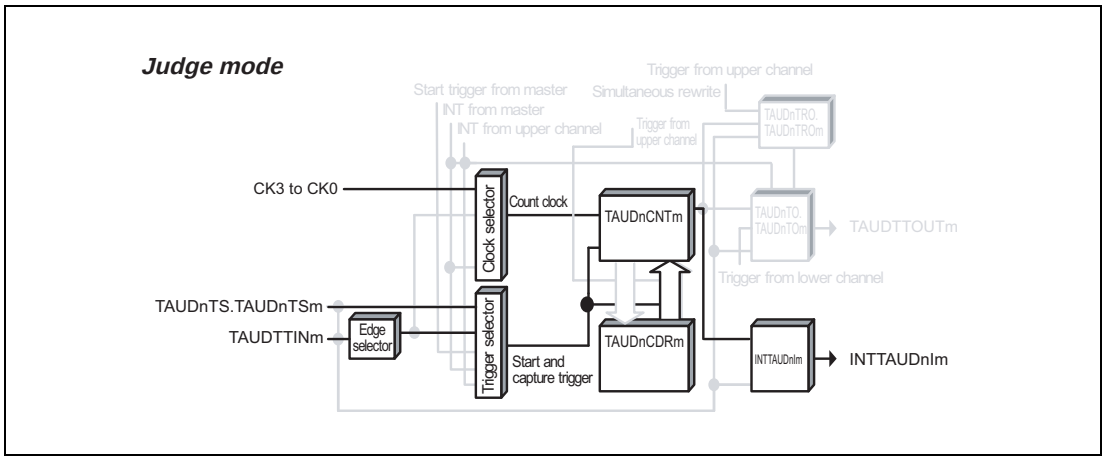


Figure 25.70 Block Diagram of TAUDTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

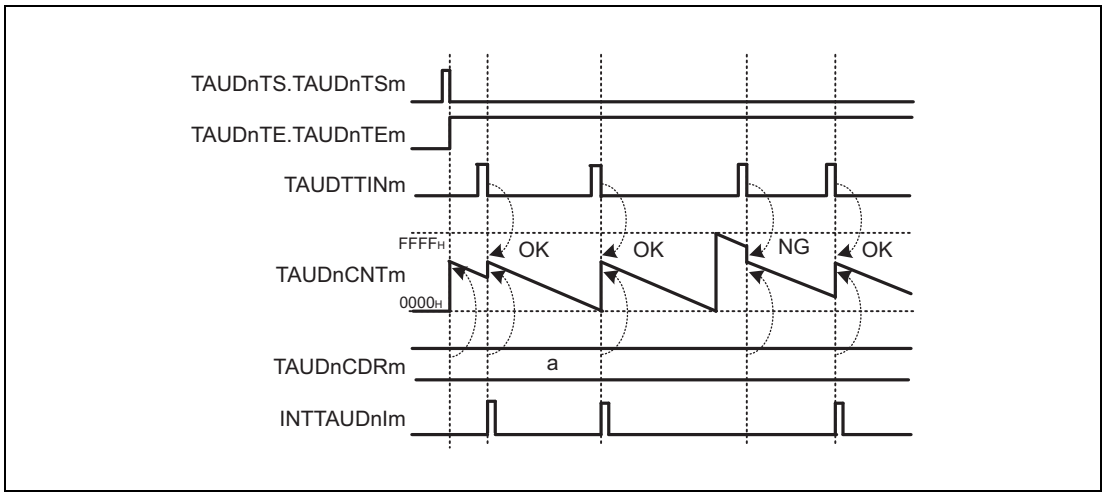


Figure 25.71 General Timing Diagram of TAUDTTINm Input Pulse Interval Judgment Function

25.12.11.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.94 Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0001: Judge mode
0	TAUDnMD0	0: INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm 1: INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm

(2) TAUDnCMURm

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

Table 25.95 Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input pulse interval judgment function. Therefore, these registers should be set to 0.

Table 25.96 Simultaneous Rewrite Settings for TAUDTTINm Input Pulse Interval Judgment Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.11.4 Operating Procedure for TAUDTTINm Input Pulse Interval Judgment Function

Table 25.97 Operating Procedure for TAUDTTINm Input Pulse Interval Judgment Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.94, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Judgment Function**, and **Table 25.95, Contents of the TAUDnCMURm Register for TAUDTTINm Input Pulse Interval Judgment Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCDRm value is loaded into TAUDnCNTm.
During Operation	The following register can be changed at any time: • TAUDnCDRm register	When TAUDnCMORm.TAUDnMD0 = 0 If TAUDnCNTm ≤ TAUDnCDRm when a TAUDTTINm input edge is detected, INTTAUDnIm is generated. When TAUDnCMORm.TAUDnMD0 = 1 If TAUDnCNTm > TAUDnCDRm when a TAUDTTINm input edge is detected, INTTAUDnIm is generated. If a TAUDTTINm input edge is detected, then TAUDnCNTm starts to count down from the value of TAUDnCDRm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

25.12.12 TAUDTTINm Input Signal Width Judgment Function

25.12.12.1 Overview

Summary

This function compares the count value (TAUDnCNTm) for the high or low level width of a TAUDTTINm input signal and the TAUDnCDRm value, and outputs the judgment result from the interrupt request signal INTTAUDnIm.

Prerequisites

- The operating mode should be set to judge and one-count mode. (See **Table 25.98, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Judgment Function**.)
- TAUDTTOUTm is not used with this function.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation. When a valid TAUDTTINm input start edge is detected, the current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value.

When a TAUDTTINm valid stop edge is detected, the function compares the current values of TAUDnCNTm and TAUDnCDRm. An interrupt request signal INTTAUDnIm is generated if the result of the comparison is true. The counter TAUDnCNTm retains its value until the next valid TAUDTTINm start edge is detected, regardless of the result of the comparison.

If the counter reaches 0000_H before a valid TAUDTTINm stop edge is detected, TAUDnCNTm overflows and is set to $FFFF_H$. The counter then continues to count down.

The value of TAUDnCDRm can be rewritten at any time, and the changed value of TAUDnCDRm is applied the next time the counter starts to count down.

Conditions

- The TAUDnCMORm.TAUDnMD0 bit specifies the type of comparison:
  - If TAUDnCMORm.TAUDnMD0 = 0, INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm.
  - If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm.
- The TAUDnCMURm.TAUDnTIS[1:0] bits specify a type of width measurement:
  - For high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B), TAUDTTINm rising edge is used as a start edge and TAUDTTINm falling edge as a stop edge.
  - For low width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 10_B), TAUDTTINm falling edge is used as a start edge and TAUDTTINm rising edge as a stop edge.
- This function cannot make a forced restart.

25.12.12.2 Block Diagram and General Timing Diagram

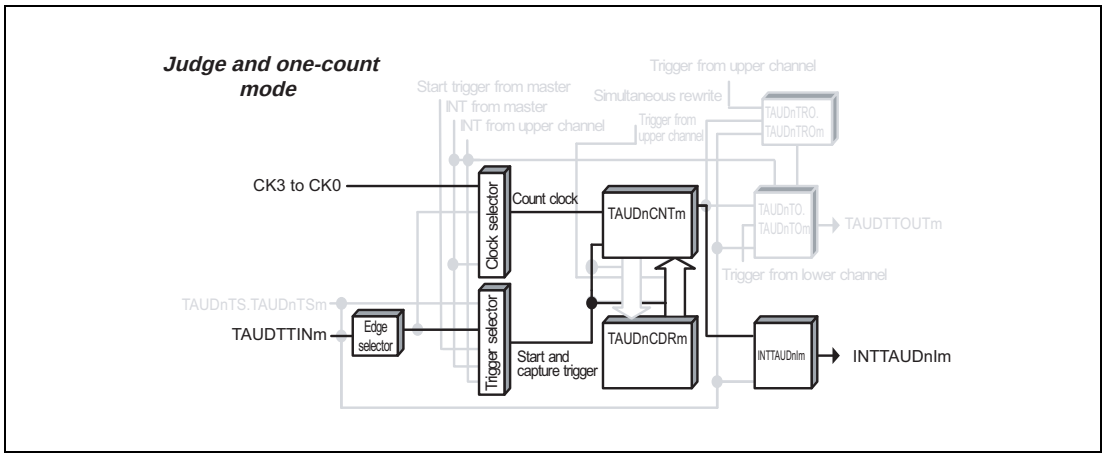


Figure 25.72 Block Diagram of TAUDTTINm Input Signal Width Judgment Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm (TAUDnCMORm.TAUDnMD0 = 0).
- TAUDTTINm valid start edge = rising edge, TAUDTTINm valid stop edge = falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

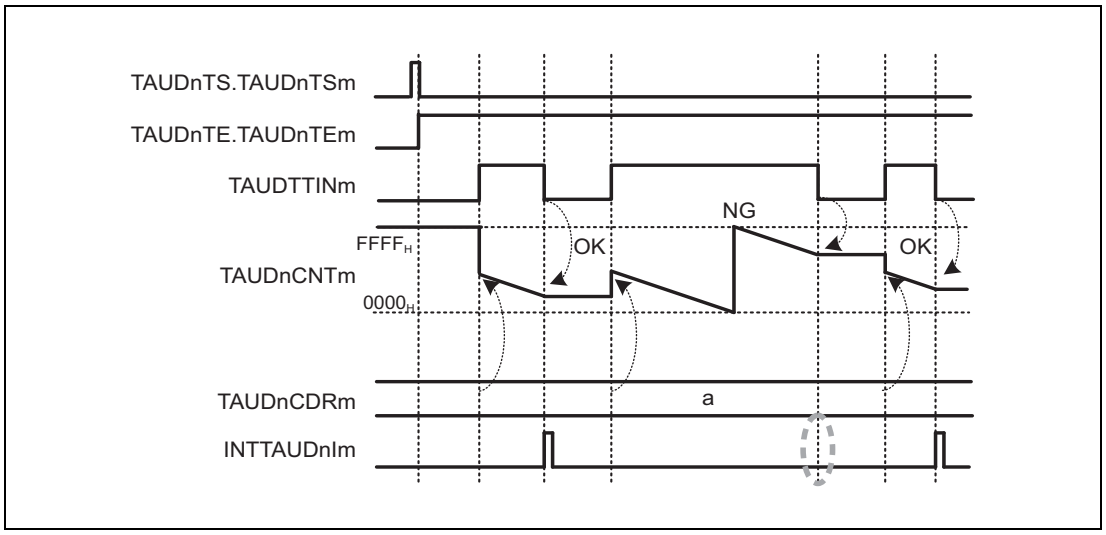


Figure 25.73 General Timing Diagram of TAUDTTINm Input Signal Width Judgment Function

25.12.12.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.98 Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0111: Judge and one-count mode
0	TAUDnMD0	0: INTTAUDnIm is generated when TAUDnCNTm ≤ TAUDnCDRm 1: INTTAUDnIm is generated when TAUDnCNTm > TAUDnCDRm

(2) TAUDnCMURm

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

Table 25.99 Contents of the TAUDnCMURm Register for TAUDTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the TAUDTTINm input signal width judgment function. Therefore, these registers should be set to 0.

Table 25.100 Simultaneous Rewrite Settings for TAUDTTINm Input Signal Width Judgment Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.12.4 Operating Procedure for TAUDTTINm Input Signal Width Judgment Function

Table 25.101 Operating Procedure for TAUDTTINm Input Signal Width Judgment Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.98, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Judgment Function**, and **Table 25.99, Contents of the TAUDnCMURm Register for TAUDTTINm Input Signal Width Judgment Function**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the TAUDTTINm start edge.
During Operation	The following register can be changed at any time: • TAUDnCDRm register	Upon detection of a TAUDTTINm start edge, TAUDnCNTm starts count down from the value of TAUDnCDRm. When TAUDnCMORm.TAUDnMD0 = 0 If TAUDnCNTm ≤ TAUDnCDRm when a TAUDTTINm input stop edge is detected, INTTAUDnIm is generated. When TAUDnCMORm.TAUDnMD0 = 1 If TAUDnCNTm > TAUDnCDRm when a TAUDTTINm input stop edge is detected, INTTAUDnIm is generated. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

25.12.13 Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

25.12.13.1 Overview

Summary

This function measures the width of an individual TAUDTTINm input signal. An interrupt is generated if the TAUDTTINm input width is longer than $FFFF_H$ + 1.

Prerequisites

- The operation mode must be set to One-Count Mode (see **Table 25.102, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)**).
- TAUDTTOUTm is not used for this function.
- The value of TAUDnCDRm must be set to $FFFF_H$.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected. $FFFF_H$ is loaded to TAUDnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUDTTINm input start edge is detected, TAUDnCNTm loads $FFFF_H$ and starts to count down.

If the counter reaches 0000_H before a stop edge is detected, an interrupt is generated.

Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TAUDnTIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, the TAUDTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] = 11_B, the TAUDTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

25.12.13.2 Block Diagram and General Timing Diagram

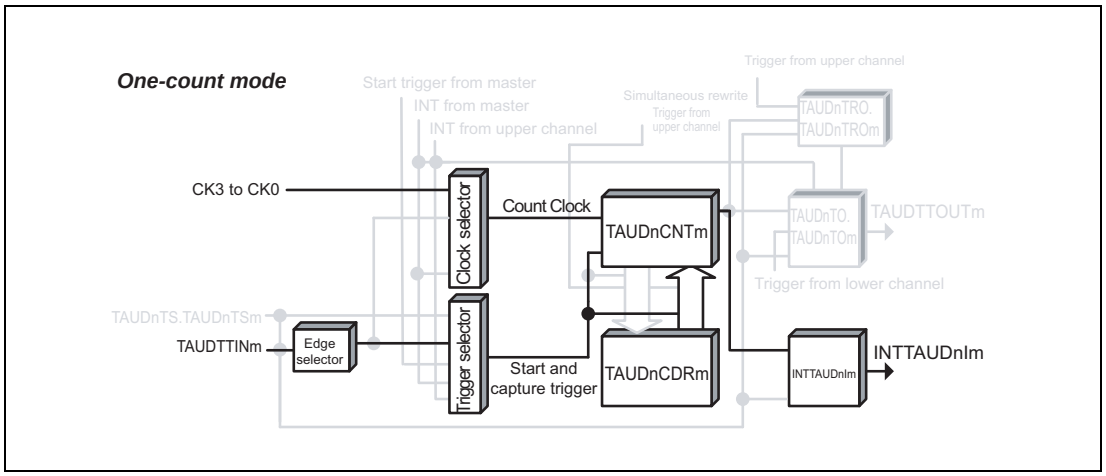


Figure 25.74 Block Diagram for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

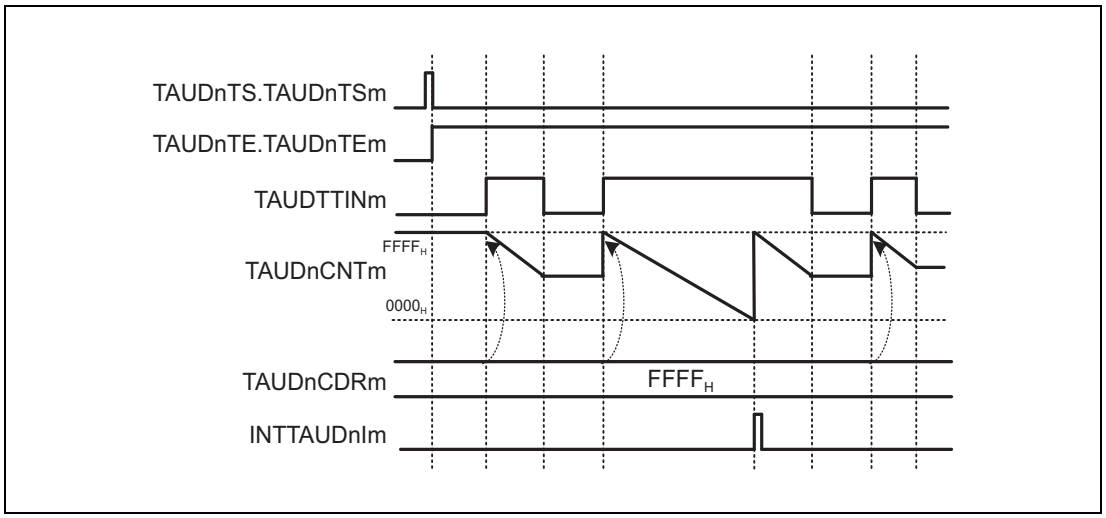


Figure 25.75 General Timing Diagram for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

25.12.13.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.102 Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	0: Disables the start trigger during operation

(2) TAUDnCMURm

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

Table 25.103 Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUDTTINm Width Measurement). Therefore, these registers must be set to 0.

Table 25.104 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.13.4 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Table 25.105 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.102, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)**, and **Table 25.103, Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)**. Set the value of TAUDnCDRm register to $FFFF_H$.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the start edge. When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value ($FFFF_H$).
During Operation	The TAUDnCNTm register can be read at any time.	TAUDnCNTm counts down. When the counter reaches 0000_H: • INTTAUDnIm is generated. When TAUDTTINm input stop edge is detected during count operation: • TAUDnCNTm stops and retains its current value. When TAUDTTINm input start edge is detected while the counter is stopped: • TAUDnCNTm loads the TAUDnCDRm value ($FFFF_H$) again, and continues to count down. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

25.12.14 Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

25.12.14.1 Overview

Summary

This function measures the cumulative width of a TAUDTTINm input signal. If the cumulative TAUDTTINm input width is longer than $FFFF_H$, an interrupt is generated and an overflow interrupt can be output.

Prerequisites

- The operation mode must be set to Gate Count Mode, (see **Table 25.106, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)**).
- TAUDTTOUTm is not used with this function.
- The value of TAUDnCDRm must be set to $FFFF_H$.

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected. $FFFF_H$ is loaded to TAUDnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUDTTINm input start edge and then continues to count down from the current value.

When the counter reaches 0000_H an interrupt is generated. $FFFF_H$ is loaded to TAUDnCNTm and the counter continues to count down until a TAUDTTINm input stop edge is detected.

Conditions

The valid start and stop edges are specified by the TAUDnCMURm.TIS[1:0] bits.

- If TAUDnCMURm.TAUDnTIS[1:0] = 10_B, the TAUDTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUDnCMURm.TAUDnTIS[1:0] = 11_B, the TAUDTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

25.12.14.2 Block Diagram and General Timing Diagram

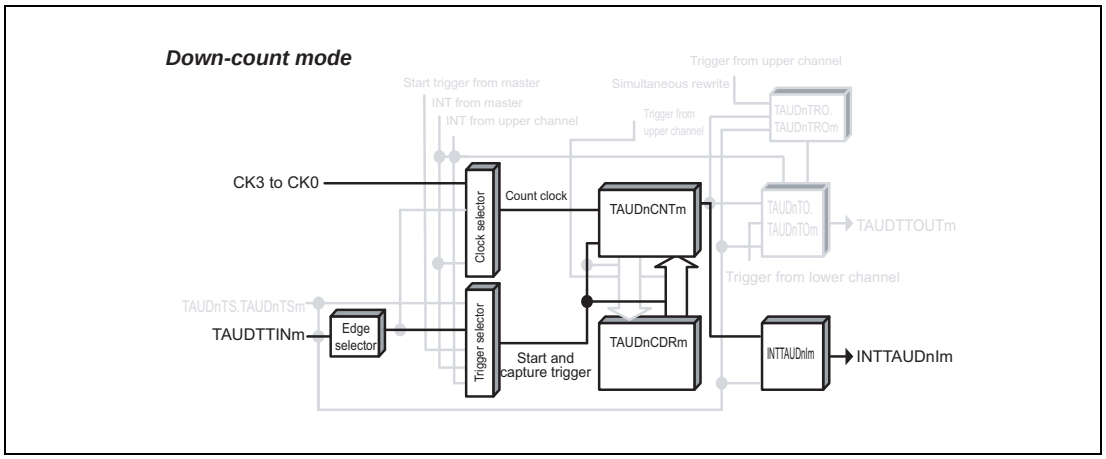


Figure 25.76 Block Diagram for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

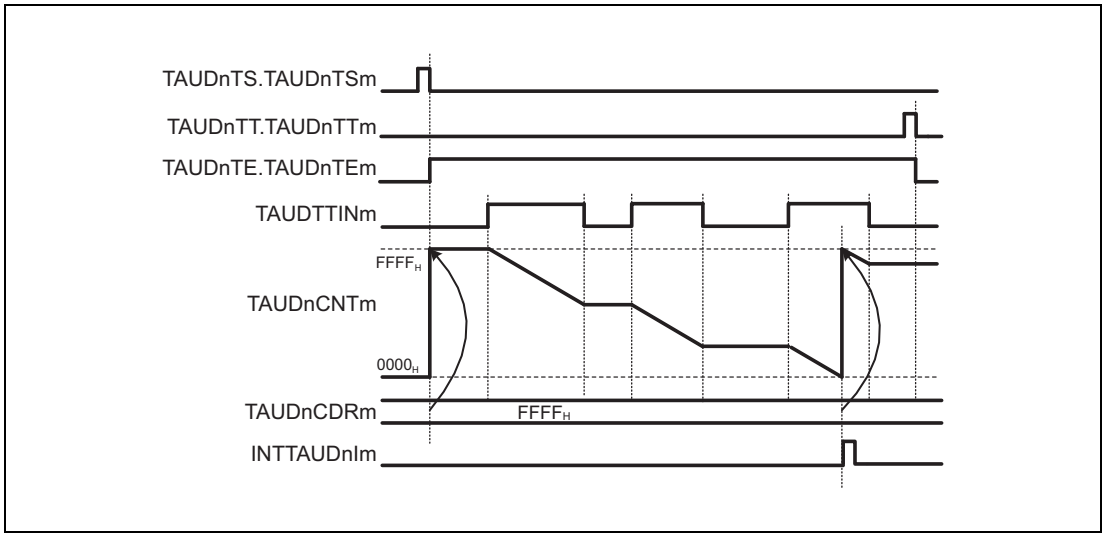


Figure 25.77 General Timing Diagram for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

25.12.14.3 Register Settings

(1) TAUDnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.106 Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	010: Valid edge of the TAUDTTINm input signal is used as an external start trigger and the reverse edge as a stop trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1100: Gate count mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation

(2) TAUDnCMURm

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

Table 25.107 Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 25.108 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.14.4 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Table 25.109 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 25.106, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)**, and **Table 25.107, Contents of the TAUDnCMURm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)**. Set the value of TAUDnCDRm register to $FFFF_H$.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 and TAUDnCNTm waits for detection of the start edge. When a start edge is detected, TAUDnCNTm loads the TAUDnCDRm value ($FFFF_H$).
During Operation	The TAUDnCNTm register can be read at all times.	TAUDnCNTm counts down. When the counter reaches 0000_H: • INTTAUDnIm is generated. • TAUDnCNTm loads the TAUDnCDRm value ($FFFF_H$) and continues to count down. When TAUDTTINm input stop edge is detected during count operation: • TAUDnCNTm stops and retains the current value. When TAUDTTINm input start edge is detected while the counter is stopped: • TAUDnCNTm counts down from the stop value. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

25.12.15 One-Phase PWM Output Function

25.12.15.1 Overview

Summary

This function adds dead time to a TAUDTTINm input signal. The resulting PWM signal is output via TAUDTTOUTm of the channel and TAUDTTOUTm of upper channels.

Prerequisites

- Each of two (or more) channels is enabled for dead time control (TAUDnTDE.TAUDnTDEm = 1).
- The operating mode for the lower channel should be set to one-count mode. (See **Table 25.111, Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function**.)
- Any operating mode can be set to upper channels.
- Channel output mode for upper and lower channels should be set to synchronous channel output mode 2 with one-phase PWM output. (See **Section 25.7, Channel Output Modes**.)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation.

The counter starts when a valid TAUDTTINm input start edge is detected. The value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value.

When the counter reaches 0000_H, an interrupt occurs. The counter is reset to $FFFF_H$ and waits for the next valid TAUDTTINm input start edge.

Table 25.110 TAUDTTOUTm to which Dead Time is Added and State of TAUDTTINm

TAUDnCMURm. TAUDnTIS[1:0]	TAUDnTOL. TAUDnTOLm	TAUDTTOUTm to which Dead Time is Added	TAUDnTDL. TAUDnTDLm	TAUDTTINm State when Added
10	0	TAUDTTOUTm low	0	High
			1	Low
	1	TAUDTTOUTm high	0	High
			1	Low
11	0	TAUDTTOUTm low	0	Low
			1	High
	1	TAUDTTOUTm high	0	Low
			1	High

Conditions

- TAUDnCMURm.TAUDnTIS[1:0] bits specify the type of width measurement:
  - TAUDnCMURm.TAUDnTIS[1:0] = 10_B: Uses both edges as valid edges for detection (Low width measurement).
  - TAUDnCMURm.TAUDnTIS[1:0] = 11_B: Uses both edges as valid edges for detection (High width measurement).
- The TAUDnTDL.TAUDnTDLm bit specifies the operation of TAUDTTOUTm for each channel when an interrupt or valid TAUDTTINm edge is detected on the lower channel:
  - If TAUDnTDL.TAUDnTDLm = 0, an interrupt is used as a TAUDTTOUTm set trigger and a valid TAUDTTINm edge as a TAUDTTOUTm reset trigger.
  - If TAUDnTDL.TAUDnTDLm = 1, a valid TAUDTTINm edge is used as a TAUDTTOUTm set trigger and an interrupt as a TAUDTTOUTm reset trigger.
- This function cannot make a forced restart.

25.12.15.2 Block Diagram and General Timing Diagram

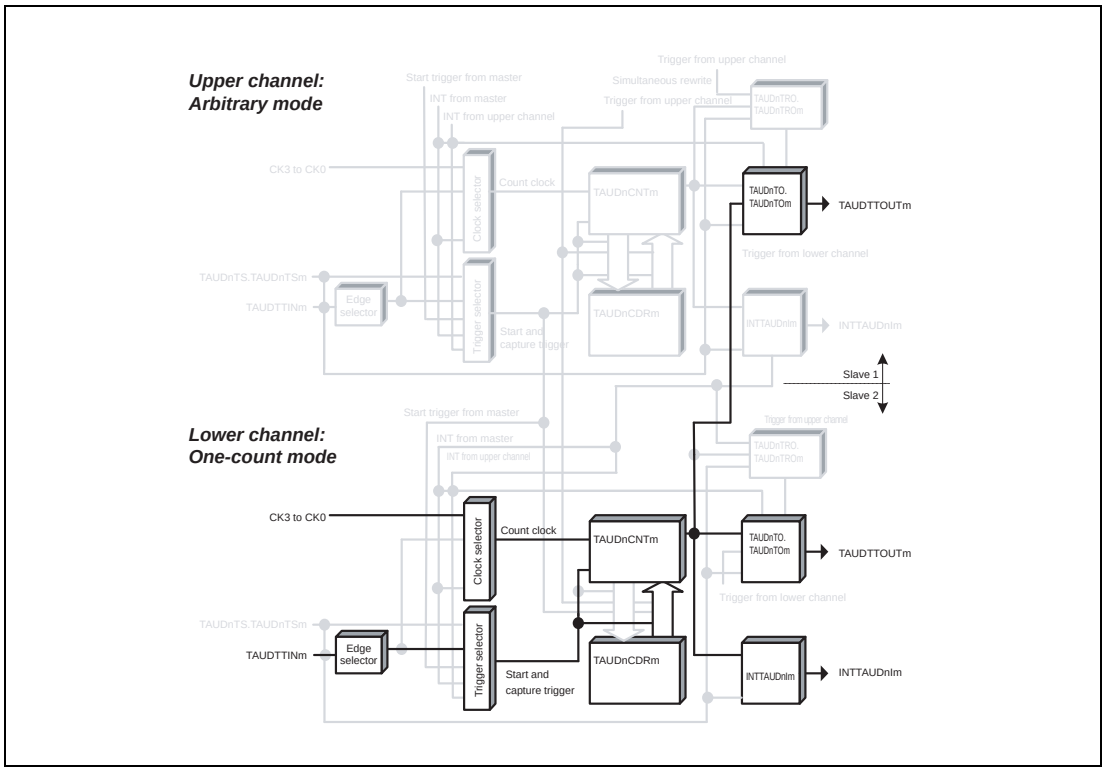


Figure 25.78 Block Diagram of One-Phase PWM Output Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

This setting considers a duty as an active high.

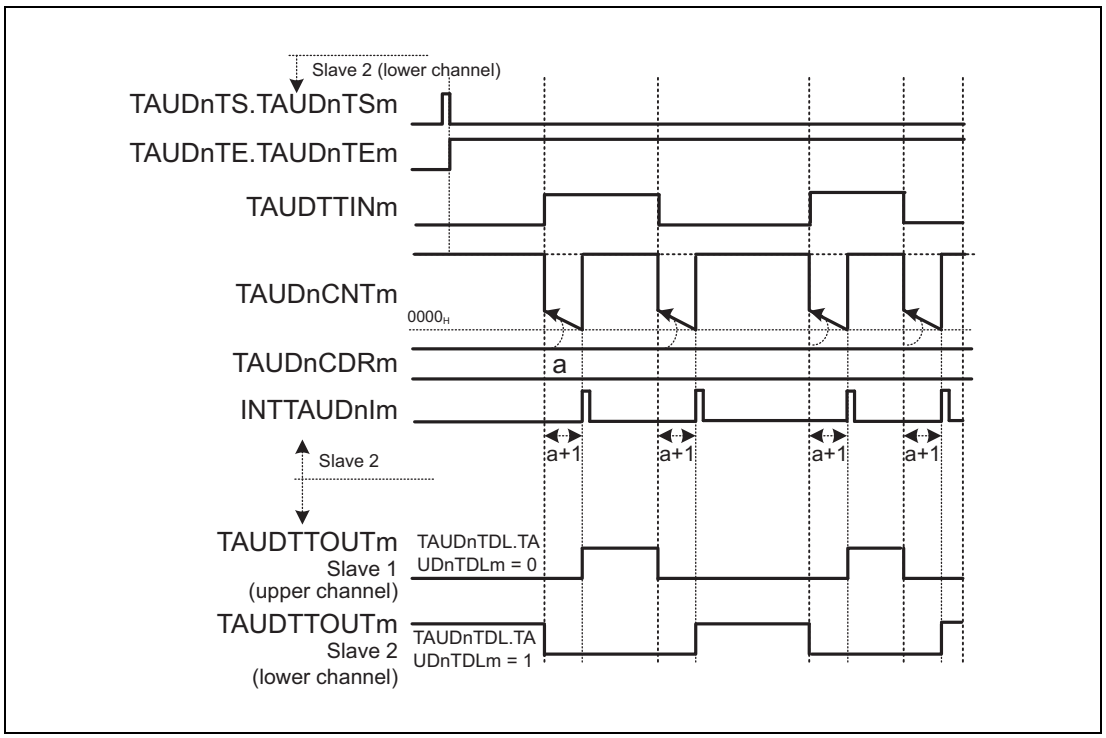


Figure 25.79 General Timing Diagram of One-Phase PWM Output Function

25.12.15.3 Register Settings for Lower Channels

(1) TAUDnCMORm for lower channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.111 Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for lower channels

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

Table 25.112 Contents of the TAUDnCMURm Register for the Lower Channel of the One-Phase PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	10: Detection of rising and falling edges (low width measurement) 11: Detection of rising and falling edges (high width measurement)

(3) Channel output mode for lower channels

Table 25.113 Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	1: Adds dead time upon detection of a TAUDTTINm input edge on a lower odd channel.
TAUDnTDL.TAUDnTDLm	0: Adds dead time of the positive-phase width 1: Adds dead time of the negative-phase width
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from upper channels.

(4) Simultaneous rewrite for lower channels

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 25.114 Simultaneous Rewrite Settings for One-Phase PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.12.15.4 Register Settings for Upper Channels

(1) TAUDnCMORm for upper channels

TAUDnCMORm register for upper channels can be set arbitrarily.

(2) TAUDnCMURm for upper channels

TAUDnCMURm register for upper channels can be set arbitrarily.

(3) Channel output mode for upper channels

Table 25.115 Control Bit Settings for Upper Channels in Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	1: Adds dead time upon detection of a TAUDTTINm input edge on a lower odd channel.
TAUDnTDL.TAUDnTDLm	0: Adds dead time of the positive-phase width 1: Adds dead time of the negative-phase width
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from lower channels.

(4) Simultaneous rewrite for upper channels

Simultaneous rewrite register for upper channels can be set arbitrarily.

25.12.15.5 Operating Procedure for One-phase PWM Output Function

Table 25.116 Operating Procedure for One-phase PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in **Table 25.111, Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function**, and **Table 25.112, Contents of the TAUDnCMURm Register for the Lower Channel of the One-Phase PWM Output Function**. Set TAUDnCMORm and TAUDnCMURm registers for the upper channel as described in **Section 25.12.15.4, Register Settings for Upper Channels**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 25.113, Control Bit Settings in Synchronous Channel Output Mode 2 with One-Phase PWM Output**.	Channel operation is stopped.
Start Operation	Set TAUDnTOE.TAUDnTOEm (slave channels 1 and 2) to 1 (at restart time only). Set TAUDnTS.TAUDnTSm = 1 for slave channel 2. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0. Detection of TAUDTTINm start edge	TAUDnTE.TAUDnTEm is set to 1 (slave channel 2) and TAUDnCNTm waits for detection of TAUDTTINm start edge. TAUDnCNTm loads TAUDnCDRm value.
During Operation	The TAUDnCDRm register value can be changed at any time. The TAUDnCNTm register can be read at any time.	TAUDnCNTm of slave channel 2 counts down. When the counter reaches 0000_H: • INTTAUDnIm is generated. • TAUDnCNTm stops counting. TAUDTTOUTm is changed by a TAUDTTINm edge detection signal and slave channel 2 INTTAUDnIm signal to output one-phase PWM waveform with dead time. Afterwards, this operation is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm = 1 for slave channel 2. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm and TAUDTTOUTm retain their current values.

Restart Operation

25.13 Independent Channel Real-Time Functions

This section describes functions that output the value of the TAUDnTRO.TAUDnTROm bit in real time.

25.13.1 Real-Time Output Function Type 1

25.13.1.1 Overview

Summary

This function outputs a value of the TAUDnTRO.TAUDnTROm bit from TAUDTTOUTm when a specified channel generates an interrupt (INTTAUDnIm). In this function, the interrupt is generated at certain specified intervals.

The upper channel is a channel which generates a real-time output trigger (TAUDnTRC.TAUDnTRCm = 1), and the lower channel is a channel which makes a real-time output in response to the upper channel trigger (TAUDnTRC.TAUDnTRCm = 0).

Prerequisites

- Channels should use the TAUDTTOUTm control of other channels.
- The operating mode for the upper channel should be set to interval timer mode. (See **Table 25.117, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1**.)
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See **Section 25.7, Channel Output Modes**.)
- Real-time output should be enabled for the upper channel (TAUDnTRE.TAUDnTREm = 1).

Functional description

The counter of the upper channel is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm to 1, enabling count operation. The current value of the data register of the upper channel (TAUDnCDRm) is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value.

When the counter of the upper channel reaches 0000_H, INTTAUDnIm is generated and TAUDTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) of every channel (only channels with TAUDnTRE.TAUDnTREm = 1). TAUDnCNTm then reloads the TAUDnCDRm value to continue operation subsequently.

The TAUDTTOUTm signal changes only when an interrupt is generated, and when its value is different from the current value of TAUDnTRO.TAUDnTROm at the moment that the interrupt is generated.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnTRC.TAUDnTRCm to 1 for the corresponding channel. The TAUDnTRC.TAUDnTRCm bit should be 0 for all other channels that do not generate a real-time output trigger.
- If real-time output of a lower channel is disabled (TAUDnTRE.TAUDnTREm = 0) or if the

channel itself is used as a rewrite trigger (TAUDnTRC.TAUDnTRCm = 1), the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in that channel.

- If real-time output of a lower channel is enabled (TAUDnTRE.TAUDnTREm = 1) and TAUDnTRC.TAUDnTRCm = 0, the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in the upper channel.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details, see **Section 25.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

25.13.1.2 Equations

INTTAUDnIm generation cycle = count clock cycle × (TAUDnCDRm value + 1)

25.13.1.3 Block Diagram and General Timing Diagram

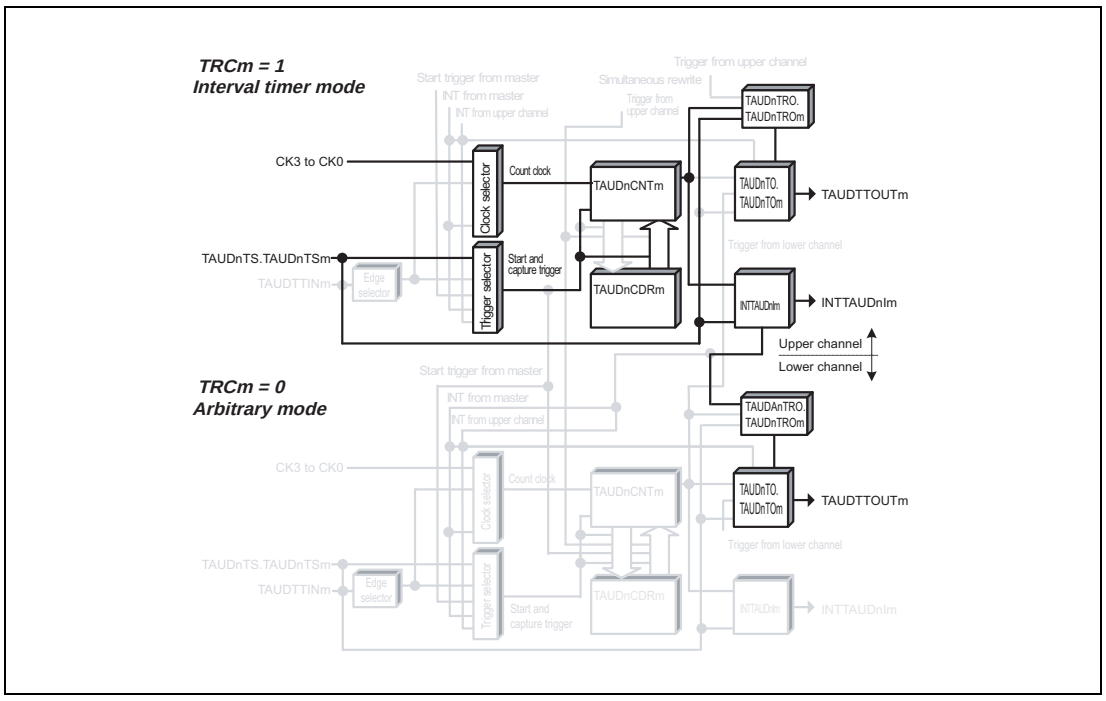


Figure 25.80 Block Diagram of Real-Time Output Function Type 1

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

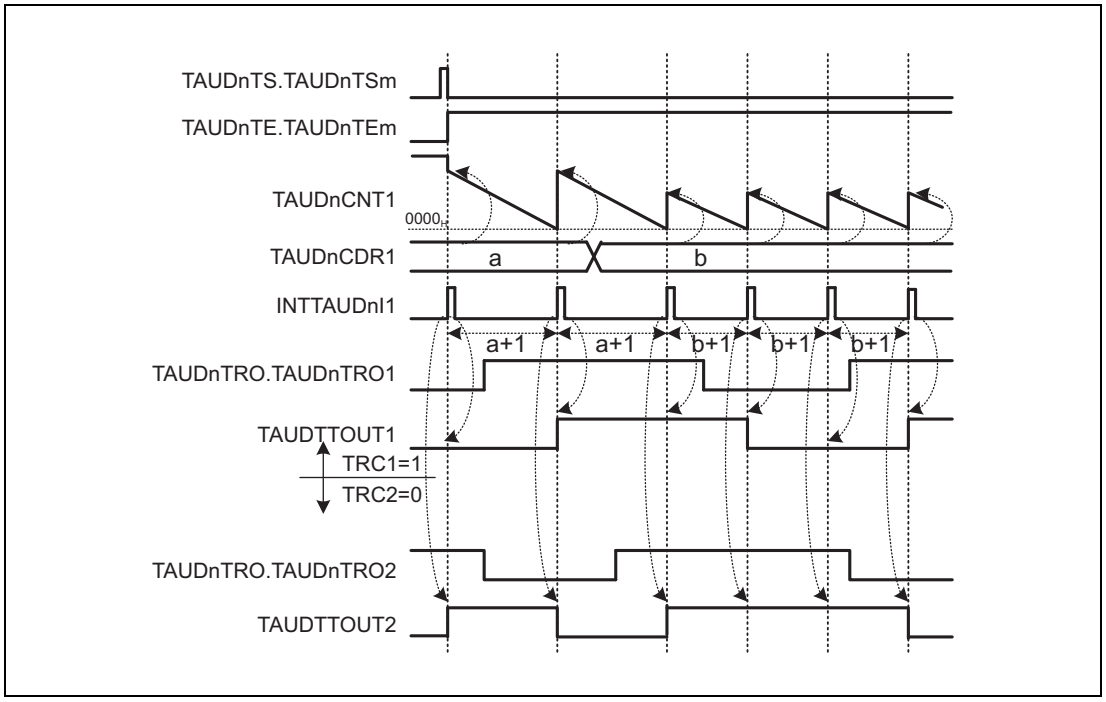


Figure 25.81 General Timing Diagram of Real-Time Output Function Type 1

25.13.1.4 Register Settings for Upper Channels

(1) TAUDnCMORm for upper channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.117 Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS [1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for upper channels

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

Table 25.118 Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for upper channels

Table 25.119 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	1: Channel m generates a unique real-time output trigger
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for upper channels

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the real-time output function type 1. Therefore, these registers should be set to 0.

Table 25.120 Simultaneous Rewrite Settings for Real-Time Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.13.1.5 Register Settings for Lower Channels

(1) TAUDnCMORm for lower channels

The TAUDnCMORm register for lower channels can be set arbitrarily.

(2) TAUDnCMURm for lower channels

The TAUDnCMURm register for lower channels can be set arbitrarily.

(3) Channel output mode for lower channels

Table 25.121 Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTDL.TAUDnTDLm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTRE.TAUDnTREm	1: Enables real-time output
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for lower channels

Simultaneous rewrite registers for lower channels can be set arbitrarily.

25.13.1.6 Operating Procedure for Real-Time Output Function Type 1

Table 25.122 Operating Procedure for Real-Time Output Function Type 1

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for upper channels as described in **Table 25.117, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1**, and **Table 25.118, Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 1**. Set TAUDnCMORm and TAUDnCMURm registers for lower channels as described in **Section 25.13.1.5, Register Settings for Lower Channels**. Set the value of TAUDnCDRm register (only channels with TAUDnTRC.TAUDnTRCm = 1) Set channel output mode by setting the control bits as described in **Table 25.119, Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output**. Set channel output mode by setting the control bits as described in **Table 25.121, Control Bit Settings for the Lower Channels in Independent Channel Output Mode 1 with Real-Time Output**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm = 1 on the channel with TAUDnTRC.TAUDnTRCm set to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	[Channel with TAUDnTRC.TAUDnTRCm set to 1] TAUDnTE.TAUDnTEm is set to 1 and the counter starts. The TAUDnCDRm value is loaded in TAUDnCNTm. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm is generated.
During Operation	TAUDnCDRm and TAUDnTRO.TAUDnTROm can be changed at any time. The TAUDnCNTm register can be read at any time.	TAUDnCNTm counts down. When the counter reaches 0000_H: • The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. • INTTAUDnIm is generated. • TAUDTTOUTm outputs the current value of the real-time output bit TAUDnTRO.TAUDnTROm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. Both TAUDnCNTm and TAUDTTOUTm retain their current values.

Restart Operation

25.13.1.7 Specific Timing Diagrams

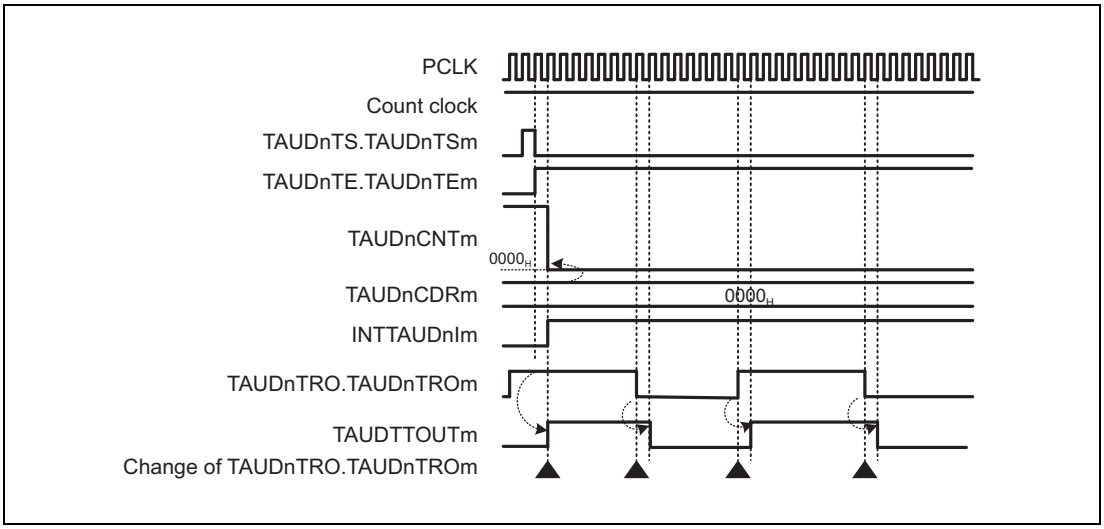


Figure 25.82 TAUDnCDRm = 0000_H, TAUDnCMORm.TAUDnMD0 = 1

- The value of TAUDTTOUTm changes according to the setting of TAUDnTRO.TAUDnTROm with a delay of one PCLK cycle.

25.13.2 Real-Time Output Function Type 2

25.13.2.1 Overview

Summary

This function outputs the value of TAUDnTRO.TAUDnTROm bit from TAUDTTOUTm when a specified channel generates an interrupt (INTTAUDnIm). In this function, the interrupt is generated when a valid TAUDTTINm input edge is detected or the function starts.

The upper channel is a channel which generates a real-time output trigger (TAUDnTRC.TAUDnTRCm = 1), and the lower channel is a channel which makes a real-time output in response to the upper channel trigger (TAUDnTRC.TAUDnTRCm = 0).

Prerequisites

- Channels should use the TAUDTTOUTm control of the other channels.
- The operating mode for the upper channel should be set to capture mode. (See **Table 25.123, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2**.)
- Any operating mode can be set for lower channels.
- The channel output mode for all the channels should be set to independent channel output mode 1 with real-time output. (See **Section 25.7, Channel Output Modes**.)
- Real-time output should be enabled for the upper channel (TAUDnTRE.TAUDnTREm = 1).

Functional description

The counter for upper channels is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm to 1, enabling count operation. The counter starts to count up.

When a valid TAUDTTINm input edge is generated on one of upper channels, an interrupt occurs and TAUDTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) of every channel (only channels with TAUDnTRE.TAUDnTREm = 1).

The TAUDTTOUTm signal changes only when an interrupt is generated, and when TAUDTTOUTm value is different from the current value of TAUDnTRO.TAUDnTROm during the occurrence of the interrupt.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnTRC.TAUDnTRCm to 1 for the corresponding channel. The TAUDnTRC.TAUDnTRCm bit should be 0 for all other channels that do not generate a real-time output trigger.
- If real-time output of a lower channel is disabled (TAUDnTRE.TAUDnTREm = 0) or if the channel itself is used as a rewrite trigger (TAUDnTRC.TAUDnTRCm = 1), the value of that channel’s TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in that channel.
- If real-time output of a lower channel is enabled (TAUDnTRE.TAUDnTREm = 1) and TAUDnTRC.TAUDnTRCm = 0, the value of that channel's TAUDnTRO.TAUDnTROm bit is output when INTTAUDnIm is generated in the upper channel.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not

output. For details, see **Section 25.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

25.13.2.2 Block Diagram and General Timing Diagram

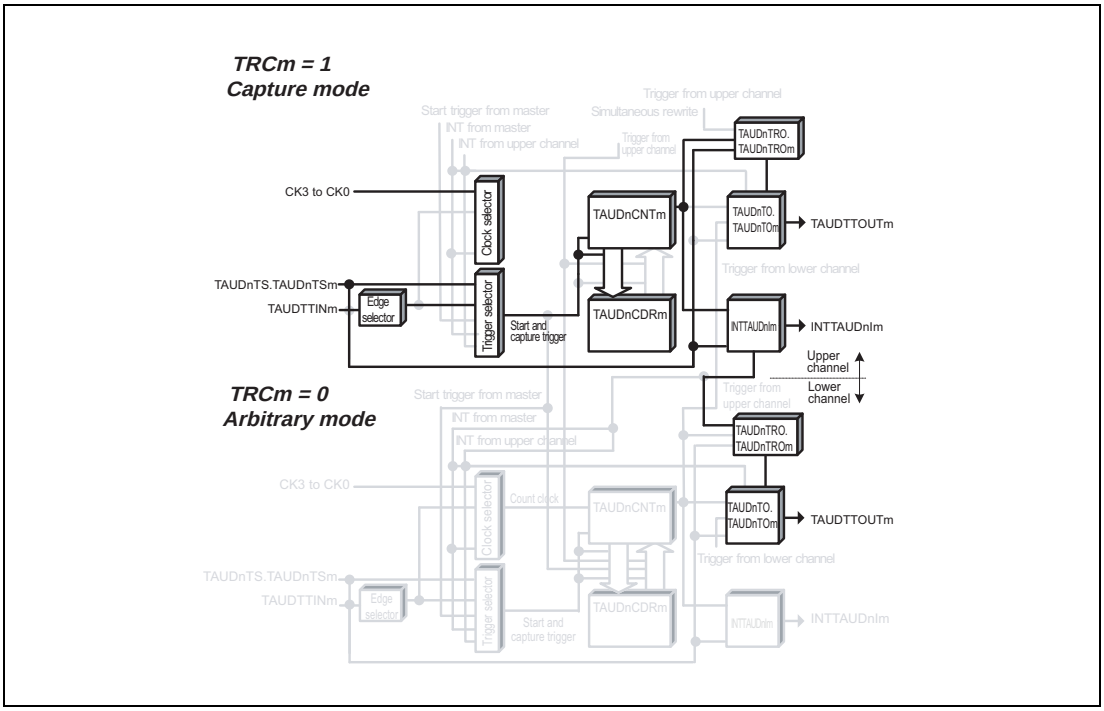


Figure 25.83 Block Diagram of Real-Time Output Function Type 2

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)

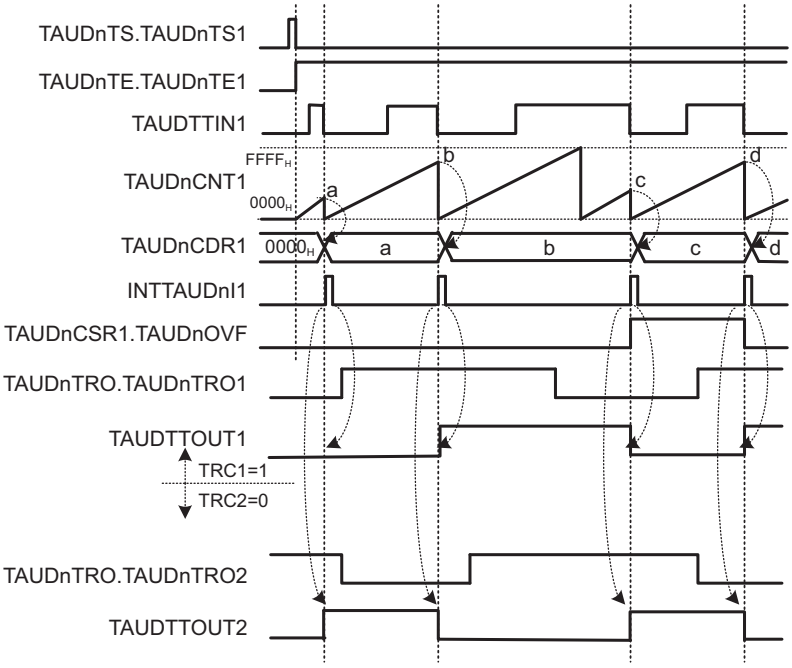


Figure 25.84 General Timing Diagram of Real-Time Output Function Type 2

25.13.2.3 Register Settings for Upper Channels

(1) TAUDnCMORm for upper channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.123 Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as an external start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for upper channels

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

Table 25.124 Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode for upper channels

Table 25.125 Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	1: Channel m generates a unique real-time output trigger
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for upper channels

The simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with the real-time output function type 2. Therefore, these registers should be set to 0.

Table 25.126 Simultaneous Rewrite Settings for Real-Time Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.13.2.4 Register Settings for Lower Channels

(1) TAUDnCMORm for lower channels

The TAUDnCMORm register for lower channels can be set arbitrarily.

(2) TAUDnCMURm for lower channels

The TAUDnCMURm register for lower channels can be set arbitrarily.

(3) Channel output mode for lower channels

Table 25.127 Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode. (The value after reset.)
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low 1: Real-time output is high
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for lower channels

Simultaneous rewrite registers for lower channels can be set arbitrarily.

25.13.2.5 Operating Procedure for Real-Time Output Function Type 2

Table 25.128 Operating Procedure for Real-Time Output Function Type 2

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for upper channels as described in **Table 25.123, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2**, and **Table 25.124, Contents of the TAUDnCMURm Register for the Upper Channel of Real-Time Output Function Type 2**. Set TAUDnCMORm and TAUDnCMURm registers for the lower channel as described in **Section 25.13.2.4, Register Settings for Lower Channels**. The TAUDnCDRm register functions as a capture register (only channels with TAUDnTRC.TAUnTRCm = 1). Set channel output mode by setting the control bits as described in **Table 25.125, Control Bit Settings in Independent Channel Output Mode 1 with Real-Time Output**. Set channel output mode by setting the control bits as described in **Table 25.127, Control Bit Settings for Lower Channels in Independent Channel Output Mode 1 with Real-Time Output**.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm = 1 on the channel with TAUDnTRC.TAUDnTRCm set to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	[Channel with TAUDnTRC.TAUDnTRCm set to 1] TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000_H. If TAUDnCMORm.TAUDnMD0 is 1, INTTAUDnIm is generated.
During Operation	TAUDnTRO.TAUDnTROm can be changed at any time.	TAUDnCNTm starts to count up from 0000_H. When a valid TAUDTTINm input edge is detected: • TAUDnCNTm captures the TAUDnCDRm value, and the counter is cleared to 0000_H. • INTTAUDnIm is generated. • When the TAUDTTINm input valid edge is detected immediately after the generation of an overflow, the TAUDnCSRm.TAUDnOVF bit is set to 1. When detected before the generation of an overflow, the TAUDnCSRm.TAUDnOVF bit is cleared to 0. TAUDTTOUTm outputs the current value of real-time output bit TAUDnTRO.TAUDnTROm. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops. TAUDnCNTm, TAUDnCSRm.TAUDnOVF, and TAUDTTOUTm retain their current values.

Restart Operation

25.13.2.6 Specific Timing Diagrams

(1) Operation start and stop

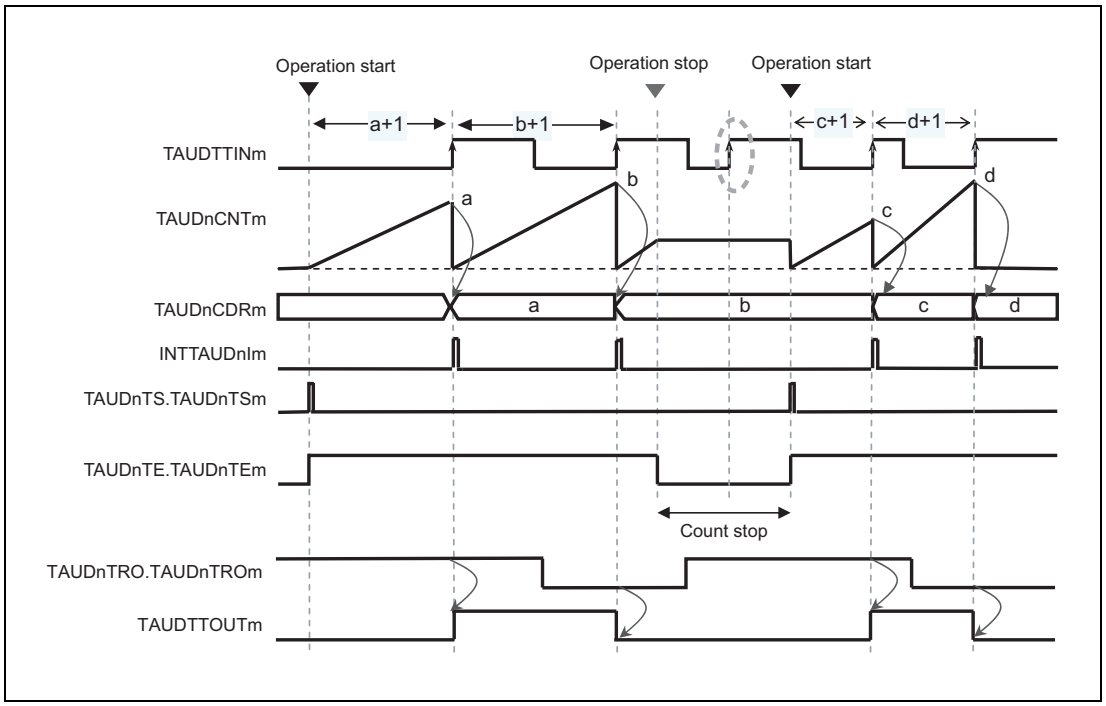


Figure 25.85 Operation Start and Stop (TAUDnCMORm.TAUDnMD0 = 0)

- When TAUDnTS.TAUDnTSm is set to 1, the counter starts counting up.
- When a valid input edge is detected, the current value of the counter is written to the data register (TAUDnCDRm) and an interrupt is generated.
- TAUDTTOUTm outputs the current value of the real-time output bit (TAUDnTRO.TAUDnTROm) and the counter resets and starts to count up again.
- The TAUDTTOUTm signal only changes when an interrupt is generated, and then only when its value is different from the current value of TAUDnTRO.TAUDnTROm at the moment that the interrupt is generated.
- If the counter is stopped (TAUDnTE.TAUDnTEm = 0), valid input edges are ignored and no interrupt is generated.

25.14 Independent Channel Simultaneous Rewrite Functions

This section describes functions that carry out simultaneous rewrite.

25.14.1 Simultaneous Rewrite Trigger Generation Function Type 1

25.14.1.1 Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is generated at regular intervals.

The upper channel is a channel which generates a simultaneous rewrite trigger (TAUDnRDC.TAUDnRDCm = 1), and the lower channel is a channel which makes a simultaneous rewrite in response to the upper channel trigger (TAUDnRDC.TAUDnRDCm = 0).

Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous rewrite (TAUDnRDE.TAUDnRDEm = 1).
- The operating mode for the upper channel should be set to interval timer mode. (See **Table 25.129, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1**.)
- For the operating mode that can be set for lower channels, see **Table 25.46, Channel Functions and the Methods They Use for Simultaneous Rewrite**.
- TAUDTTOUTm is not used for any channel in this function.

Functional description

The counter operation is enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) for upper and lower channels to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of the data register buffer for upper channels (TAUDnCDRm buf) is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value. The counter for lower channels start to count according to the selected operating mode.

Once the counter reaches 0000_H, an interrupt occurs on the channel. The current value of the corresponding TAUDnCDRm buffer is loaded into TAUDnCNTm to continue operation subsequently.

If the channel where an interrupt occurs is specified as a trigger channel for simultaneous rewrite (TAUDnRDC.TAUDnRDCm = 1) and is an upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUDnRSF.TAUDnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers. Each time a counter starts to count down, it reads the value in the data register buffer and counts down from this value.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Conditions

- The channel which is monitored for INTTAUDnIm occurrence is specified by setting TAUDnRDC.TAUDnRDCm = 1 for the corresponding channel. The TAUDnRDC.TAUDnRDCm

bit should be set to 0 for all other channels in which simultaneous rewrite should take place.

- If the TAUDnCMORm.TAUDnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, see **Section 25.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

25.14.1.2 Equations

Simultaneous rewrite trigger generation cycle = count clock cycle × (TAUDnCDRm + 1)

To control simultaneous rewrite, the following condition should be satisfied:

[PWM]

TAUDnCDRm = [(value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1) × number of interrupts] – 1

[Triangle PWM]

TAUDnCDRm = [(value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1) × 2 × number of interrupts] – 1

That is, the ratio of TAUDnCDRm + 1 and value of TAUDnCDRm of master channel subject to simultaneous rewrite + 1 should be an integer. This integer corresponds to the number of interrupts.

For triangle PWM, remember that the cycle doubles.

25.14.1.3 Block Diagram and General Timing Diagram

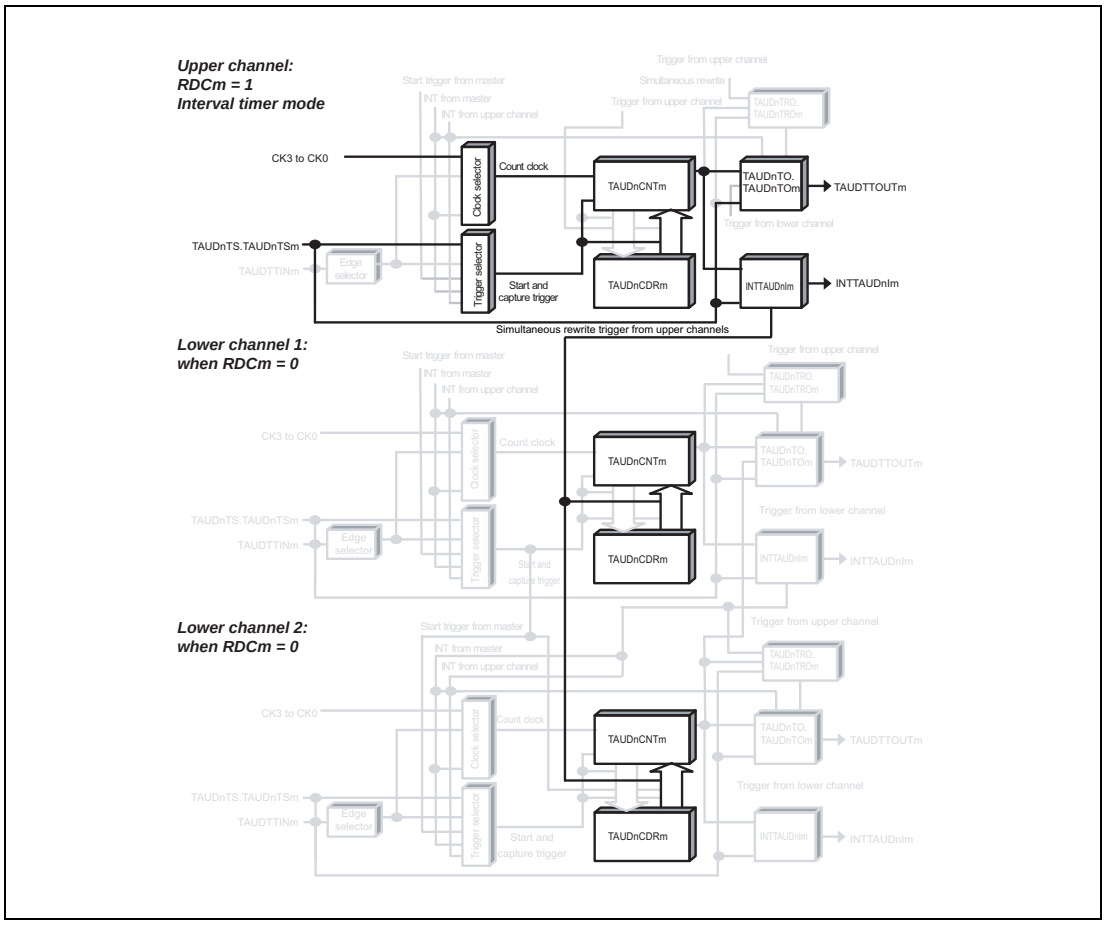


Figure 25.86 Block Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

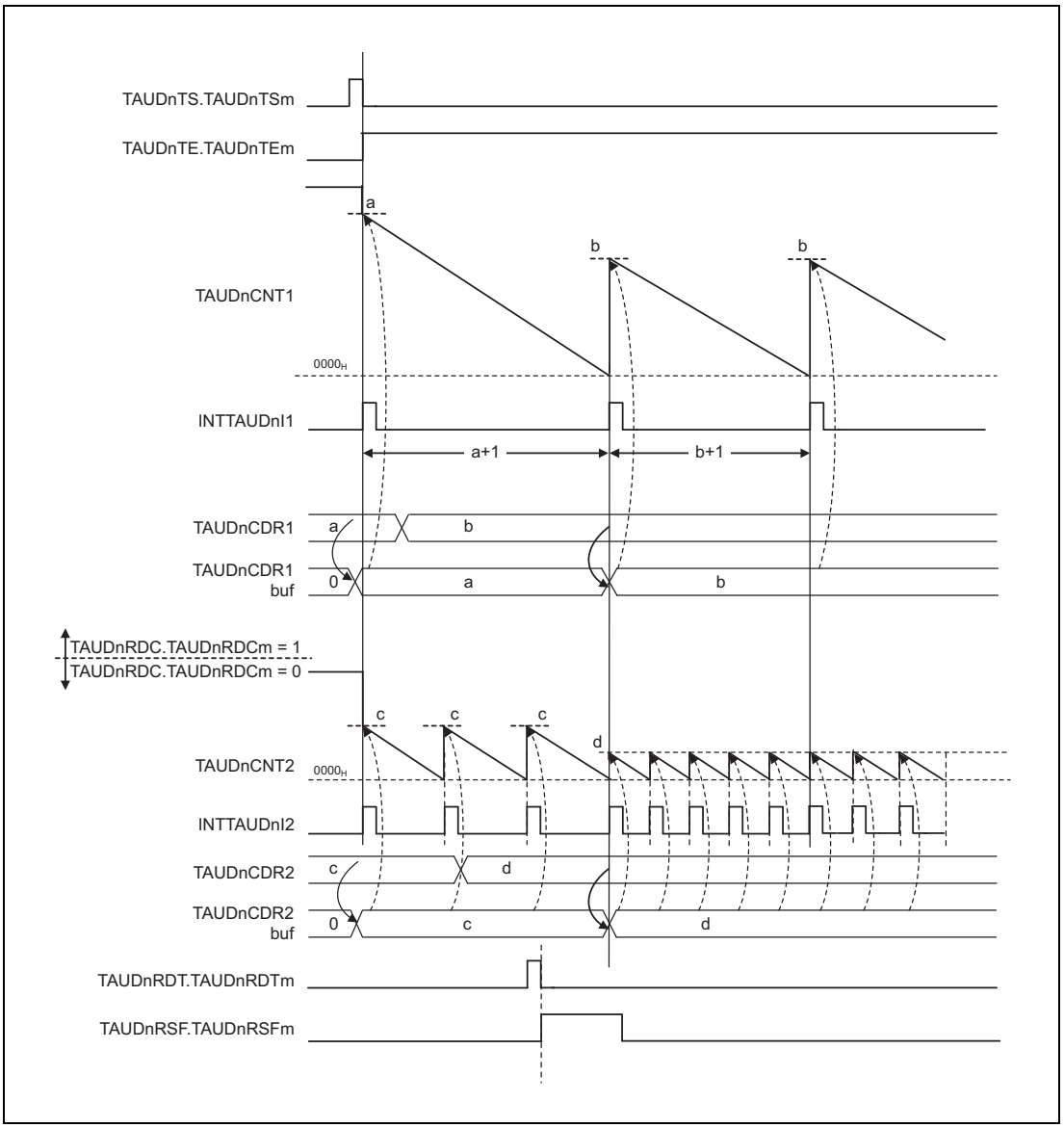


Figure 25.87 General Timing Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

25.14.1.4 Register Settings for Upper Channels

(1) TAUDnCMORm for upper channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.129 Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation. Set to 0.
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for upper channels

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

Table 25.130 Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for upper channels

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function. However, this mode can be used in independent channel output mode controlled by software.

(4) Simultaneous rewrite for upper channels

Table 25.131 Simultaneous Rewrite Settings for Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts counting.
TAUDnRDC.TAUDnRDCm	1: Monitors INTTAUDnIm signal which triggers a simultaneous rewrite on the channel.

25.14.1.5 Register Settings for Lower Channels

(1) TAUDnCMORm for lower channels

TAUDnCMORm register for lower channels must follow the TAUDnCMORm register settings in the operating mode which can be set. (See **Table 25.46, Channel Functions and the Methods They Use for Simultaneous Rewrite**.)

(2) TAUDnCMURm for lower channels

TAUDnCMURm register for lower channels must follow the TAUDnCMURm register settings in the operating mode which can be set. (See **Table 25.46, Channel Functions and the Methods They Use for Simultaneous Rewrite**.)

(3) Channel output mode for lower channels

Output can be made according to the setting for lower channels (master/slave). For the available function for simultaneous rewrite trigger generation function type 1, see **Table 25.45, Simultaneous Rewrite Methods and when They are Triggered**.

(4) Simultaneous rewrite for lower channels

Table 25.132 Simultaneous Rewrite Settings for Lower Channels in Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts counting.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.14.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Table 25.133 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers for the upper channel as described in **Table 25.129, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1**, and **Table 25.130, Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1**. Set TAUDnCMORm and TAUDnCMURm registers for lower channels as described in **Section 25.14.1.5, Register Settings for Lower Channels**. Set the value of TAUDnCDRm register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCDRm value is loaded into TAUDnCNTm. If TAUDnCMORm.TAUDnMD0 = 1, INTTAUDnIm is generated.
During Operation	TAUDnRDT.TAUDnRDTm and TAUDnCDR.TAUDnCDRm are changeable. TAUDnRSF.TAUDnRSFm can be always read.	TAUDnCNTm counts down. When the counter reaches 0000_H: • The TAUDnCDRm value is loaded in TAUDnCNTm again and count operation continues. • INTTAUDnIm is generated. If INTAUDnIm is generated on the channel where TAUDnRDC.TAUDnRDCm is set to 1, simultaneous rewrite is controlled. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

Restart Operation

25.14.2 Simultaneous Rewrite Trigger Generation Function Type 2

25.14.2.1 Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. In this function, the interrupt is generated when a valid TAUDTTINm input edge is detected or the function starts.

The upper channel is a channel which generates a simultaneous rewrite trigger (TAUDnRDC.TAUDnRDCm = 1), and the lower channel is a channel which makes a simultaneous rewrite in response to the upper channel trigger (TAUDnRDC.TAUDnRDCm = 0).

Prerequisites

- Two or more channels lower than the channel used as upper channel are enabled for simultaneous rewrite (TAUDnRDE.TAUDnRDEm = 1).
- The operation mode of the upper channel must be set to Capture Mode (see **Table 25.134, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2**).
- For the operation mode that can be set for a lower channel, see **Table 25.46, Channel Functions and the Methods They Use for Simultaneous Rewrite**.

Functional description

The counter operation is enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) for upper and lower channels to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The counter for the upper channel starts to count up, and then the counter for lower channels start to count according to the selected operating mode.

When a TAUDTTINm input edge occurs on the upper channel, an interrupt is generated. The trigger is detected by the lower channel(s), which then also generate an interrupt.

When TAUDnRDC.TAUDnRDCm = 1 on the upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUDnRSF.TAUDnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Conditions

- The channel which is monitored for INTTAUDnIm is specified by setting TAUDnRDC.TAUDnRDCm = 1 for the corresponding channel. The TAUDnRDC.TAUDnRDCm bit must be 0 for all other channels in which simultaneous rewrite should take place.
- If the TAUDnCMORm.TAUDnMD0 bit is set to 1, an interrupt is generated when the function starts. For details see **Section 25.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

25.14.2.2 Block Diagram and General Timing Diagram

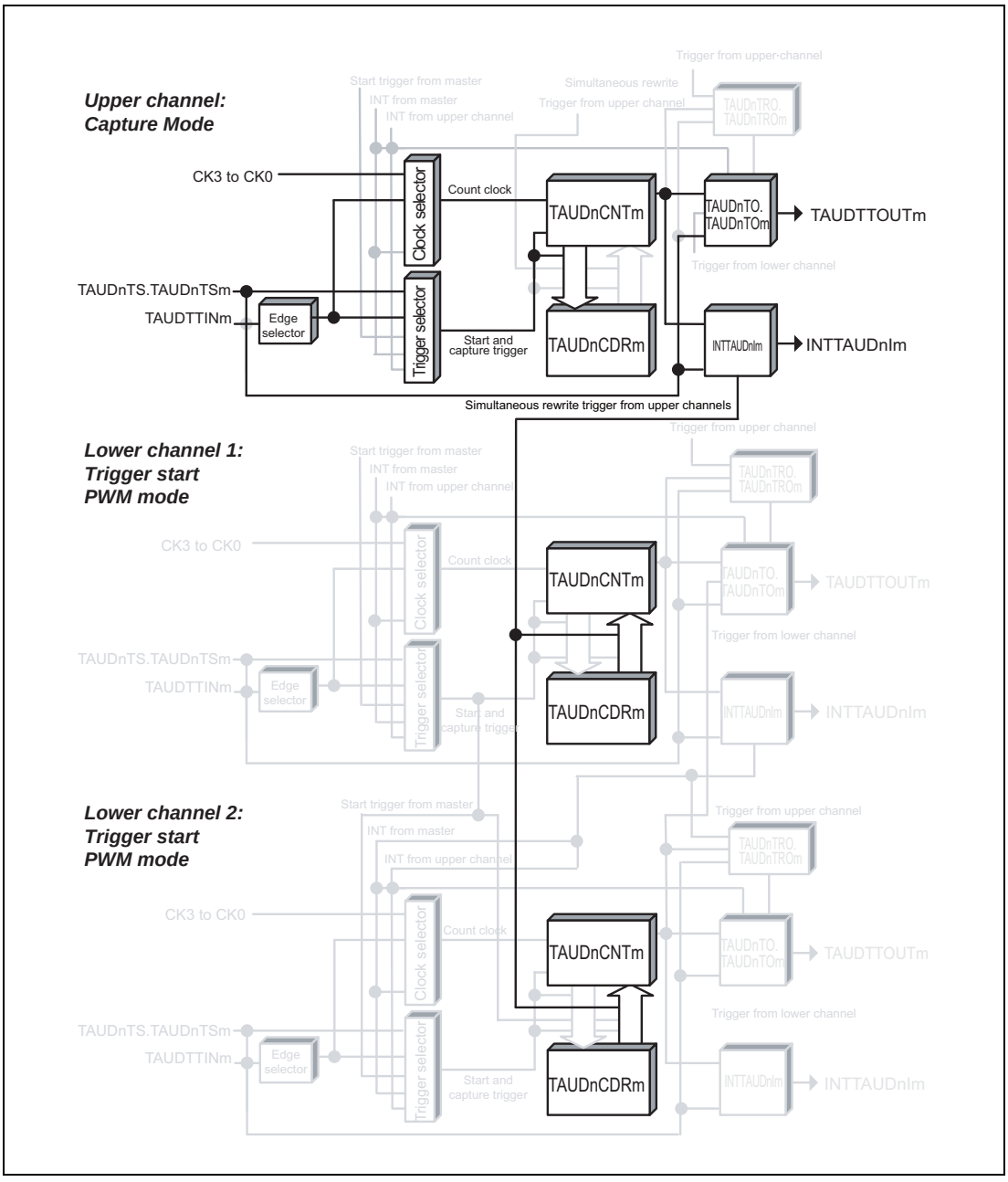


Figure 25.88 Block Diagram for Simultaneous Rewrite Trigger Generation Function Type 2

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)
- Upper channel (CH1) generates simultaneous rewrite trigger.

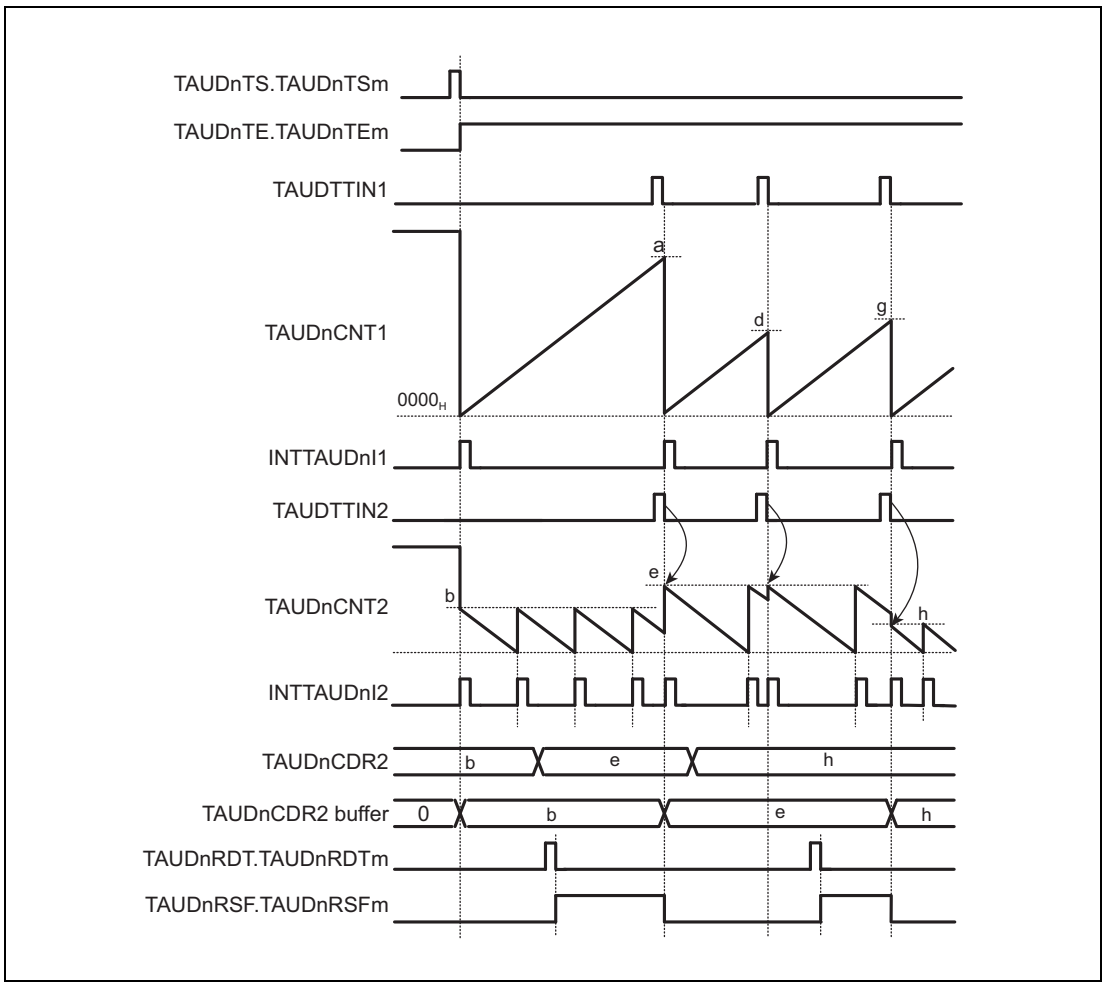


Figure 25.89 General Timing Diagram for Simultaneous Rewrite Trigger Generation Function Type 2

25.14.2.3 Register Settings for Upper Channels

(1) TAUDnCMORm for upper channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.134 Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Independent operation, Set to 0.
10 to 8	TAUDnSTS[2:0]	001: Valid edge of the TAUDTTINm input signal is used as the external capture trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for upper channels

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

Table 25.135 Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode for upper channels

The channel output mode is not used by this function.

(4) Simultaneous rewrite for upper channels

Table 25.136 Simultaneous Rewrite Settings for Simultaneous Rewrite Trigger Generation Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	1: Monitors INTTAUDnIm signal which triggers a simultaneous rewrite on the channel.

25.14.2.4 Register Settings for Lower Channels

(1) TAUDnCMORm for lower channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.137 Contents of the TAUDnCMORm Register for the Lower channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for lower channels

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

Table 25.138 Contents of the TAUDnCMURm Register for the Lower Channel of Simultaneous Rewrite Trigger Generation Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode for lower channels

Output can be made according to the trigger start PWM mode setting.

(4) Simultaneous rewrite for lower channels

Table 25.139 Simultaneous Rewrite Settings for the Lower Channel in Simultaneous Rewrite Trigger Generation Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	1: Selects one of upper channels as simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Loads a simultaneous rewrite control signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.14.2.5 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 2

Table 25.140 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 2

	Operation	TAUDn Status
Initial Channel Setting	Set the TAUDnCMORm register and TAUDnCMURm registers for the upper channel as described in **Table 25.134, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2** and **Table 25.135, Contents of the TAUDnCMURm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2**. Set the TAUDnCMORm register and TAUDnCMURm registers for the lower channel as described in **Table 25.137, Contents of the TAUDnCMORm Register for the Lower channel of Simultaneous Rewrite Trigger Generation Function Type 2** and **Table 25.138, Contents of the TAUDnCMURm Register for the Lower Channel of Simultaneous Rewrite Trigger Generation Function Type 2**. The TAUDnCDRm register functions as a capture register.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm to 1. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is set to 1 and the counter starts. TAUDnCNTm is cleared to 0000_H. INTTAUDnIm is generated when TAUDnCMORm.TAUDnMD0 is set to 1.
During Operation	TAUDnRDT.TAUDnRDTm can be set at any time. TAUDnRSF.TAUDnRSFm can be read at any time.	TAUDnCNTm counts up from 0000_H. When a TAUDTTINm valid edge is detected: • TAUDnCNTm transfers (captures) its value to TAUDnCDRm and returns to 0000_H. • INTTAUDnIm is generated. Simultaneous rewrite is controlled when INTTAUDnIm is generated from the channel where TAUDnRDC.TAUDnRDCm is set to 1. Afterwards, this procedure is repeated.
Stop Operation	Set TAUDnTT.TAUDnTTm to 1. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and it retains its current value.

Restart Operation

25.15 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the timer array unit D. For a general overview of synchronous channel operation, see **Section 25.2, Overview**

This section describes functions that generate PWM signals at regular intervals.

25.15.1 PWM Output Function

25.15.1.1 Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty cycle of the TAUDTTOUTm to be set. The pulse cycle is set in the master channel. The duty cycle is set in the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to interval timer mode. (See **Table 25.141, Contents of the TAUDnCMORm Register for the Master Channel of the PWM Output Function**.)
- The operating mode for the slave channels should be set to one-count mode. (See **Table 25.144, Contents of the TAUDnCMORm Register for the Slave Channel of the PWM Output Function**.)
- TAUDTTOUTm is not used with the master channel of this function.
- The channel output mode for the slave channels should be set to Synchronous Channel Output Mode 1. (See **Section 25.7, Channel Output Modes**.)

Functional description

The counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm, and the counter starts counting down from the TAUDnCDRm value. If an INTTAUDnIm is generated on the master channel and TAUDTTOUTm (slave) is set/reset, PWM output is made.

- Master channel:
  When the master channel counter reaches 0000_H and the pulse cycle time has passed, INTTAUDnIm is generated. The counter loads TAUDnCDRm value into TAUDnCNTm and counts down.
- Slave channel:
  When INTTAUDnIm is generated on the master channel, the counter operation of the slave channel is triggered. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave) and the counter starts counting down from the TAUDnCDRm value. TAUDTTOUTm signal is set to the active level.
  When the counter reaches to 0000_H (duty time has elapsed), INTTAUDnIm is generated and a TAUDTTOUTm signal is set to an inactive level. The counter is reset to $FFFF_H$ and waits for the next INTTAUDnIm (start of the next pulse cycle) of the master channel.

Counter operation can be stopped by setting the TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. See **Section 25.6, Simultaneous Rewrite**.

25.15.1.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUDnCDRm (slave)/(TAUDnCDRm (master) + 1)) × 100

– Duty cycle = 0%
  TAUDnCDRm (slave) = 0000_H
– Duty cycle = 100%
  TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

25.15.1.3 Block Diagram and General Timing Diagram

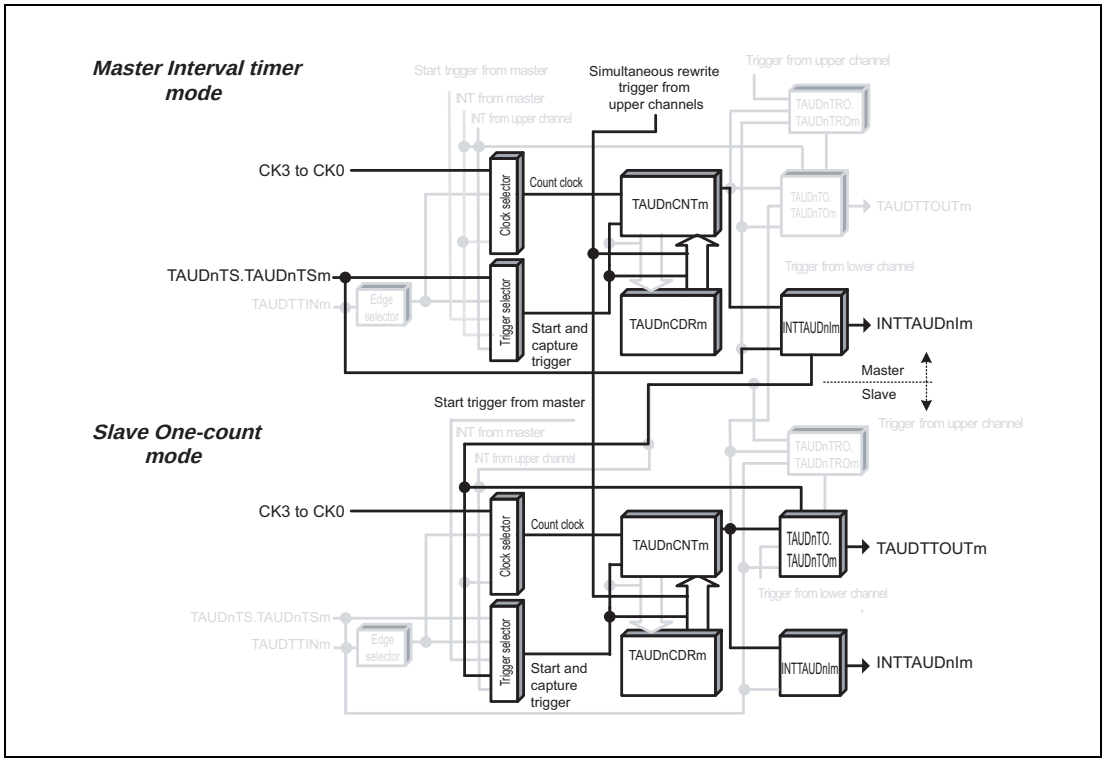


Figure 25.90 Block Diagram of PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

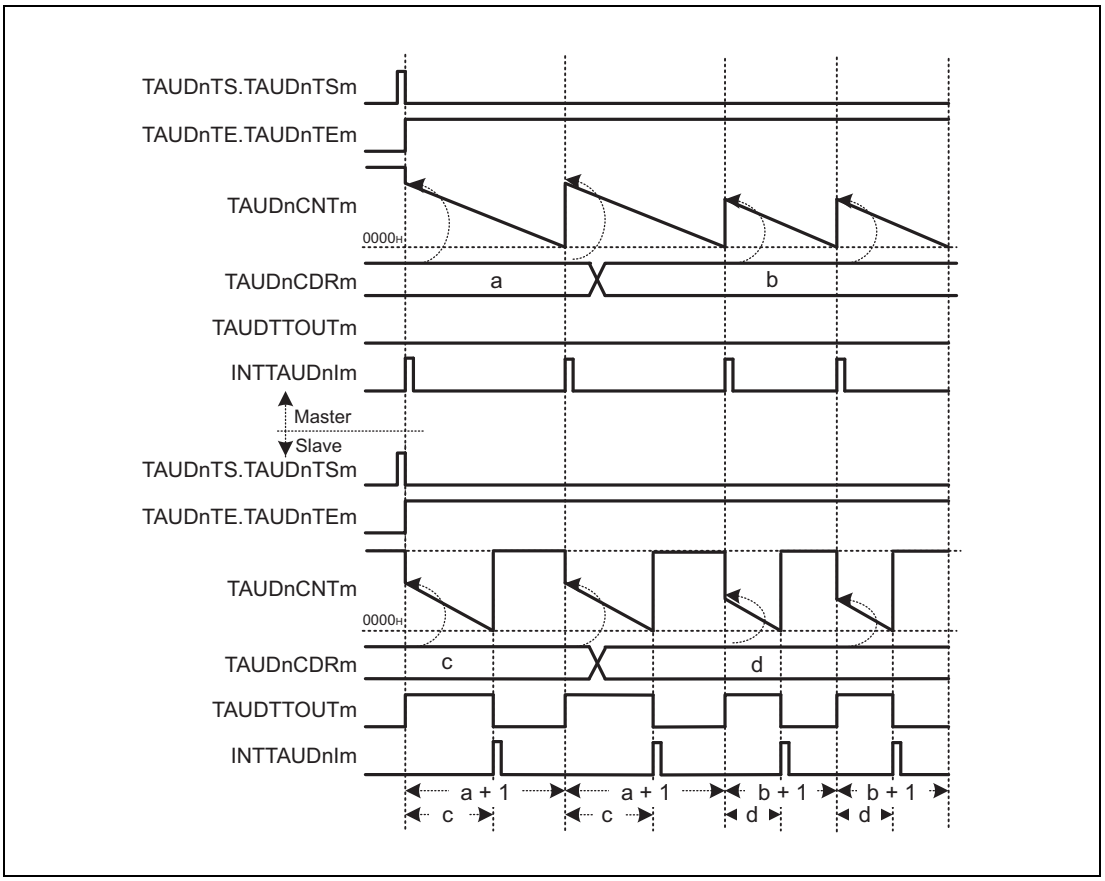


Figure 25.91 General Timing Diagram of PWM Output Function

NOTES

1. The interval between the starting to count and an interrupt being generated is the value of corresponding TAUDnCDRm + 1.
2. TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

25.15.1.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.141 Contents of the TAUDnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the master channel

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

Table 25.142 Contents of the TAUDnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the master channel

The channel output mode is not used with this function.

(4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.143 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

Use with TAUDnRDS.TAUDnRDSm bit = 1 requires a channel higher than the master channel that operates with **Section 25.14.1, Simultaneous Rewrite Trigger Generation Function Type 1**.

Conduct operation settings under the following conditions:

- Simultaneous rewrite trigger output function type 1 setting channel: TAUDnRDCm = 1, TAUDnRDSm = 1
  TAUDnCDRm settings for this channel are as follows:
  = ((TAUDnCDR setting for the master channel targeted for simultaneous rewrite + 1) × interrupt count) – 1
- Master channel: TAUDnRDCm = 0, TAUDnRDSm = 1
- Slave channel: TAUDnRDCm = 0, TAUDnRDSm = 1

If TAUDnCDRm (slave) setting > TAUDnCDRm (master) setting + 1, the duty value (which exceeds 100%) is aggregated to be 100% output.

25.15.1.5 Register Settings for Slave Channels

(1) TAUDnCMORm for slave channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.144 Contents of the TAUDnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid.

(2) TAUDnCMURm for slave channels

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

Table 25.145 Contents of the TAUDnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channels

Table 25.146 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for slave channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.147 Simultaneous Rewrite Settings for Slave Channels of PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.1.6 Operating Procedure for PWM Output Function

Table 25.148 Operating Procedure for PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.1.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.1.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel and TAUDTTOUTm (slave) is set.
During operation	TAUDnCDRm can be changed at any time. TAUDnTOL.TAUDnTOLm can be changed. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCDRm value is loaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is loaded into TAUDnCNTm (slave) to perform counting down. • TAUDTTOUTm (slave) is set to the active level. If TAUDnCNTm (slave) reaches 0000_H: • INTTAUDnIm (slave) is generated. • TAUDTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

25.15.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

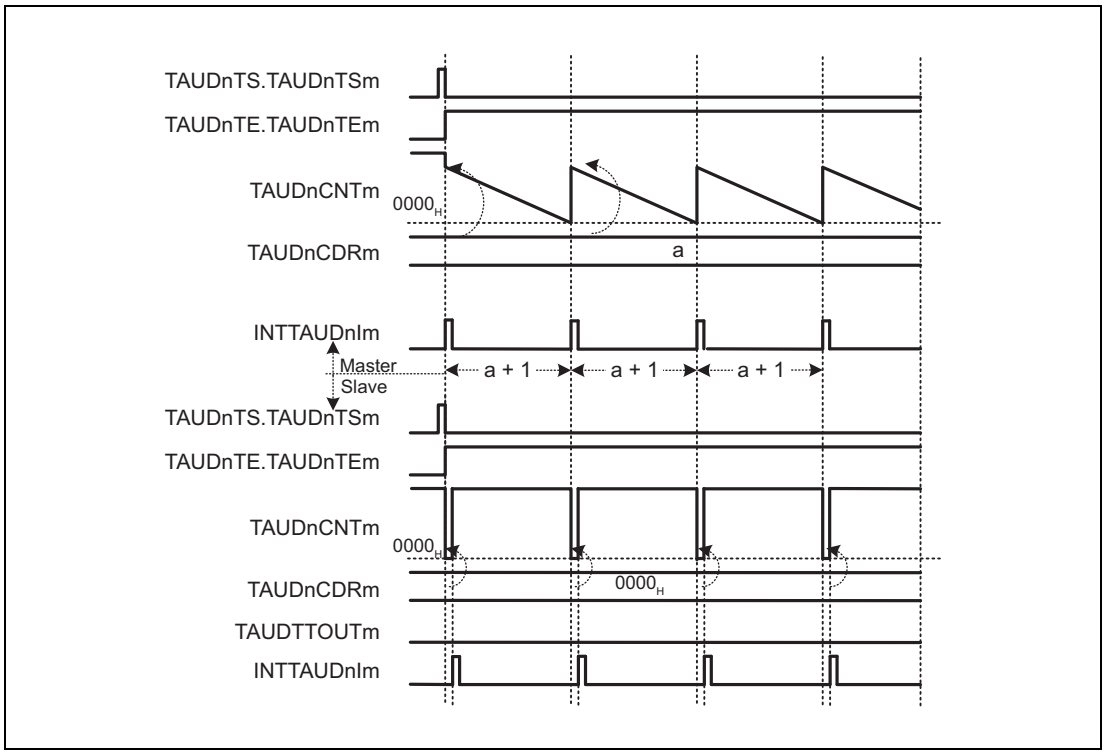


Figure 25.92 TAUDnCDRm (Slave) = 0000_H, Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- Every time the master channel generates an interrupt (INTTAUDnIm), 0000_H is loaded into TAUDnCNTm (slave). As a result, a slave channel interrupt (INTTAUDnIm) is generated at the same time and TAUDTTOUTm remains inactive.
- TAUDnCDRm value is loaded into TAUDnCNTm (slave) to generate an interrupt.

(2) Duty cycle = 100%

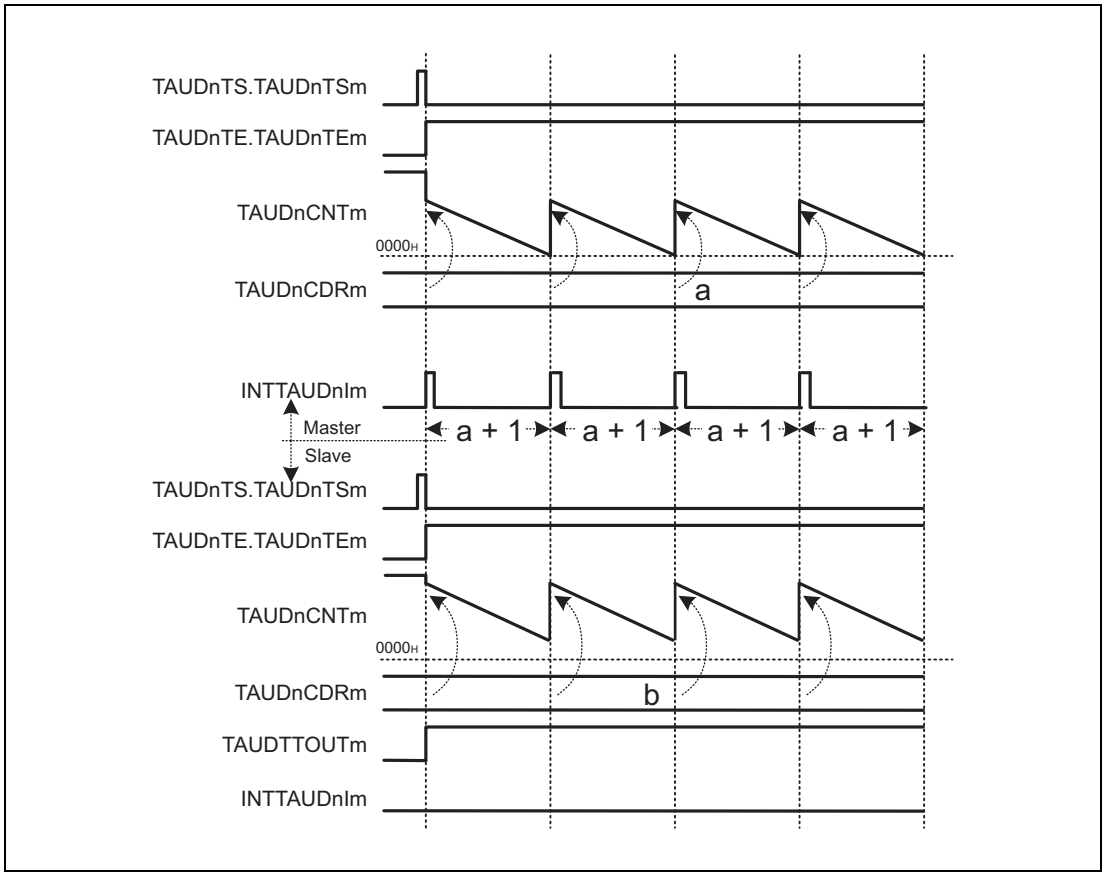


Figure 25.93 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- If TAUDnCDRm (slave) value is greater than TAUDnCDRm (master) value, the slave channel counter does not reach 0000_H and consequently, no interrupt occurs. TAUDTTOUTm remains active.

(3) Operation stop and restart

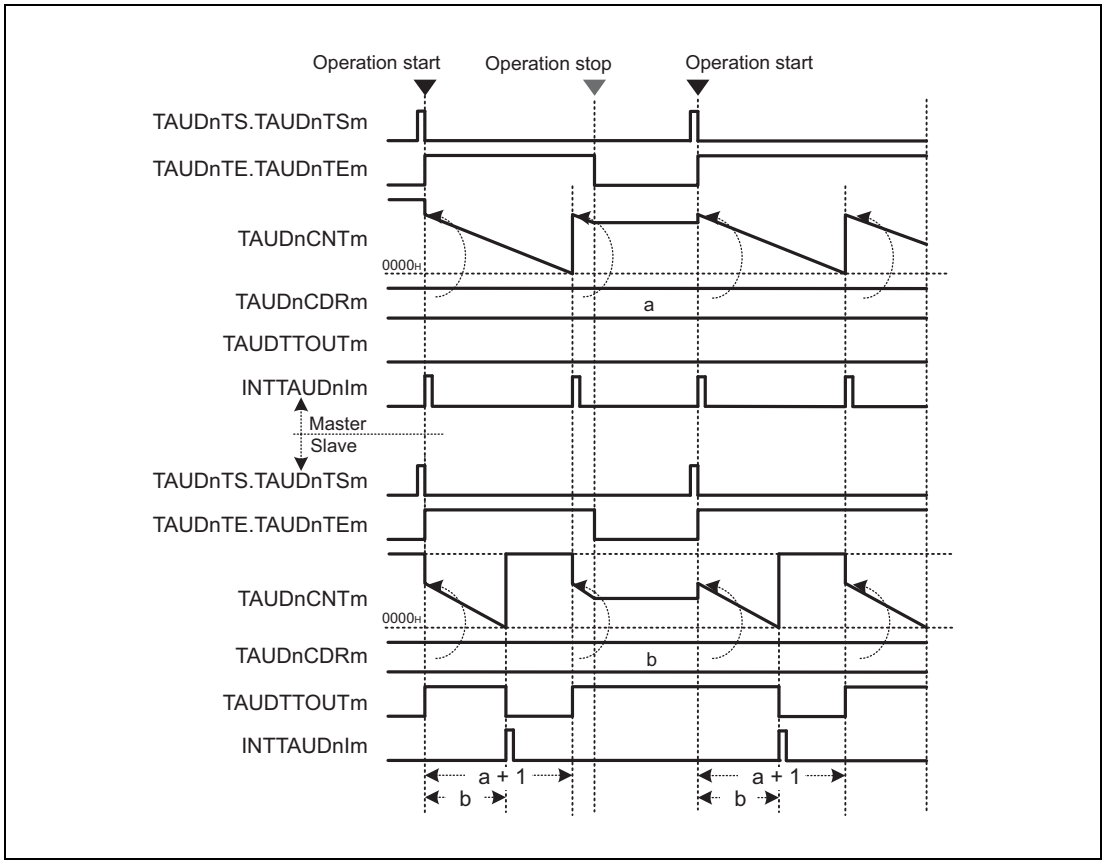


Figure 25.94 Operation Stop and Restart
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0.
- TAUDnCNTm and TAUDTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUDnTS.TAUDnTSm of master and slave channels to 1. TAUDnCDRm values of the master and slave channels are loaded to TAUDnCNTm and start to count down from these values.

25.15.2 One-Shot Pulse Output Function

25.15.2.1 Overview

Summary

This function outputs a signal pulse with a specific pulse width and delay time (both defined relative to an external input signal pulse) by using a master and a slave channel. The delay time is specified using the master channel. The pulse width is specified using the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to one-count mode. (See **Table 25.149, Contents of the TAUDnCMORm Register for the Master Channel of the One-Shot Pulse Output Function**.)
- The operating mode for slave channels should be set to pulse one-count mode. (See **Table 25.152, Contents of the TAUDnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function**.)
- TAUDTTOUTm is not used with the master channel of this function.
- The channel output mode for the slave channel should be set to independent channel output mode 2. (See **Section 25.7, Channel Output Modes**.)
- TAUDTTINm (master) has to be detected while TAUDnCNTm (master) and TAUDnCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUDTTINm (slave).

Functional description

The counters are enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) to 1 for the master and slave channels. This sets TAUDnTE.TAUDnTEm to 1, enabling count operation.

- Master channel:
  When the next valid TAUDTTINm input edge is detected, the current value of TAUDnCDRm is loaded into TAUDnCNTm. The counter starts to count down from this value. If TAUDnCMORm.TAUDnMD0 = 0, a trigger (TAUDTTINm) which is detected within the delay time is ignored.
  When the counter of master channel reaches 0000_H, INTTAUDnIm is generated. The counter is reset to $FFFF_H$ and waits for the next valid TAUDTTINm input edge.
- Slave channel:
  INTTAUDnIm generated on master channel triggers the counter operation of slave channel. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave). The counter starts counting down from this value. An interrupt occurs and the TAUDTTOUTm signal is set.
  When the counter reaches 0001_H, INTTAUDnIm is generated and TAUDTTOUTm signal is reset.
  The counter stops at 0000_H and waits for the next INTTAUDnIm of master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Setting TAUDnTS.TAUDnTSm to 1 while counting allows the counter to restart counting of master channel without making a stop (forced restart).

Conditions

- If TAUDnCMORn.TAUDnMD0 of master channel is set to 0, TAUDTTINm input edges detected during counting are ignored.
- Simultaneous rewrite can be used with this function. See **Section 25.6, Simultaneous Rewrite**.

25.15.2.2 Equations

Delay from trigger input to pulse output
= (TAUDnCDRm (master) + 1) × count clock cycle

Pulse width = (TAUDnCDRm (slave)) × count clock cycle

25.15.2.3 Block Diagram and General Timing Diagram

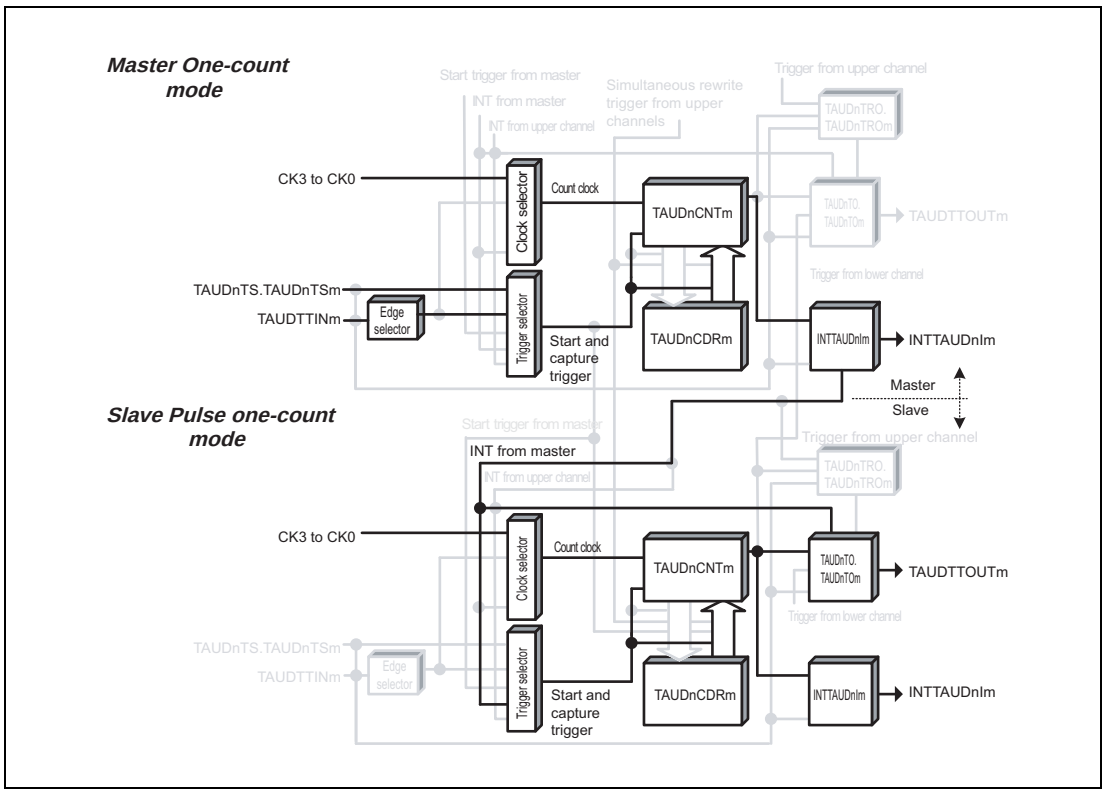


Figure 25.95 Block Diagram of One-Shot Pulse Output Function

The following settings apply to the general timing diagram.

- Start trigger detection is disabled during counting (TAUDnCMORm.TAUDnMD0 = 0).
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

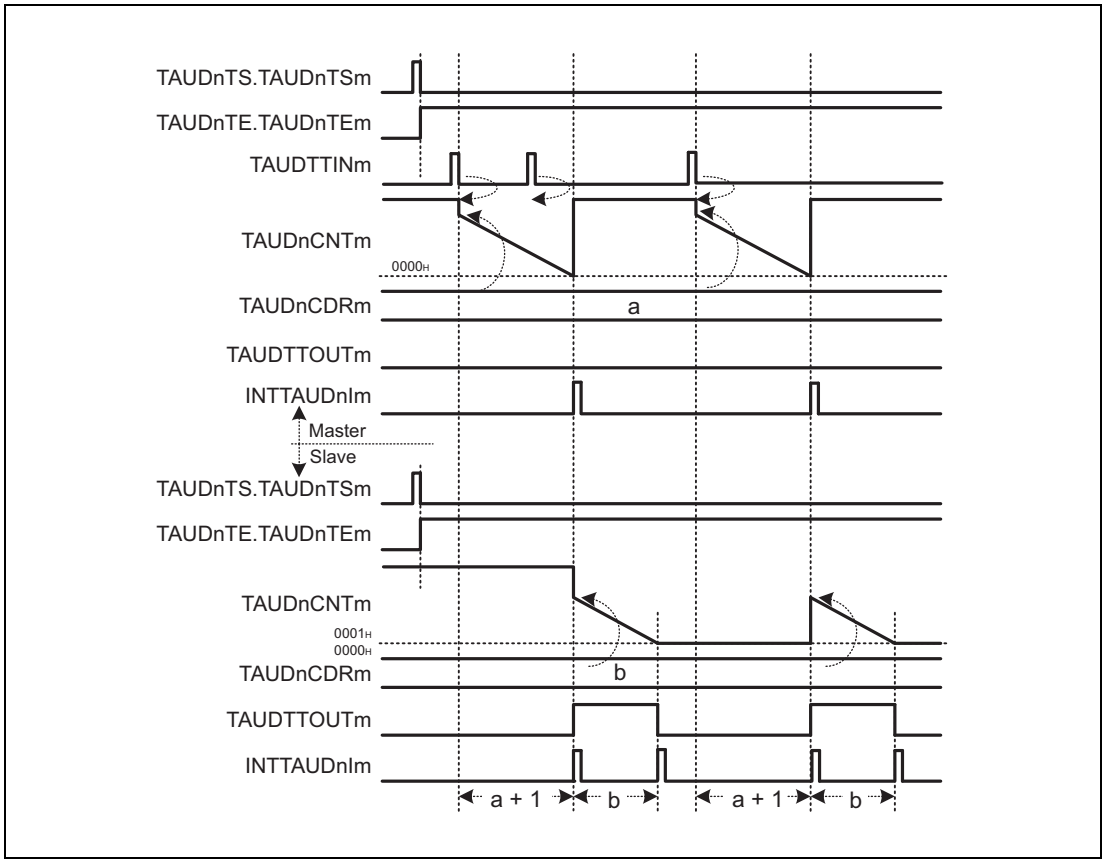


Figure 25.96 General Timing Diagram of One-Shot Pulse Output Function

25.15.2.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.149 Contents of the TAUDnCMORm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	0: Disables detection of start trigger during count operation. 1: Enables detection of start trigger during count operation. The MD0 bit of master and slave channels should have the same value.

(2) TAUDnCMURm for the master channel

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

Table 25.150 Contents of the TAUDnCMURm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode for the master channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.151 Simultaneous Rewrite Settings for the Master Channel of One-Shot Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.2.5 Register Settings for Slave Channels

(1) TAUDnCMORm for slave channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.152 Contents of the TAUDnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMD0	0: Disables detection of start trigger during count operation. 1: Enables start trigger detection while counting. The MD0 bit of master and slave channels should have the same value.

(2) TAUDnCMURm for slave channels

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

Table 25.153 Contents of the TAUDnCMURm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the slave channel

Table 25.154 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTDL.TAUDnTDLm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set this bit to 0
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for slave channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.155 Simultaneous Rewrite Settings for Slave Channels of One-Shot Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.2.6 Operating Procedure for One-Shot Pulse Output Function

Table 25.156 Operating Procedure for One-Shot Pulse Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.2.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and channel output mode as described in **Section 25.15.2.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the master channel awaits a TAUDTTINm input.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	When valid TAUDTTINm input edge is detected, TAUDnCDRm value of master channel is loaded into TAUDnCNTm to perform counting down. When the counter reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCNTm (master) is reset to $FFFF_H$ and waits for the next valid TAUDTTINm input edge. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave) to start counting. • INTTAUDnIm (slave) is generated. • TAUDTTOUTm (slave) is set to the active level. When TAUDnCNTm (slave) reaches 0001_H: • INTTAUDnIm (slave) is generated. • TAUDTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

25.15.2.7 Specific Timing Diagrams

(1) TAUDnCDRm (master) = 0000_H

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

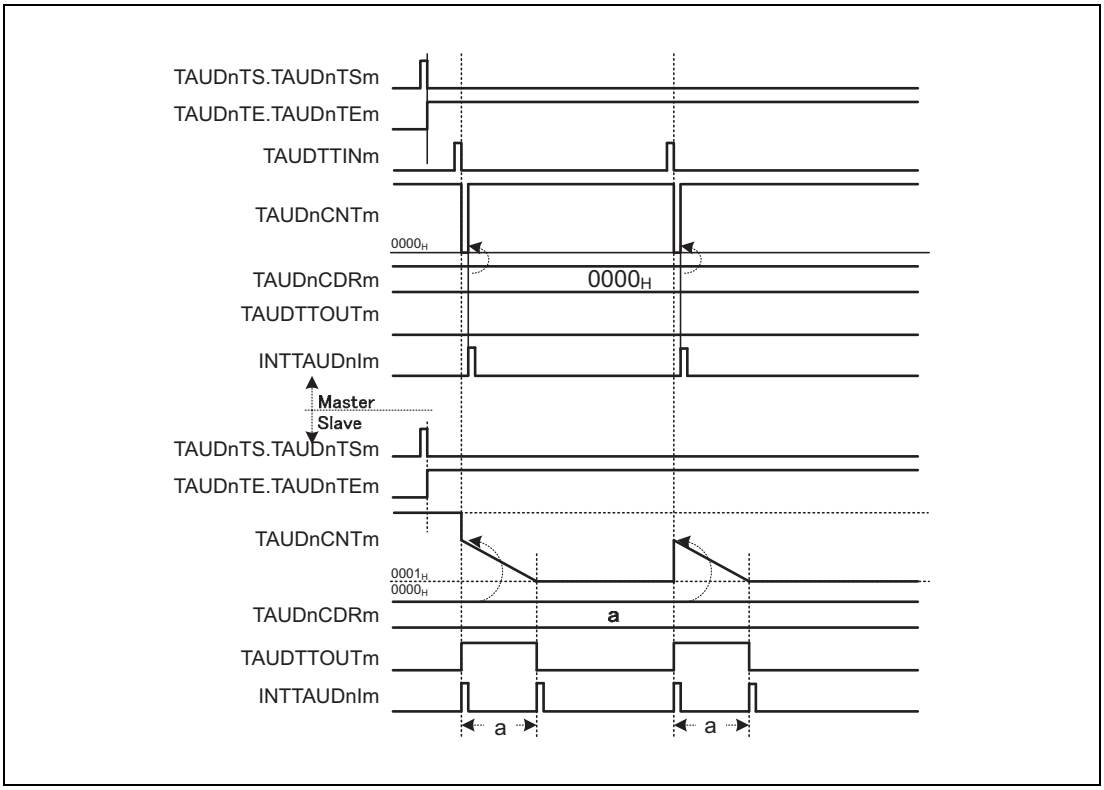


Figure 25.97 TAUDnCDRm (Master) = 0000_H

- When a valid TAUDTTINm input edge is detected, the value 0000_H is written to TAUDnCNTm (master). The counter is set to 0000_H for one count and returns to $FFFF_H$.
  Thus the slave channel starts to count down one count clock later than TAUDTTINm (master).

(2) TAUDnCDRm (slave) = 0000_H

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

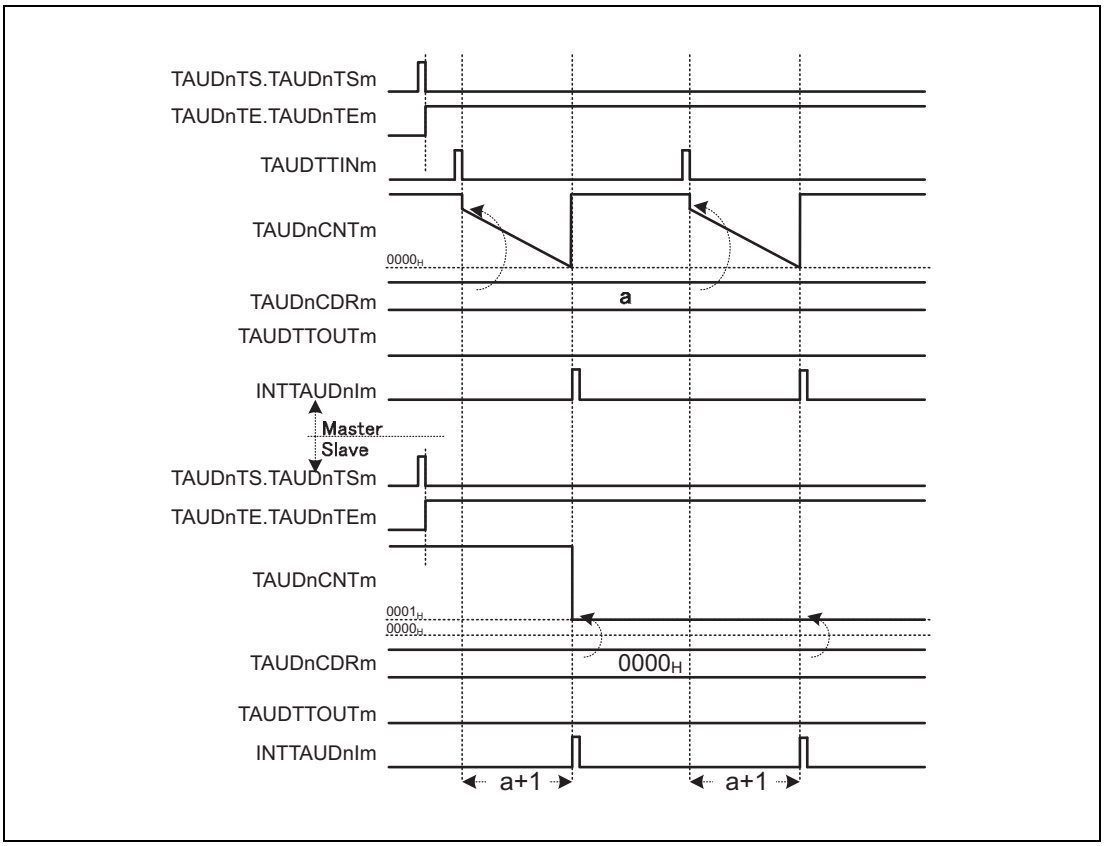


Figure 25.98 TAUDnCDRm (Slave) = 0000_H

- TAUDTTOUTm remains inactive, because the pulse width is zero.

(3) TAUDnCMORm.TAUDnMD0 = 1

The following settings apply to this diagram.

- Enables start trigger detection while counting. (TAUDnCMORm.TAUDnMD0 = 1)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

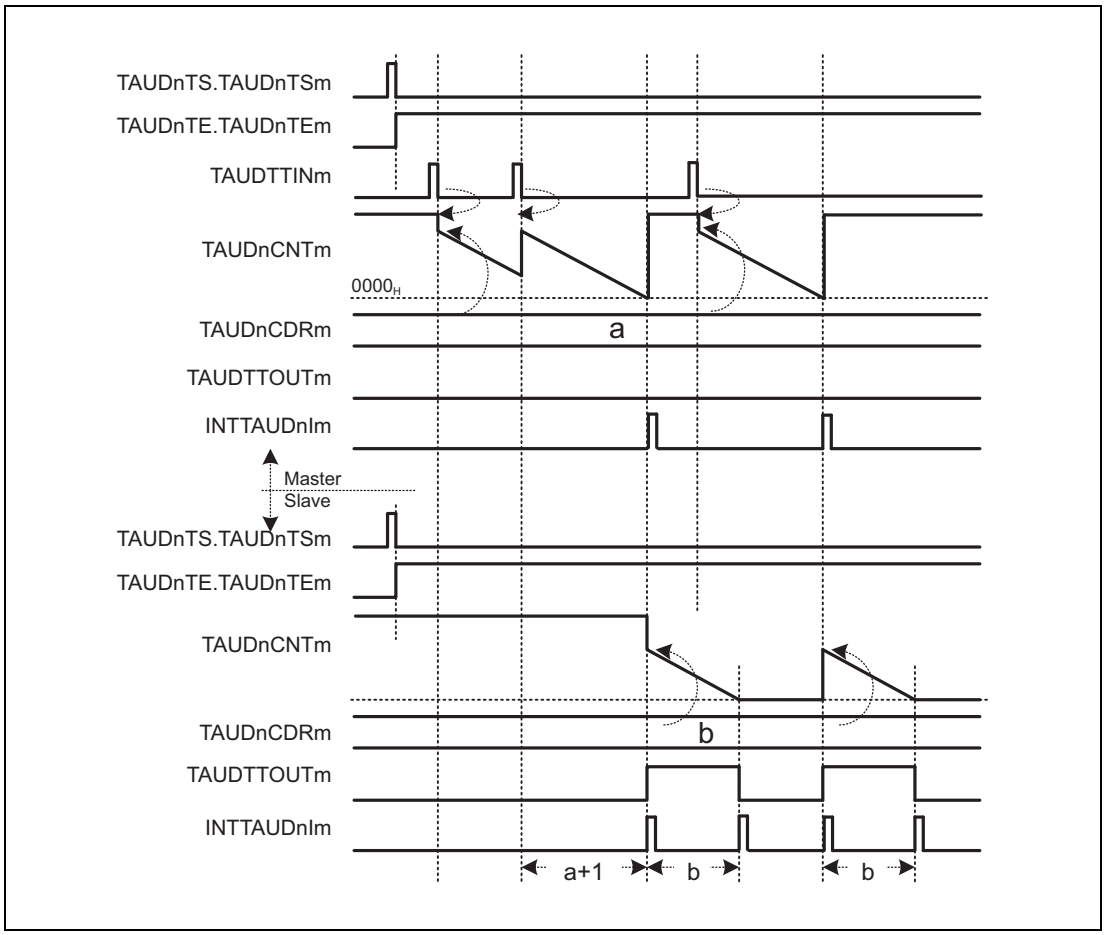


Figure 25.99 TAUDnCMORm.TAUDnMD0 = 1

- If a valid TAUDTTINm input edge is detected while the counter of the master channel counts down, TAUDnCNTm reloads the value of TAUDnCDRm. The counter restarts to count down. This means the delay is extended by the value of TAUDnCNTm at the time when a valid TAUDTTINm input edge is detected.

(4) Restarting the master channel while the slave channel is counting

The following settings apply to this diagram.

- Disables detection of start trigger during count operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

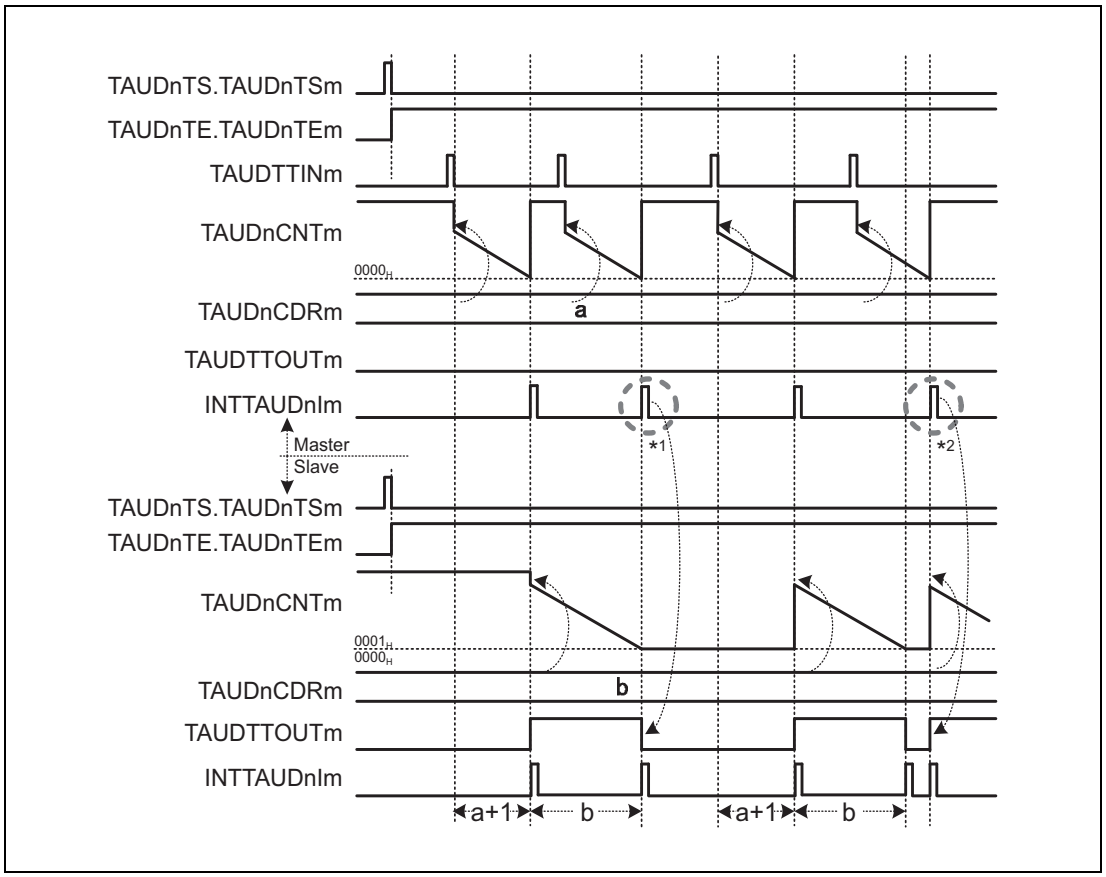


Figure 25.100 TAUDTTINm input interval ≤ Delay Time + Pulse Width +1

- If the master channel generates an interrupt before the counter of the slave channel has reached 0001_H or exactly when 0001_H is reached[*1], the interrupt (master) is ignored.
- If an interrupt of the master channel occurs when the counter of the slave channel awaits the next trigger, the value of TAUDnCDRm (slave) is reloaded. An interrupt is generated and TAUDTTOUTm toggles. If TAUDnCNTm (master) has started to count down while the TAUDnCNTm (slave) is still counting[*2], TAUDTTOUTm is not output with the expected delay time.
- To generate the correct one-shot pulse, the start trigger for the master channel must be detected while the master and slave channels are waiting for the start trigger, and not while they are counting.

25.15.3 Trigger Start PWM Output Function

25.15.3.1 Overview

Summary

This function generates a PWM output using a master and a slave channel. It enables the pulse cycle (frequency) and the duty of the TAUDTTOUTm to be set. The pulse cycle is specified using the master channel. The duty is specified using the slave channel. The Trigger Start PWM Output Function is identical to PWM Output Function except that the master channel of this function can be reset by a valid TAUDTTINm input edge.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to Interval Timer Mode (see **Table 25.157, Contents of the TAUDnCMORm Register for the Master Channel of the Trigger Start PWM Output Function**).
- The operation mode of the slave channel must be set to One-Count Mode (see **Table 25.160, Contents of the TAUDnCMORm Register for the Slave Channel of the Trigger Start PWM Output Function**).
- The channel output mode of the slave channel must be set to Synchronous Channel Output Mode 1 (see **Section 25.7, Channel Output Modes**).
- TAUDTTOUTm is not used with the master channel of this function.

Functional description

The counters (master and slave) are enabled by setting the channel trigger bits (TAUDnTS.TAUDnTSm) to 1. This in turn sets TAUDnTE.TAUDnTEm to 1, enabling count operation. The current value of TAUDnCDRm is loaded to TAUDnCNTm, and the counter starts to count down from this value. INTTAUDnIm is generated on the master channel, and a PWM output is recreated by setting and resetting TAUDTTOUTm (slave).

- Master channel:
  The current value of TAUDnCDRm is loaded to the counter (TAUDnCNTm), INTTAUDnIm is generated and the counter starts to count down from this value.
  When the counter reaches 0000_H and the pulse cycle time has elapsed, INTTAUDnIm is generated and the counters (master and slave) load the current TAUDnCDRm values.
  If a valid TAUDTTINm input edge is detected, the counter of the master channel loads the current TAUDnCDRm value, restarts counting down and generates an interrupt.
- Slave channel:
  When the slave detects an interrupt from the master channel, it starts to count down from the current value of TAUDnCDRm. The TAUDTTOUTm signal is set to the active level.
  When the counter reaches 0000_H (duty time has elapsed), INTTAUDnIm is generated and the TAUDTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUDnIm of the master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1 for the master and slave channel, which in turn sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channel stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. See **Section 25.6, Simultaneous Rewrite**.

25.15.3.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = [TAUDnCDRm (slave) / (TAUDnCDRm (master) + 1)] × 100

– Duty cycle = 0%
  TAUDnCDRm (slave) = 0000_H
– Duty cycle = 100%
  TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

25.15.3.3 Block Diagram and General Timing Diagram

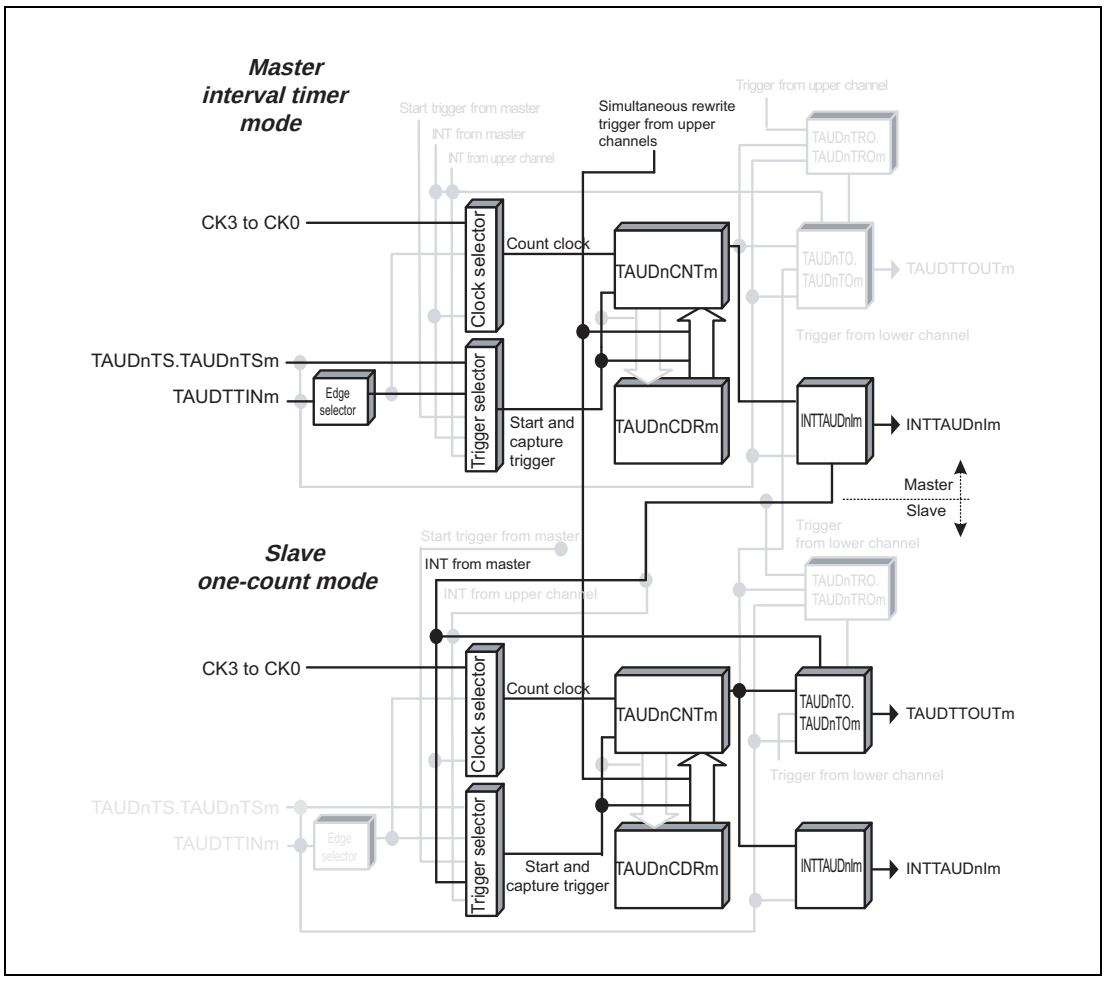


Figure 25.101 Block Diagram for Trigger Start PWM Output Function

The following settings apply to the general timing diagram.

- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)
- Positive logic (TAUDnTOL.TAUDnTOLm (slave) = 0)

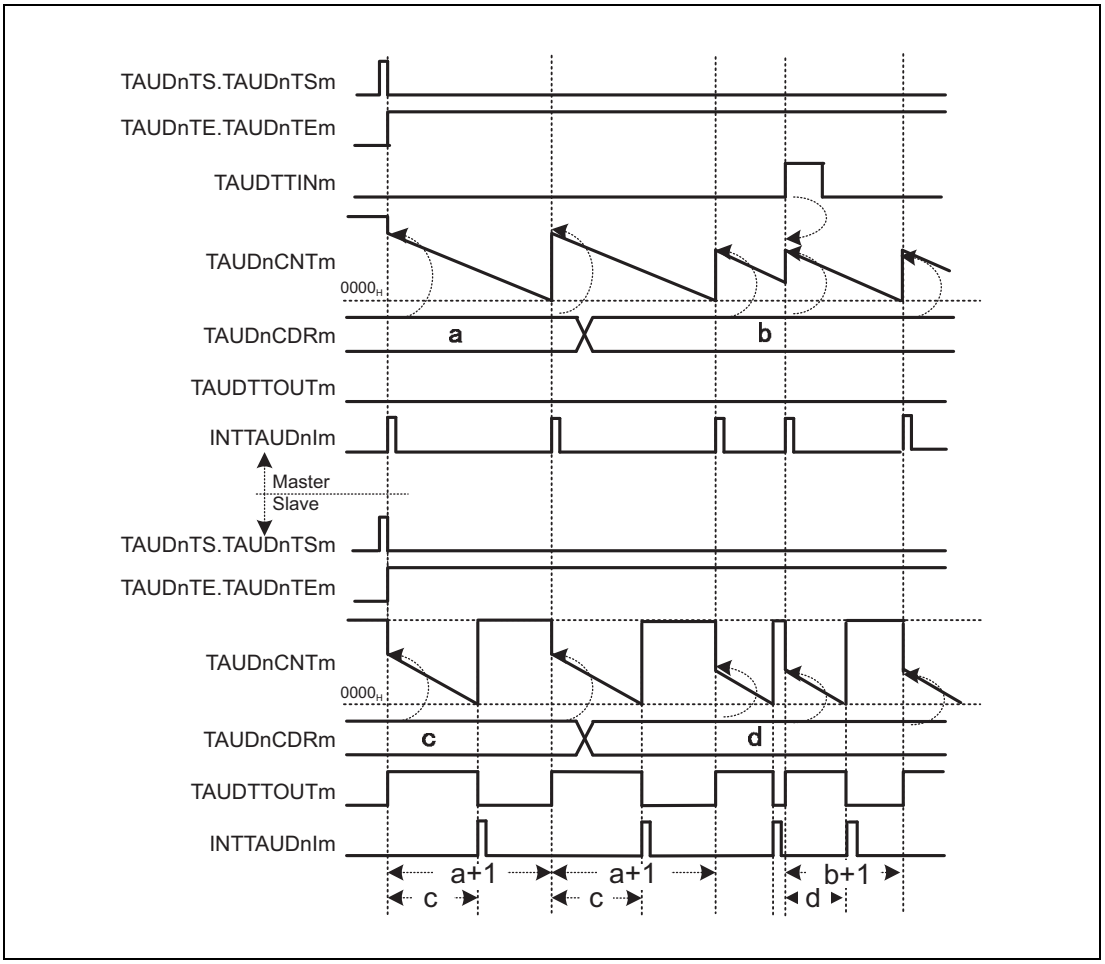


Figure 25.102 General Timing Diagram for Trigger Start PWM Output Function

NOTE

TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

25.15.3.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.157 Contents of the TAUDnCMORm Register for the Master Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the master channel

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

Table 25.158 Contents of the TAUDnCMURm Register for the Master Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode for the master channel

The channel output mode is not used by this function.

(4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.159 Simultaneous Rewrite Settings for the Master Channel of the Trigger Start PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.3.5 Register Settings for Slave Channels

(1) TAUDnCMORm for slave channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.160 Contents of the TAUDnCMORm Register for the Slave Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid. The value of the TAUDnMD[0] bit of the master and slave channel must be identical.

(2) TAUDnCMURm for slave channels

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

Table 25.161 Contents of the TAUDnCMURm Register for the Slave Channel of the Trigger Start PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Output mode for the slave channel

Table 25.162 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for slave channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.163 Simultaneous Rewrite Settings for the Slave Channel of the Trigger Start PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.3.6 Operating Procedure for Trigger Start PWM Output Function

Table 25.164 Operating Procedure for Trigger Start PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.3.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.3.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCDRm value is loaded into TAUDnCNTm (master) to continue count operation. • TAUDnCNTm (slave) loads the TAUDnCDRm value and starts to count down • TAUDTTOUTm (slave) is set When TAUDnCNTm of the slave = 0000_H: • INTTAUDnIm (slave) is generated. • TAUDTTOUTm (slave) is set to an inactive level. In addition, the counter of slave channel stops. If a TAUDTTINm input is detected on the master channel while TAUDnCNTm of the master channel is counting down: • TAUDnCNTm (master and slave) loads the TAUDnCDRm value and counts down • INTTAUDnIm (master) is generated. • TAUDTTOUTm (slave) is set to the active level.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

25.15.3.7 Specific Timing Diagrams

(1) Duty cycle = 0%

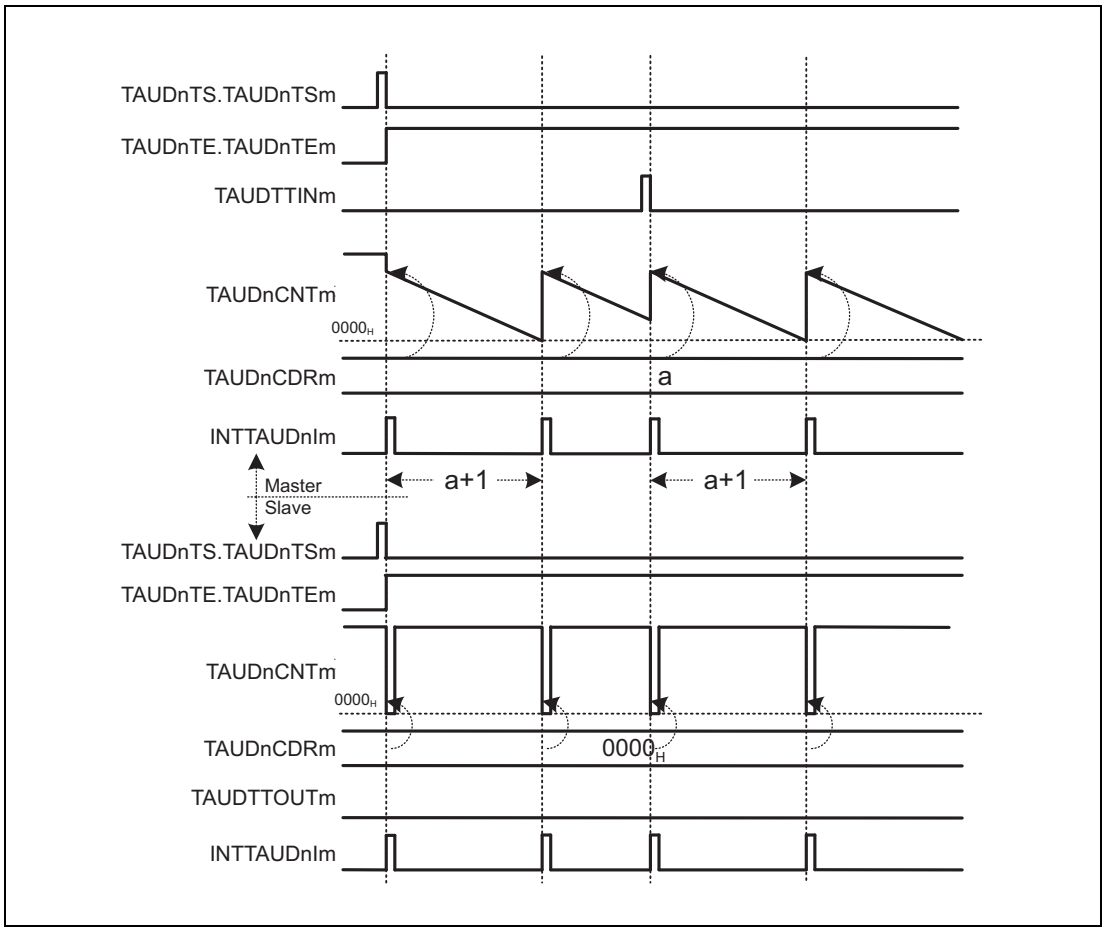


Figure 25.103 TAUDnCDRm (Slave) = 0000_H, Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0) Detection of Falling Edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- Every time the master channel generates an interrupt (INTTAUDnIm), 0000_H is loaded to TAUDnCNTm (slave). Therefore, TAUDnCNTm (slave) cannot start to count and TAUDTTOUTm remains inactive.
- TAUDnCNTm (slave) generates an interrupt every time the value of TAUDnCDRm is reloaded. The detection of a valid TAUDTTINm input edge has no effect on TAUDTTOUTm (slave).

(2) Duty cycle = 100%

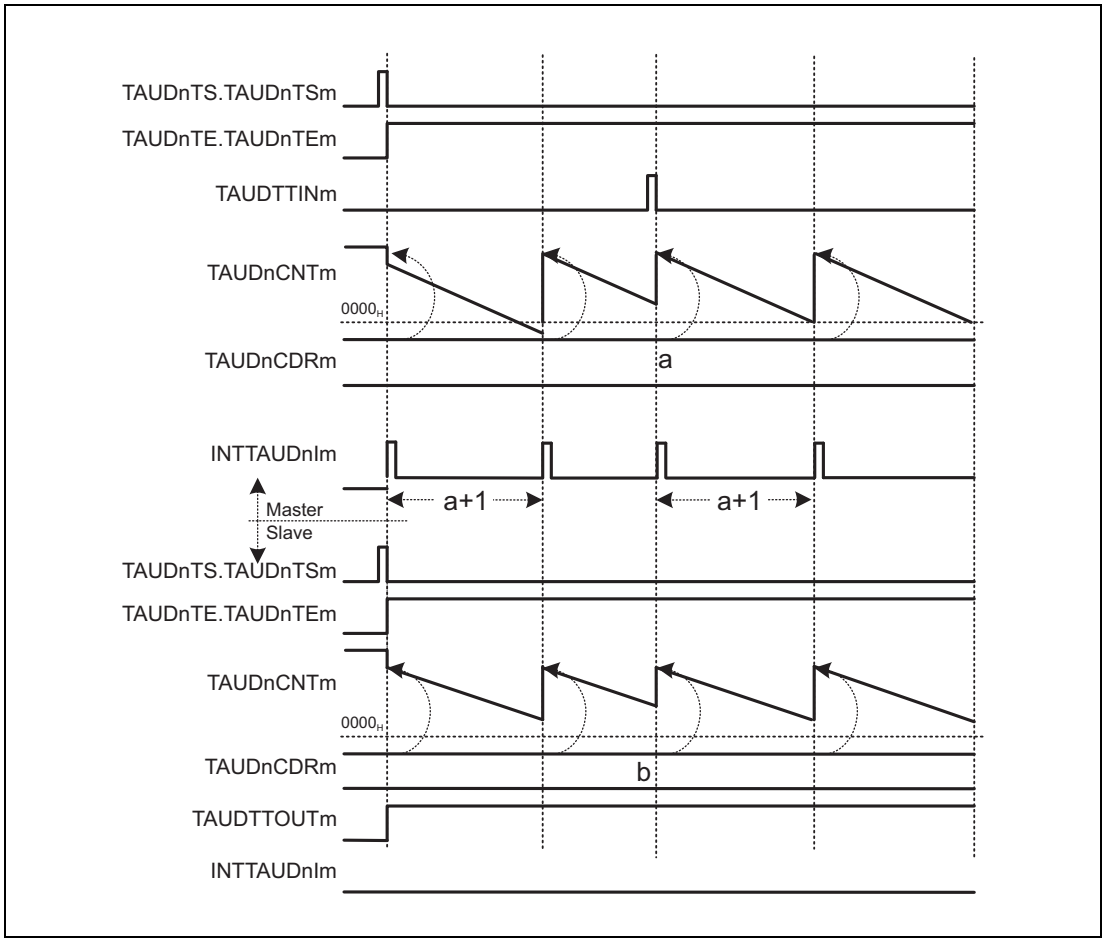


Figure 25.104 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1, Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0) Falling Edge Detection (TAUDnCMURm.TIS[1:0] = 00_B)

- If the value TAUDnCDRm (slave) is higher than the value TAUDnCDRm (master), the counter of the slave channel cannot reach 0000_H and cannot generate interrupts.
  The TAUDTTOUTm remains in an active state.
  The detection of a valid TAUDTTINm input edge has no effect on TAUDTTOUTm (slave).

(3) TAUDTTINm detection and active slave counter

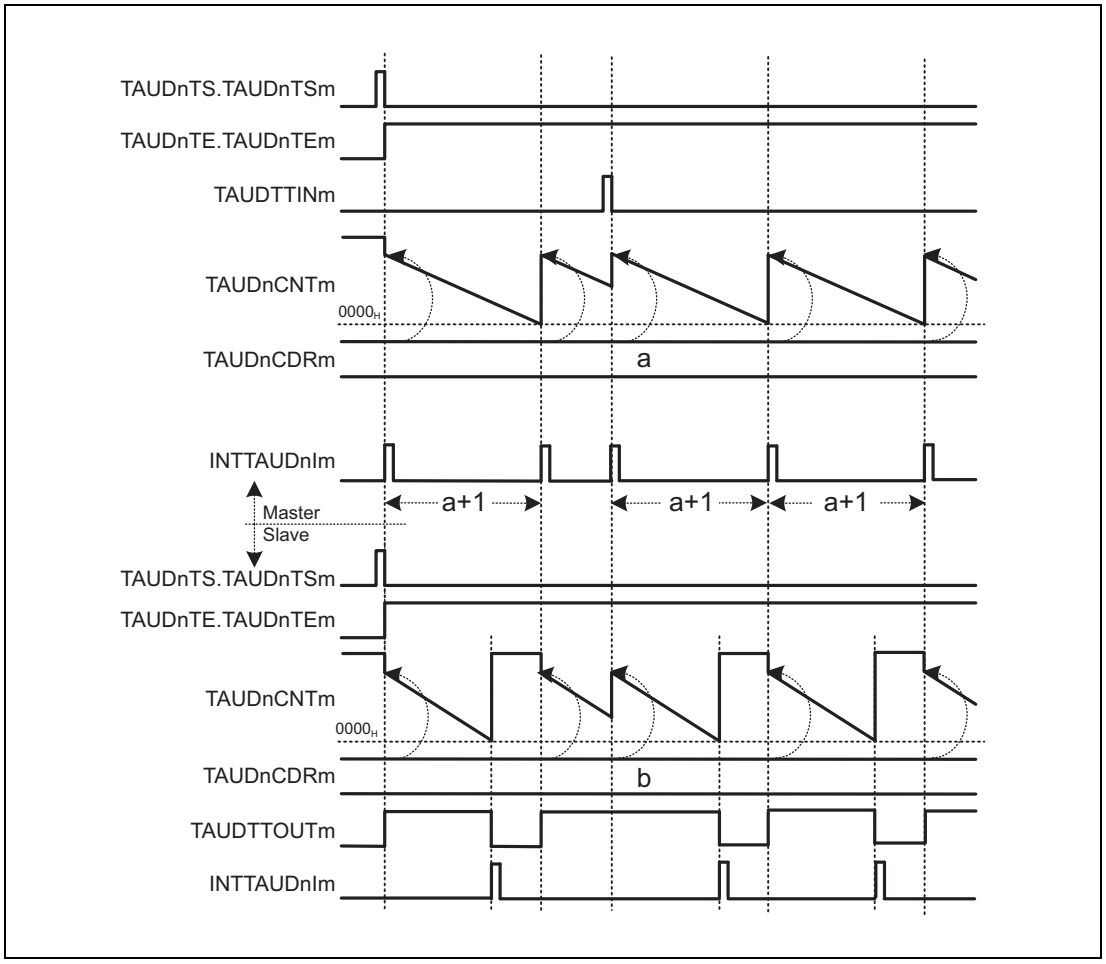


Figure 25.105 Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)
Detection of Falling Edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- If TAUDnCNTm (slave) loads the value TAUDnCDRm (slave) while it is still counting down, TAUDTTOUTm remains unchanged and extends the duty.
  The duty does not correspond to the value of the slave's data register.

25.15.4 Delay Pulse Output Function

25.15.4.1 Overview

Summary

This function outputs two signals. The pulse width and pulse cycle of the reference signal are defined using the master channel and slave channel 1. Slave channels 2 and 3 output the reference signal with a specified delay. The delay signal is identical to the reference signal, but delayed by the amount specified on slave channel 2.

The signal values are specified in the following way:

- The pulse cycle is specified using the master channel.
- The duty cycle of the reference signal is specified using slave channel 1. The duty cycle of the delay signal is specified using slave channel 3.
- The delay is specified on slave channel 2.

Prerequisites

- Four channels
- The operating mode for the master channel should be set to interval timer mode. (See **Table 25.165, Contents of the TAUDnCMORm Register for the Master Channel of the Delay Pulse Output Function**.)
- The operating mode for slave channels 1 and 2 should be set to one-count mode. (See **Table 25.168, Contents of the TAUDnCMORm Register for Slave Channel 1 of the Delay Pulse Output Function** and **Table 25.172, Contents of the TAUDnCMORm Register for Slave Channel 2 of the Delay Pulse Output Function**.)
- The operating mode for slave channel 3 should be set to pulse one-count mode. (See **Table 25.175, Contents of the TAUDnCMORm Register for Slave Channel 3 of the Delay Pulse Output Function**.)
- TAUDTTOUTm is not used with the master channel and slave channel 2.
- The channel output mode for slave channel 1 should be set to synchronous channel output mode 1. (See **Section 25.7, Channel Output Modes**.)
- The channel output mode for slave channel 3 should be set to independent channel output mode 2. (See **Section 25.7, Channel Output Modes**.)

Functional description

The counters of the channel group are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm to 1, enabling count operation.

- Master channel:
  The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from this value. INTTAUDnIm is generated on the master channel.
  When the counter value of master channel reaches 0000_H and pulse cycle time has elapsed, INTTAUDnIm is generated. The TAUDnCDRm value is reloaded into the counter to perform counting down.
- Slave channels 1 and 2:
  Slave channels 1 and 2 start to count down from the current TAUDnCDRm value when detecting

an interrupt from the master channel. TAUDTTOUTm signal (slave 1) is set.

– Slave channel 1:
When the counter of slave channel 1 reaches 0000_H (duty time has elapsed), INTTAUDnIm is generated and TAUDTTOUTm signal is reset. The counter is reset to $FFFF_H$ and waits for the next INTTAUDnIm of master channel.

– Slave channel 2:
When the counter of slave channel 2 reaches 0000_H and delay time has elapsed, INTTAUDnIm is generated. The counter is reset to $FFFF_H$ and waits for the next INTTAUDnIm of master channel.
Generating INTTAUDnIm (slave channel 2) triggers the counter of slave channel 3.

- Slave channel 3:
When slave channel 3 detects an interrupt from slave channel 2, its counter starts counting down from the current value of TAUDnCDRm. INTTAUDnIm is generated and the TAUDTTOUTm signal (slave channel 3) is set.
When the counter of slave channel 3 reaches 0001_H, INTTAUDnIm is generated and the TAUDTTOUTm signal is reset.
The delayed PWM pulse is output from slave channel 3.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but their values are retained. The counter can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. See **Section 25.6, Simultaneous Rewrite**.

25.15.4.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUDnCDRm (slave 1)) × count clock cycle

Delay width = (TAUDnCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUDnCDRm (slave 3)) × count clock cycle

However, the delay width shall be set within the following range:
$0000_H \leq$ TAUDnCDRm (slave 2) < TAUDnCDRm (master)

NOTES

1. The waveform of TAUDTTOUTm (slave 3) becomes the waveform made by delaying the waveform of TAUDTTOUTm (slave 1) by the quantity generated by slave 2. It is impossible to make a delay longer than the pulse cycle.
2. If INTTAUD0Im of slave 2 is generated while slave 3 is counting, slave 3 restarts operation. Therefore, the waveform of TAUDTTOUTm (slave 3) is retained on the active level. In this case, TAUDTTOUTm (Slave-CH-3) cannot output the waveform generated by delaying the basic pulse of TAUDTTOUTm (Slave-CH-1).

25.15.4.3 Block Diagram and General Timing Diagram

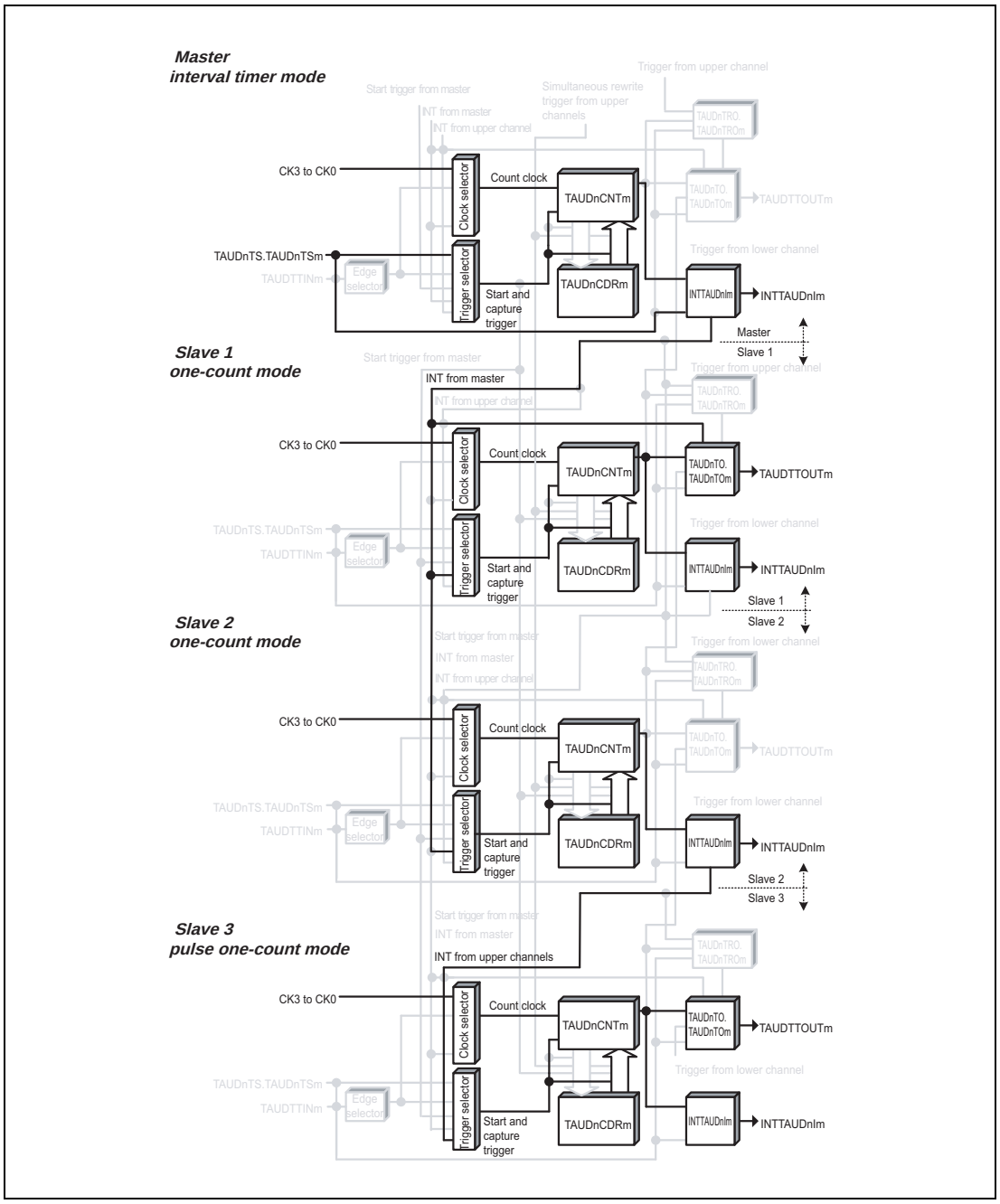


Figure 25.106 Block Diagram of Delay Pulse Output Function

The following settings apply to the general timing diagram.

- Slave channel 1: Positive logic (TAUDnTOL.TAUDnTOLm = 0)
- Slave channel 3: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

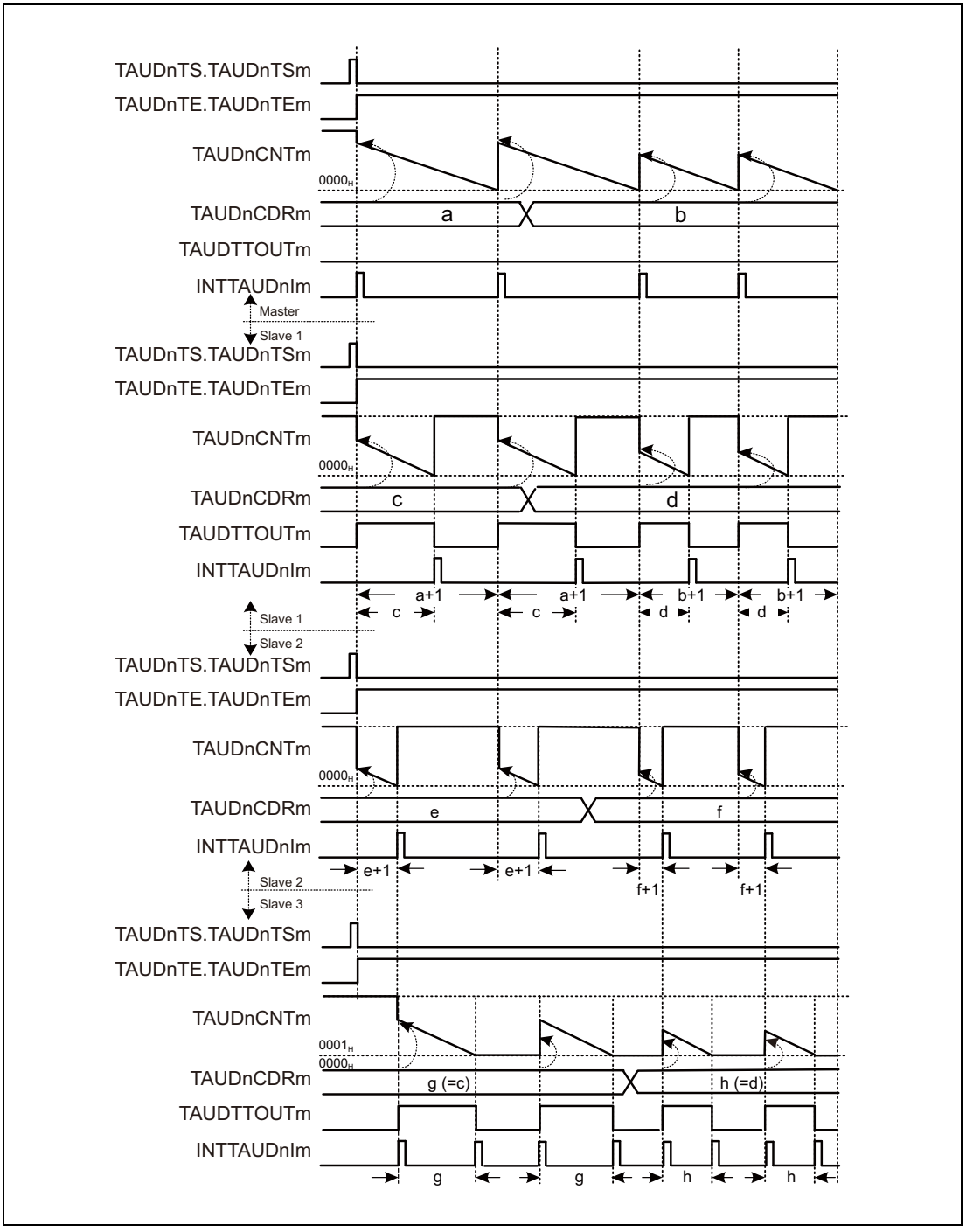


Figure 25.107 General Timing Diagram of Delay Pulse Output Function

NOTE

TAUDTTOUTm of slave channel 1 rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

25.15.4.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.165 Contents of the TAUDnCMORm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the master channel

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

Table 25.166 Contents of the TAUDnCMURm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the master channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used for the master channel with this function.

(4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.167 Simultaneous Rewrite Settings for the Master Channel of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.4.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.168 Contents of the TAUDnCMORm Register for Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Valid start trigger during operation

(2) TAUDnCMURm for slave channel 1

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

Table 25.169 Contents of the TAUDnCMURm Register for Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channel 1

Table 25.170 Control Bit Settings for Slave Channel 1 in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for slave channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.171 Simultaneous Rewrite Settings for Slave Channel 1 of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.4.6 Register Settings for Slave Channel 2

(1) TAUDnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.172 Contents of the TAUDnCMORm Register for Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Valid start trigger during operation

(2) TAUDnCMURm for slave channel 2

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

Table 25.173 Contents of the TAUDnCMURm Register for Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channel 2

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous rewrite for slave channel 2

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.174 Simultaneous Rewrite Settings for Slave Channel 2 of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.4.7 Register Settings for Slave Channel 3

(1) TAUDnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.175 Contents of the TAUDnCMORm Register for Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	101: INTTAUDnIm of upper channel (m - 1) is a start trigger regardless of master setting.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1010: Pulse one-count mode
0	TAUDnMD0	1: Valid start trigger during operation

(2) TAUDnCMURm for slave channel 3

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

Table 25.176 Contents of the TAUDnCMURm Register for Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channel 3

Table 25.177 Control Bit Settings in Independent Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for slave channel 3

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.178 Simultaneous Rewrite Settings for Slave Channel 3 of Delay Pulse Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.4.8 Operating Procedure for Delay Pulse Output Function

Table 25.179 Operating Procedure for Delay Pulse Output Function (1/2)

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.4.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.4.5, Register Settings for Slave Channel 1**. Slave channel 2: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.4.6, Register Settings for Slave Channel 2**. Slave channel 3: Set the TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.4.7, Register Settings for Slave Channel 3**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.

Table 25.179 Operating Procedure for Delay Pulse Output Function (2/2)

Restart Operation

	Operation	TAUDn Status
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master channel and slave channels 1 and 2 start. INTTAUDnIm is generated on the master channel and TAUDTTOUTm (slave channel 1) is set.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel and slave channels 1 and 2 is loaded to TAUDnCNTm and count down. When the counter of master channel reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave 1/2) to count down. • TAUDTTOUTm (slave 1) is set. When TAUDnCNTm (slave 1) reaches 0000_H: • INTTAUDnIm (slave 1) is generated. • TAUDTTOUTm (slave 1) is reset. When TAUDnCNTm (slave 2) reaches 0000_H: • INTTAUDnIm (slave 2) is generated. • INTTAUDnIm (slave 3) is generated. • TAUDTTOUTm (slave 3) is set. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave 3) to count down operation. When TAUDnCNTm (slave 3) reaches 0001_H: • INTTAUDnIm (slave 3) is generated. • TAUDTTOUTm (slave 3) is reset.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

25.15.4.9 Specific Timing Diagrams

(1) Duty cycle (slave 3) = 100%

The following values apply to **Figure 25.108**:

- TAUDnCDRm (master) = $000A_H$
- TAUDnCDRm (slave 1) = $000B_H$
- TAUDnCDRm (slave 2) = 0000_H
- TAUDnCDRm (slave 3) = $000B_H$

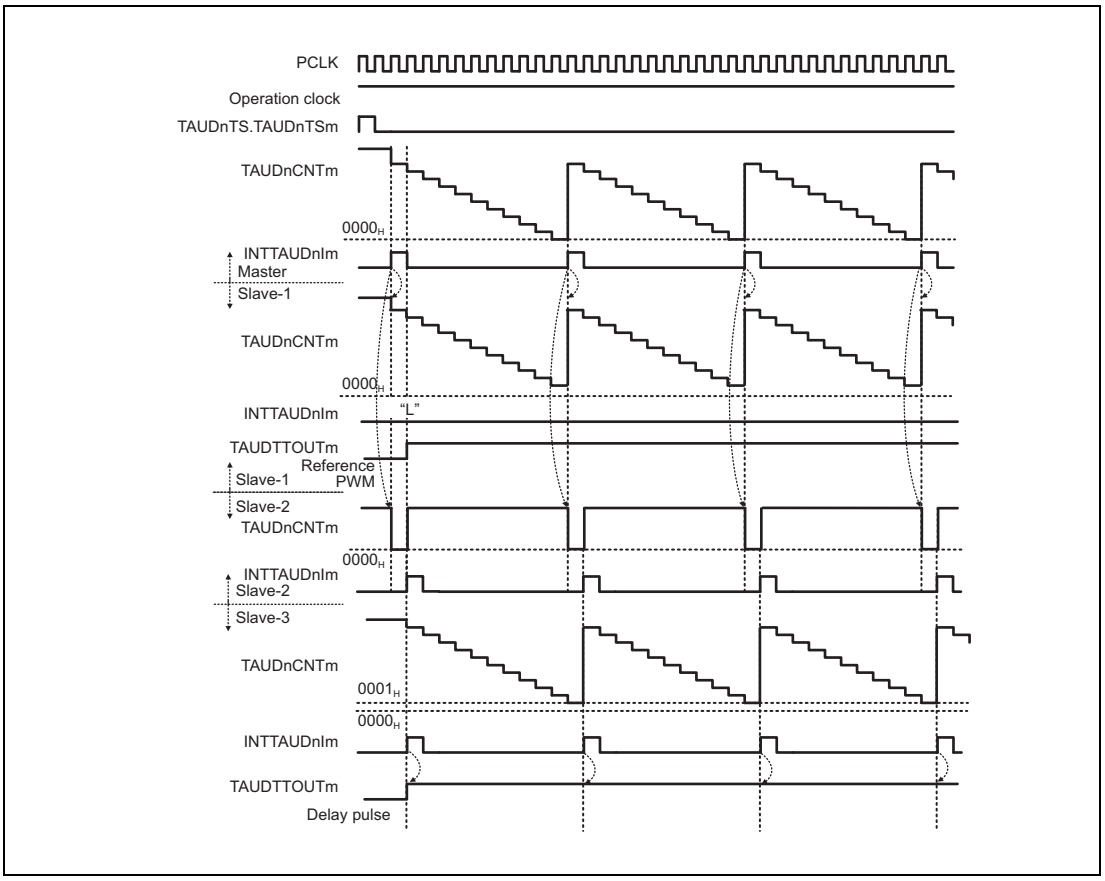


Figure 25.108 Duty Cycle (Slave 3) = 100%

- If the value of TAUDnCDRm (slaves 1 and 3) is higher than the value of TAUDnCDRm (master), the counter of the slave channel 1 cannot reach 0000_H and cannot generate interrupts. TAUDTTOUTm of channels 1 and 3 remain in the active state.

(2) TAUDTTOUTm (slave 1) = TAUDTTOUTm (slave 3)

The following values apply to **Figure 25.109**.

- TAUDnCDRm (master) = $000A_H$
- TAUDnCDRm (slave 1) = 0005_H
- TAUDnCDRm (slave 2) = 0000_H
- TAUDnCDRm (slave 3) = 0005_H

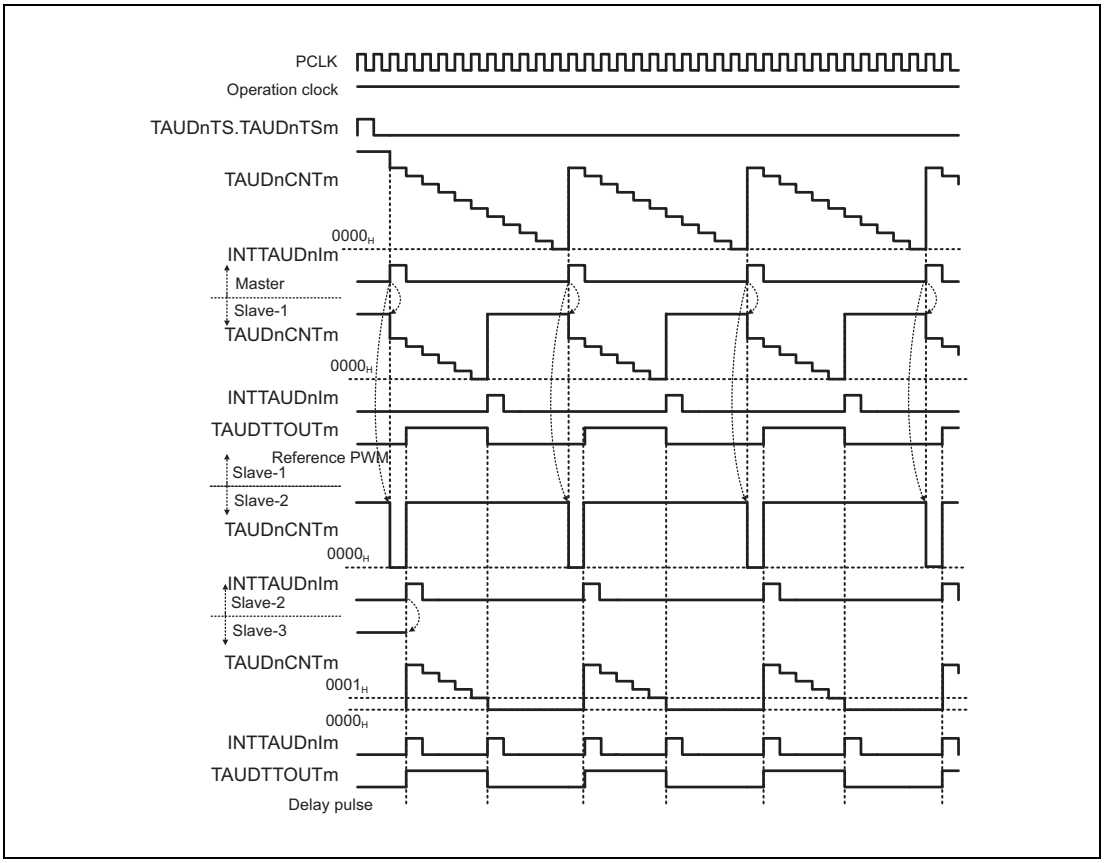


Figure 25.109 TAUDTTOUTm (Slave 1) = TAUDTTOUTm (Slave 3)

- If TAUDnCDRm (slave 2) = 0000_H, the counter of slave channel 3 starts counting one count clock later than the counter of slave channel 1. The reference pulse and the delay pulse are output with a delay of one clock count.

25.15.5 Offset Trigger Output Function

25.15.5.1 Overview

Summary

This function generates a PWM output using a master channel and a slave channel, enabling the pulse width (duration) of the TAUDTTOUTm to be set. The pulse cycle is set by detecting a valid input edge of master channel. The pulse width is specified on the slave channel.

Prerequisites

- Two channels
- The operating mode for the master channel should be set to capture mode. (See **Table 25.180, Contents of the TAUDnCMORm Register for the Master Channel of the Offset Trigger Output Function**.)
- The operating mode for slave channels should be set to one-count mode. (See **Table 25.183, Contents of the TAUDnCMORm Register for the Slave Channel of the Offset Trigger Output Function**.)
- The output mode for slave channels should be set to synchronous channel output mode 1. (See **Section 25.7, Channel Output Modes**.)
- TAUDTTOUTm is not used with the master channel of this function.

Functional description

The counter can be started by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This makes TAUDnTE.TAUDnTEm = 1, enabling the counter to count up. The master channel counter (TAUDnCNTm) starts to count up from 0000_H.

- Master channel:
  When a valid TAUDTTINm input edge is detected, the current value of the counter (TAUDnCNTm) is loaded into the data register of master channel (TAUDnCDRm). INTTAUDnIm is generated and the counter restarts to count up from 0000_H.
- Slave channel:
  If INTTAUDnIm is generated on the master channel, the TAUDTTOUTm (slave) signal is set and the counter of the slave channel is triggered. The current value of TAUDnCDRm (slave) is loaded into TAUDnCNTm (slave) and the counter starts to count down from this value.
  When the counter reaches 0000_H (duty time has elapsed), INTTAUDnIm is generated and TAUDTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUDnIm of the master channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

25.15.5.2 Equations

Pulse width = (TAUDnCDRm (slave)) × count clock cycle

Duty cycle [%] = [TAUDnCDRm (slave)/(TAUDnCDRm (master) + 1)] × 100

– Duty cycle = 0%
  TAUDnCDRm (slave) = 0000_H
– Duty cycle = 100%
  TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

25.15.5.3 Block Diagram and General Timing Diagram

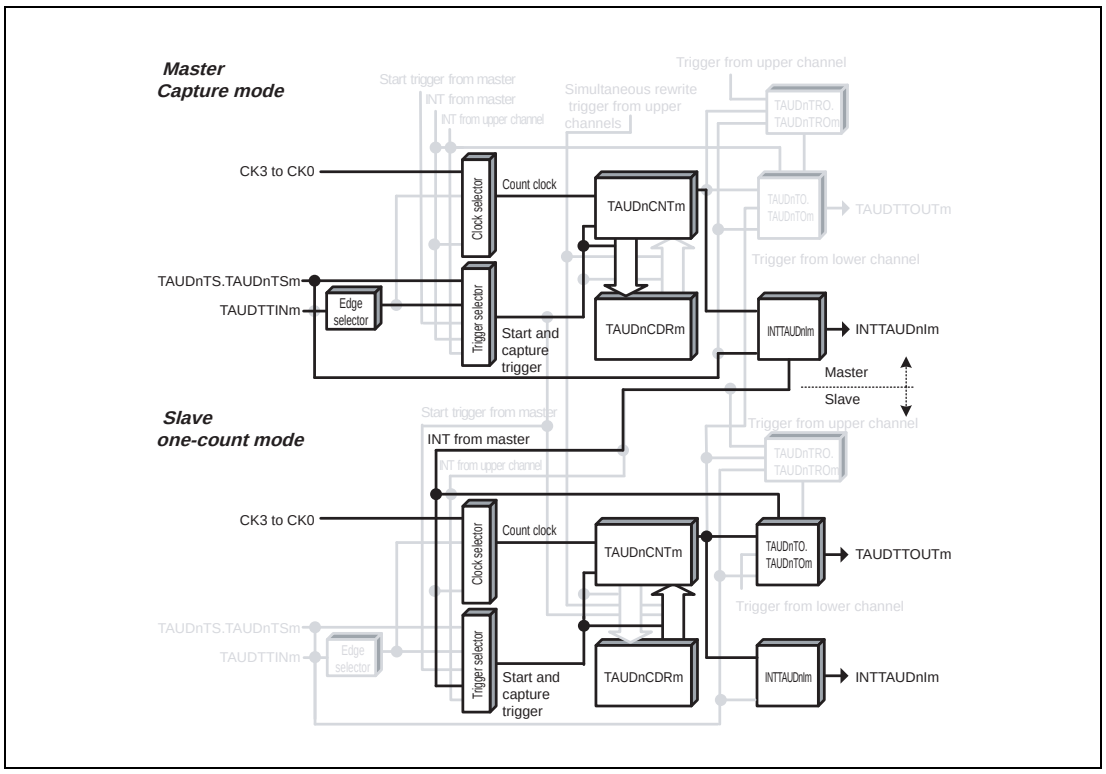


Figure 25.110 Block Diagram of Offset Trigger Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

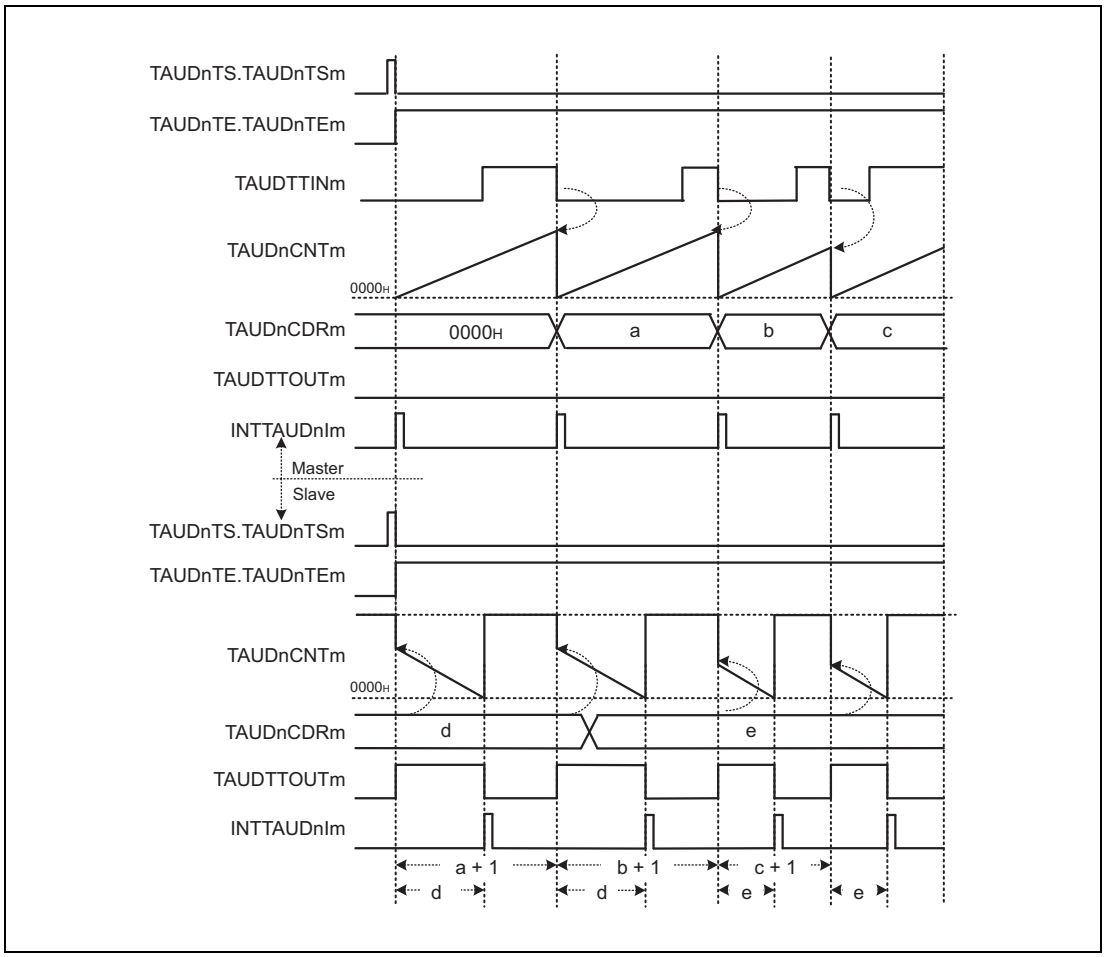


Figure 25.111 General Timing Diagram of Offset Trigger Output Function

NOTE

TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

25.15.5.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.180 Contents of the TAUDnCMORm Register for the Master Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	001: Valid TAUDTTINm input edge signal is used as the start trigger
7, 6	TAUDnCOS[1:0]	11: Capture register is updated upon detection of a valid TAUDTTINm input edge or when a counter overflow occurs: – Detection of valid TAUDTTINm input edge: The counter value is written into TAUDnCDRm. – Occurrence of overflow: $FFFF_H$ is written into TAUDnCDRm. A valid TAUDTTINm input edge to be detected next is ignored. TAUDnCSRm.TAUDnOVF is set when a counter overflow occurs, and cleared by setting TAUDnCSCm.TAUDnCLOV = 1.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0010: Capture mode
0	TAUDnMD0	1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the master channel

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

Table 25.181 Contents of the TAUDnCMURm Register for the Master Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Detection of falling edge 01: Detection of rising edge 10: Detection of rising and falling edges 11: Setting prohibited

(3) Channel output mode for the master channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous rewrite for the master channel

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 25.182 Simultaneous Rewrite Settings for the Master Channel of Offset Trigger Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.15.5.5 Register Settings for Slave Channels

(1) TAUDnCMORm for slave channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.183 Contents of the TAUDnCMORm Register for the Slave Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for slave channels

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

Table 25.184 Contents of the TAUDnCMURm Register for the Slave Channel of the Offset Trigger Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channels

Table 25.185 Control Bit Settings in Synchronous Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for slave channels

Simultaneous rewrite registers (TAUDnRDE, TAUDnRDS, TAUDnRDM, and TAUDnRDC) cannot be used with this function. Therefore, these registers should be set to 0.

Table 25.186 Simultaneous Rewrite Settings for Slave Channels of Offset Trigger Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	0: Disables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	When simultaneous rewrite is disabled (TAUDnRDE.TAUDnRDEm = 0), set these bits to 0
TAUDnRDM.TAUDnRDMm	
TAUDnRDC.TAUDnRDCm	

25.15.5.6 Operating Procedure for Offset Trigger Output Function

Table 25.187 Operating Procedure for Offset Trigger Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.5.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and channel output mode as described in **Section 25.15.5.5, Register Settings for Slave Channels**. The TAUDnCDRm register of master channel functions as a capture register. Set the value of TAUDnCDRm register of slave channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start: • TAUDnCNTm (master) counts up. • TAUDnCDRm value is loaded into TAUDnCNTm (slave) to perform counting down. INTTAUDnIm is generated on the master channel and TAUDTTOUTm (slave) is set.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCSCm.TAUDnCLOV can be set to 1. TAUDnCDRm of slave channel can be changed after the generation of INTTAUDnIm (master). TAUDnCNT.TAUDnCNTm and TAUDnCSRm can be read at any time.	When TAUDnCNTm of the slave = 0000_H: • INTTAUDnIm (slave) is generated. • TAUDTTOUTm (slave) is reset, and the counter of slave channel stops. When TAUDTTINm input edge is detected on the master channel: • INTTAUDnIm (master) is generated. • TAUDnCNTm (master) is reset to 0000_H and then continues count operation subsequently. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave) to perform counting down. • TAUDTTOUTm (slave) is set.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

25.15.5.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to this diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

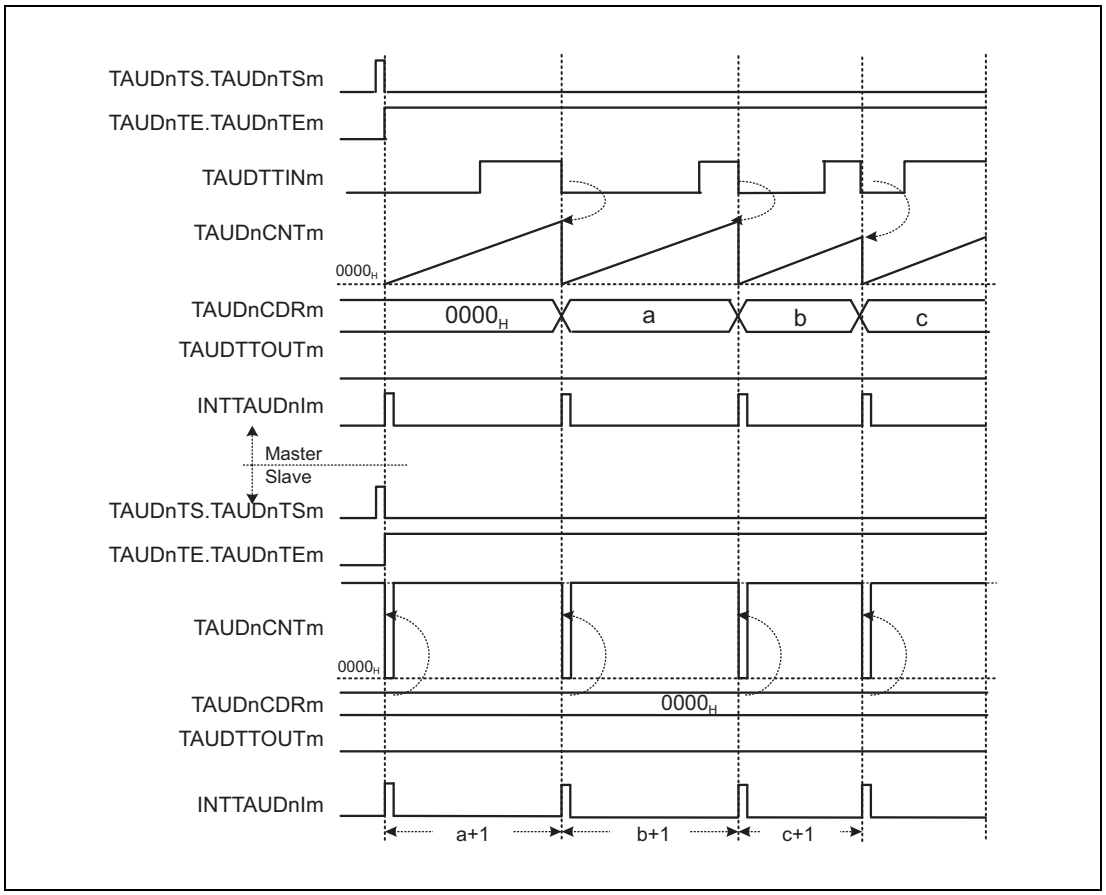


Figure 25.112 TAUDnCDRm (Slave) = 0000_H

- When TAUDnCDRm (slave) = 0000_H, 0000_H is written to TAUDnCNTm every time the master channel generates an interrupt (INTTAUDnIm), and TAUDnCNTm cannot start to count. The TAUDTTOUTm remains inactive.
- TAUDnCNTm (slave) generates an interrupt every time the value of TAUDnCDRm is reloaded. The slave and the master channels generate interrupts in the same cycle.

(2) Duty cycle = 100%

The following settings apply to this diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

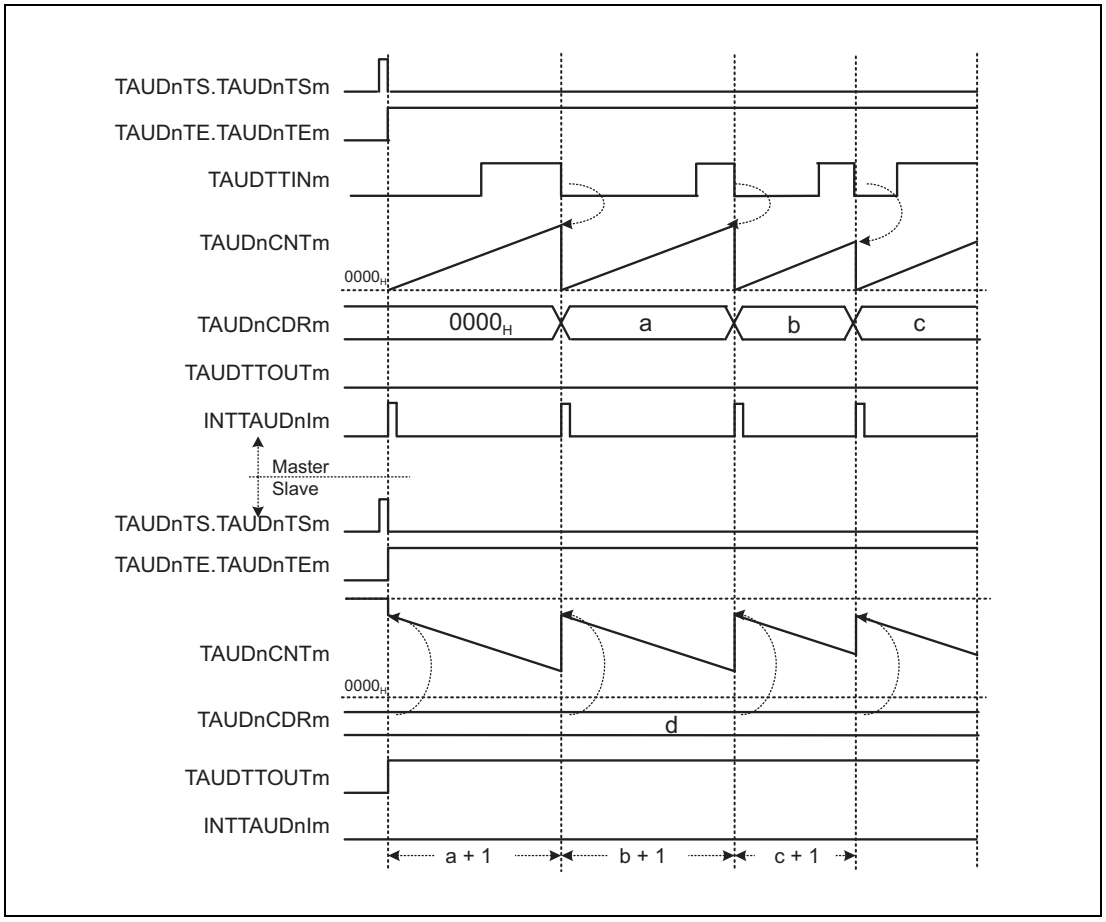


Figure 25.113 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1

- If the value TAUDnCDRm (slave) is higher than the interval of valid input edges, the counter of the slave channel cannot reach 0000_H and cannot generate interrupts. The TAUDTTOUTm remains in an active state.

25.15.6 A/D Conversion Trigger Output Function Type 1

25.15.6.1 Overview

Summary

This function is identical to **Section 25.15.1, PWM Output Function**, except that TAUDTTOUTm is not output.

This function is enabled by setting the channel output mode for the slave to independent channel output mode controlled by software.

25.15.6.2 Block Diagram and General Timing Diagram

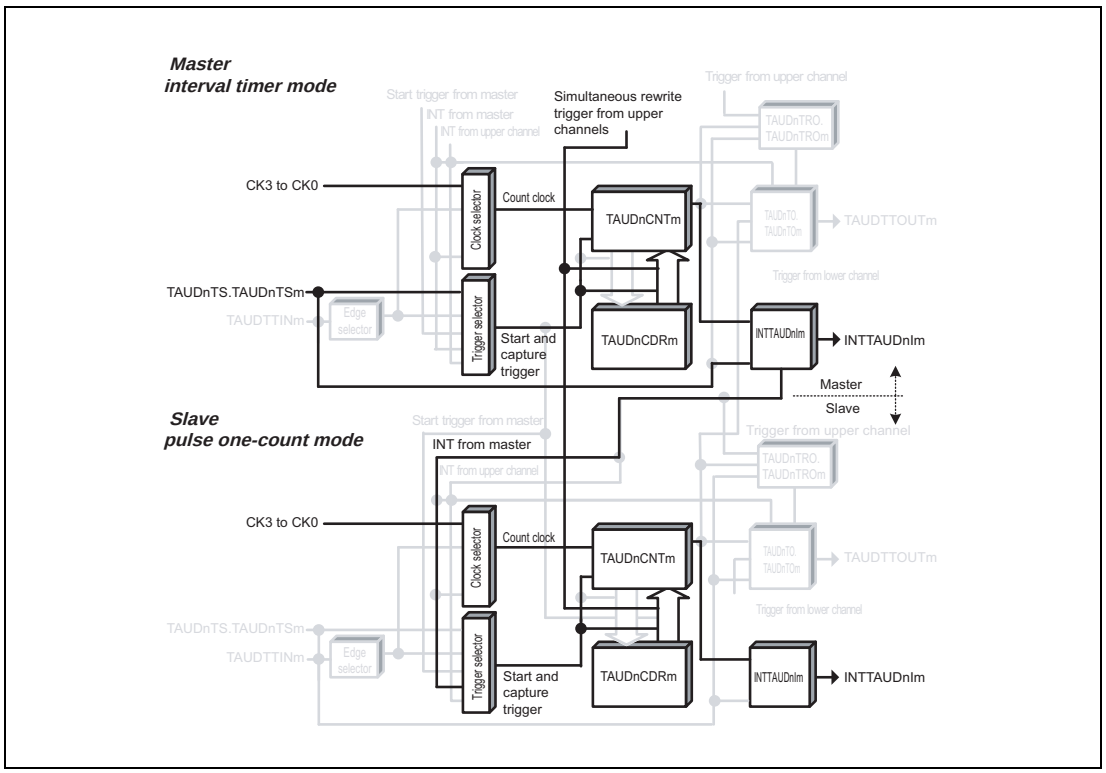


Figure 25.114 Block Diagram of A/D Conversion Trigger Output Function Type 1

The following settings apply to the general timing diagram.

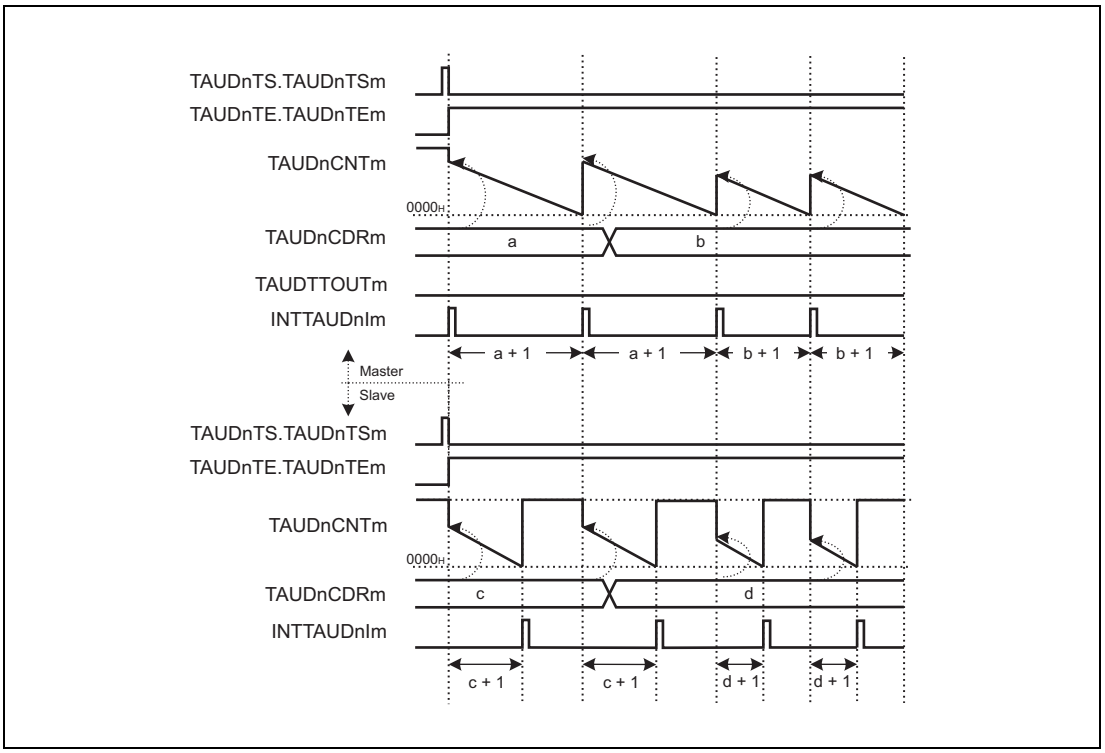


Figure 25.115 General Timing Diagram of A/D Conversion Trigger Output Function Type 1

25.15.7 Triangle PWM Output Function

25.15.7.1 Overview

Summary

This function generates multiple triangle PWM outputs by using a master and one or more slave channels. It enables the pulse cycle (frequency) and the duty cycle of TAUDTTOUTm to be set using the master and slave channels respectively.

The master channel generates a carrier cycle. The first cycle of the master channel controls the down status and the second cycle controls the up status of the slave counter.

Prerequisites

- Two channels
- The operating mode for the master channels should be set to interval timer mode. (See **Table 25.188, Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function**.)
- The operating mode for slave channels should be set to count-up/-down mode. (See **Table 25.192, Contents of the TAUDnCMORm Register for the Slave Channel of the Triangle PWM Output Function**.)
- The channel output mode for the master channel should be set to independent channel output mode 1. (See **Section 25.7, Channel Output Modes**.)
- The channel output mode for slave channels should be set to synchronous channel output mode 2. (See **Section 25.7, Channel Output Modes**.)
- The following settings allow the TAUDTTOUTm signal to be at high level during the down status of a carrier cycle.
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 0, TAUDnTO.TAUDnTOm should be set to 1 while TAUDnTOE.TAUDnTOEm is set to 0 (recommended setting).
  - If TAUDnCMORm.TAUDnMD0 (master) bit is set to 1, TAUDnTO.TAUDnTOm should be set to 0 while TAUDnTOE.TAUDnTOEm is set to 0.

Functional description

The counters are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1 for every channel. This in turn sets TAUDnTE.TAUDnTEm, enabling count operation. The current values of TAUDnCDRm (master and slave) are loaded into TAUDnCNTm (master and slave) and the counters start counting down from these values. When the TAUDnCMORm.TAUDnMD0 bit of master channel is set to 1, an interrupt is generated and TAUDTTOUTm signal of master toggles.

- Master channel:
  When the counter of master channel reaches 0000_H (pulse cycle time has elapsed), INTTAUDnIm is generated and the TAUDTTOUTm signal toggles. TAUDnCNTm then reloads the TAUDnCDRm value and counts down.

- Slave channel:
  The INTTAUDnIm of the master channel triggers the counter of the slave channel:
  - If the slave counter is counting down, the count direction changes.
  - If the slave counter is counting up, the TAUDnCDRm value is reloaded and the counter starts to count down.

  When the counter of the slave channel reaches 0001_H while counting up or down, INTTAUDnIm is generated and the TAUDTTOUTm (slave) signal is set/reset.
  The counter continues count-up/-down and waits for the next INTTAUDnIm of the master channel.
  Setting TAUDnTOL.TAUDnTOLm allows TAUDTTOUTm signal switching between normal phase and reverse phase during operation.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm = 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values.

Conditions

This function enables simultaneous rewrite. See **Section 25.6, Simultaneous Rewrite**.

25.15.7.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUDnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

Duty cycle 100 [%] =
[(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave))/(TAUDnCDRm (master) + 1)] × 100

- Duty cycle = [%]
  TAUDnCDRm (slave) = 0000_H
- Duty cycle = 0%
  TAUDnCDRm (slave) ≥ TAUDnCDRm (master) + 1

25.15.7.3 Block Diagram and General Timing Diagram

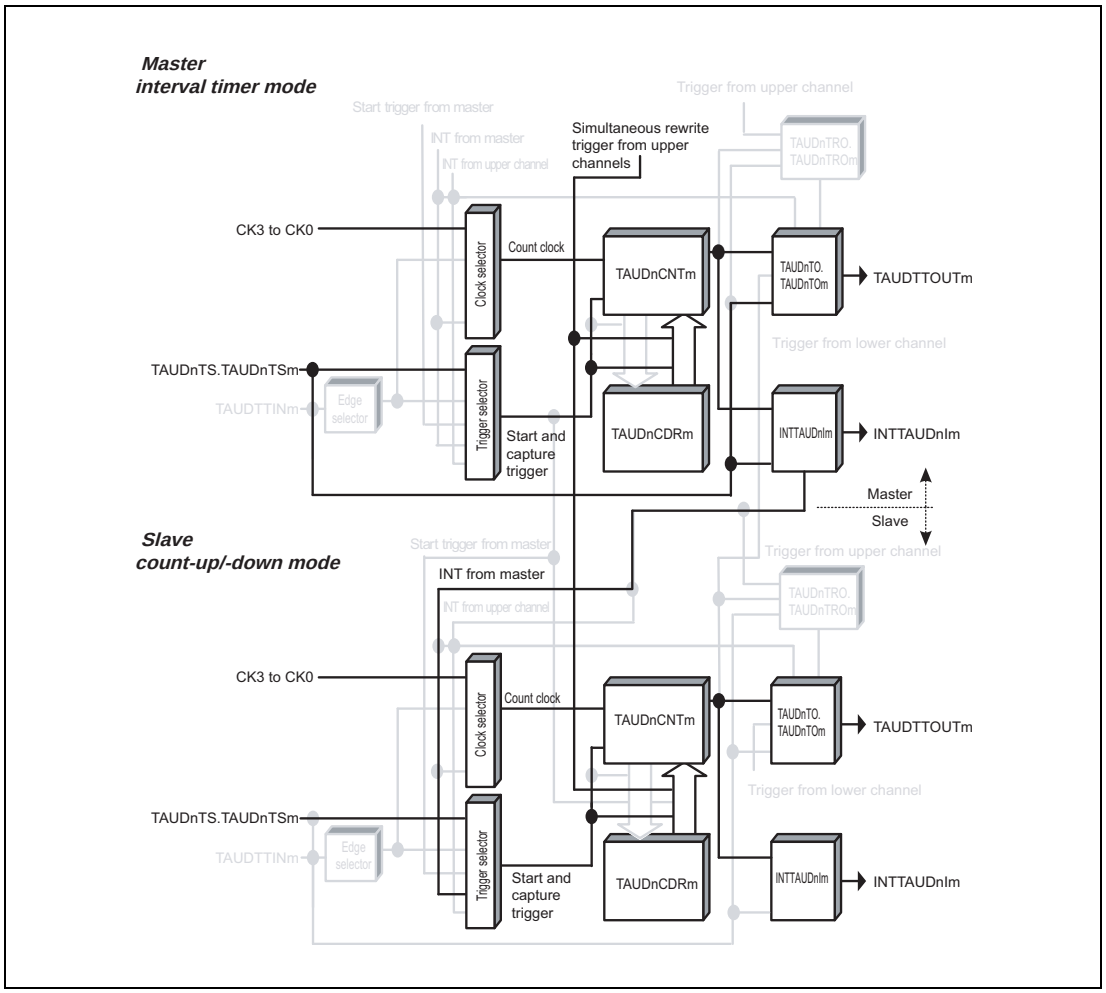


Figure 25.116 Block Diagram of Triangle PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUDnIm is generated at the beginning of operation.
    (TAUDnCMORm.TAUDnMD0 = 1)

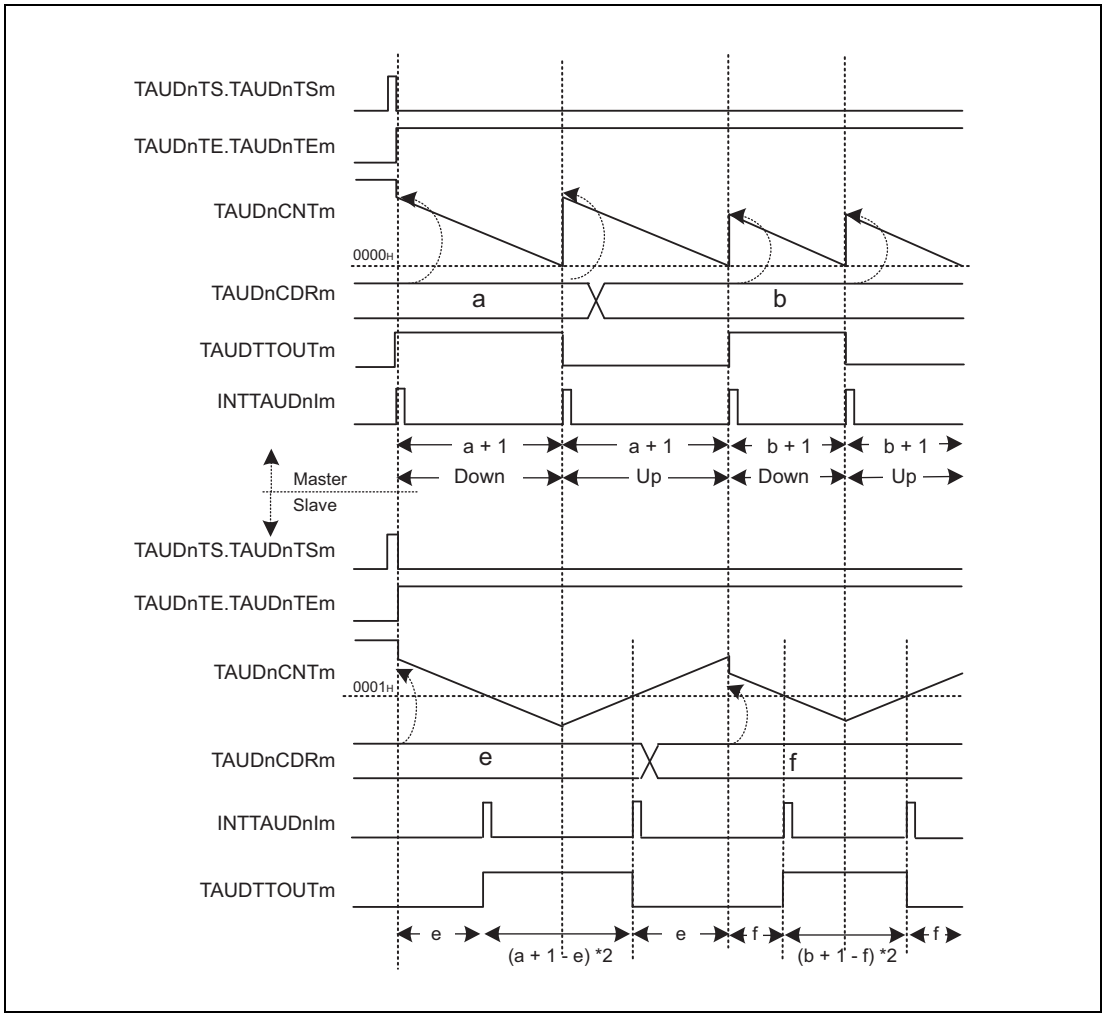


Figure 25.117 General Timing Diagram of Triangle PWM Output Function

25.15.7.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.188 Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated to toggle TAUDTTOUTm at the beginning of an operation. 1: INTTAUDnIm is generated and TAUDTTOUTm is toggled at the beginning of operation.

(2) TAUDnCMURm for the master channel

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

Table 25.189 Contents of the TAUDnCMURm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the master channel

Table 25.190 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.191 Simultaneous Rewrite Settings for the Master Channel of Triangle PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

25.15.7.5 Register Settings for Slave Channels

(1) TAUDnCMORm for slave channels

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.192 Contents of the TAUDnCMORm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm for slave channels

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

Table 25.193 Contents of the TAUDnCMURm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channels

Table 25.194 Control Bit Settings in Synchronous Channel Output Mode 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for slave channels

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.195 Simultaneous Rewrite Settings for Slave Channels of Triangle PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when the master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.7.6 Operating Procedure for Triangle PWM Output Function

Table 25.196 Operating Procedure for Triangle PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.7.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.7.5, Register Settings for Slave Channels**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm (master) is generated on the master channel if TAUDnCMORm.TAUDnMD0 is set to 1.
During Operation	TAUDnCDRm can be changed at any time. TAUDnTOL.TAUDnTOLm can be changed. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master and slave channels is loaded into TAUDnCNTm to count down. When the counter of master channel reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDTTOUTm (master) is toggled. • TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave) or counting is started in opposite direction. When TAUDnCNTm of slave channel reaches 0001_H: • INTTAUDnIm (slave) is generated. • TAUDTTOUTm (slave) is set in the count-down status or reset in count-up status.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

25.15.7.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnCDRm = a = 5_H
- Slave channel:
  - TAUDnCDRm = 6_H

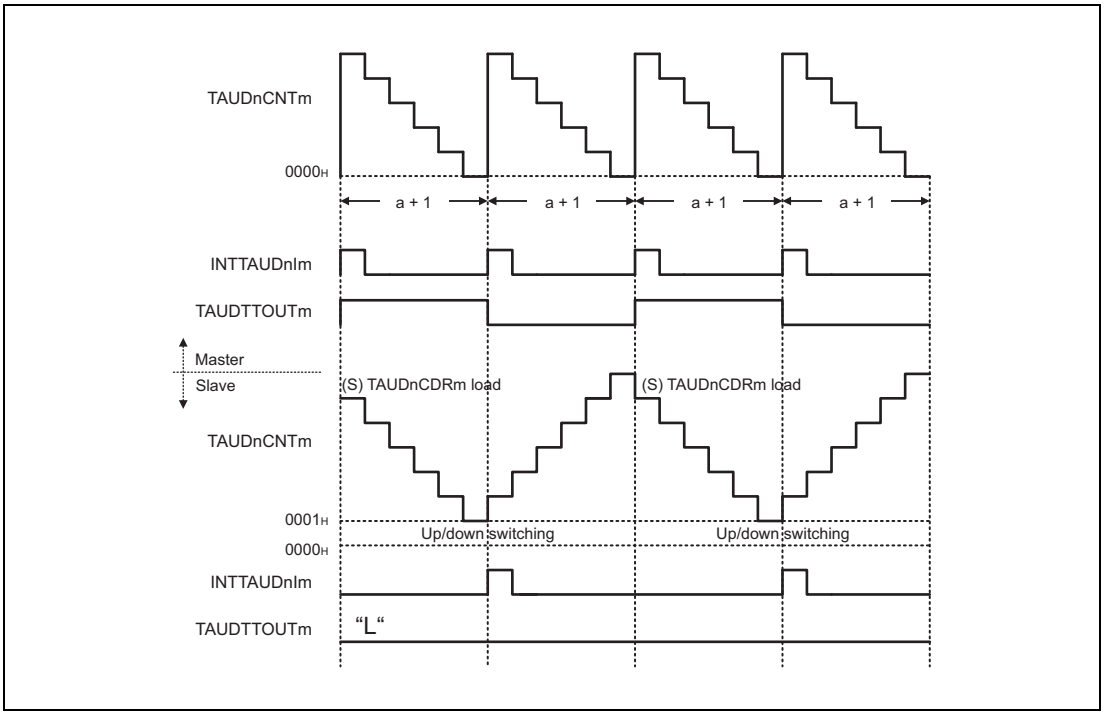


Figure 25.118 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1

- If TAUDnCDRm (slave) value ≥ TAUDnCDRm (master) value +1, INTTAUDnIm of the slave channel is not generated while counting down. TAUDTTOUTm remains low because there is no set signal to be detected.

(2) Duty cycle = 100%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnCDRm = a = 5_H
- Slave channel:
  - TAUDnCDRm = 0_H

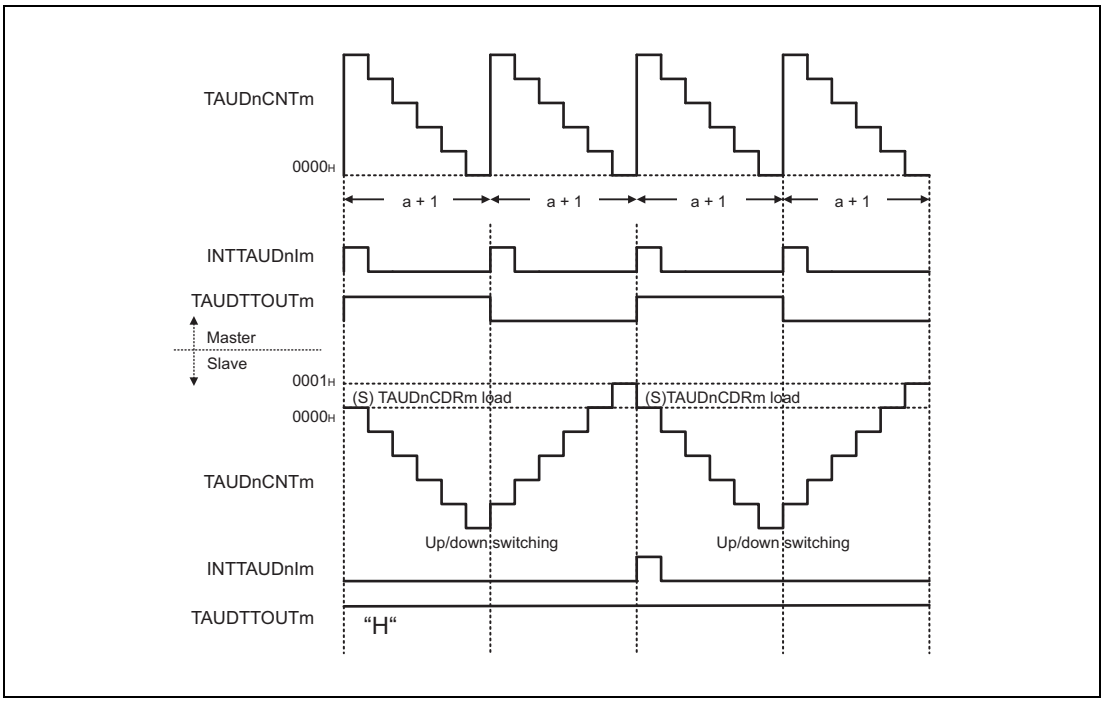


Figure 25.119 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm (slave) = 0000_H, INTTAUDnIm of the slave channel is not generated while counting up. TAUDTTOUTm remains high because there is no reset signal to be detected.

25.15.8 Triangle PWM Output Function with Dead Time

25.15.8.1 Overview

Summary

This function generates multiple triangle PWM outputs with a predefined dead time added by using a master and two or more slave channels. The resulting PWM signals with dead time are output via TAUDTTOUTm of the slave channels 2 and 3, enabling the pulse cycle (frequency) and the duty cycle of TAUDTTOUTm to be set using the master and slave channels.

Carrier cycles are generated on the master channel. The first pulse controls the down status of the slave counter and the second one controls the up status.

An interrupt on slave 2 causes TAUDTTOUTm of slave channels to be set/reset. Depending on the settings of TAUDnTDL.TAUDnTDLm, delay time is added to positive or negative logic side of the signal (i.e., whether TAUDTTOUTm is set/reset immediately or after dead time has elapsed). The duration of the dead time is specified by slave channel 3.

Prerequisites

- Three channels. For slave channels 2 and 3, select even channel CH (a) and odd channel CH (a + 1).
- The operating mode for the master channel should be set to interval timer mode. (See **Table 25.198, Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time**)
- Slave channel 1 is not used for this function. This ensures that slave channel 2 is an even channel (a), and slave channel 3 is an odd channel (a + 1). Slave channel 1 can be used as a separate timer (independent function).
- The operating mode for slave channel 2 should be set to count-up/-down mode (See **Table 25.202, Contents of the TAUDnCMORm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time**). Slave channel 2 should be an even channel.
- The operating mode for slave channel 3 should be set to one-count mode (See **Table 25.206, Contents of the TAUDnCMORm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time**). Slave channel 3 should be an odd channel.
- The channel output mode for the master channel should be set to independent channel output mode 1. (See **Section 25.7, Channel Output Modes**)
- The output mode for slave channels 2 and 3 should be set to synchronous channel output mode 2 with dead time output. (See **Section 25.7, Channel Output Modes**)
- The following settings make a TAUDTTOUTm signal at high level during the down status of the carrier cycle:

– If TAUDnCMORm.TAUDnMD0 (master) bit is set to 0, TAUDnTO.TAUDnTOm should be set to 1 while TAUDnTOE.TAUDnTOEm is set to 0 (recommended setting).
– If TAUDnCMORm.TAUDnMD0 (master) bit is set to 1, TAUDnTO.TAUDnTOm should be set to 0 while TAUDnTOE.TAUDnTOEm is set to 0.

NOTE

The triangle PWM output function with dead time does not use slave channel 1. Slave channel 1 can be used as a separate timer (independent function).

Functional description

The counter starts by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This makes TAUDnTE.TAUDnTEm = 1, enabling count operation. The current value of TAUDnCDRm is loaded into TAUDnCNTm and the counter starts to count down from the TAUDnCDRm value. If TAUDnCMORm.TAUDnMD0 bit of master channel is set to 1, an interrupt is generated and the master's TAUDTTOUTm signal is toggled.

- Master channel:
  When the counter of the master channel reaches 0000_H, an INTTAUDnIm is generated and the TAUDTTOUTm signal is toggled. The TAUDnCDRm value is reloaded to continue counting down.
- Slave channel 2:
  If INTTAUDnIm is generated on the master channel, the counter of slave channel 2 is triggered.
  – If the slave counter is counting down, the counting direction changes.
  – If the slave counter is counting up, the TAUDnCDRm value is reloaded and the counter starts counting down.

  The counter continues to count down/up and waits for the next INTTAUDnIm of the master channel.
  When the counter value of slave channel 2 reaches 0001_H, INTTAUDnIm is generated
- Slave channel 3:
  If INTTAUDnIm is generated on slave channel 2, the counter of slave channel 3 is triggered. The current value of TAUDnCDRm (slave 3) is loaded into TAUDnCNTm (slave 3) and the counter starts to count down from the TAUDnCDRm value.
  When the counter reaches 0000_H, INTTAUDnIm occurs. The counter returns to $FFFF_H$ and waits for the next INTTAUDnIm of slave channel 2.

As described in **Table 25.197, Operation of TAUDTTOUTm upon Occurrence of an Interrupt on Slave Channel 2**, the set/reset timing (immediately after occurrence of an interrupt or after dead time has elapsed) depends on the TAUDnTDL.TAUDnTDLm setting of the corresponding channel.

The setting of TAUDnTOL.TAUDnTOLm also determines whether a high level signal (TAUDnTOL.TAUDnTOLm = 0) or a low level signal (TAUDnTOL.TAUDnTOLm = 1) is output from the corresponding channel.

The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values.

TAUDTTOUTm can be 100% output by setting the TAUDnCDRm value of slave channel 2 to 0000_H.

Conditions

This function enables simultaneous rewrite. See **Section 25.6, Simultaneous Rewrite**.

TAUDnTOL.TAUDnTOLm and TAUDnTDL.TAUDnTDLm should be set before start of count operation. Slave channels 2 and 3 should have the opposite settings of TAUDnTDL.TAUDnTDLm.

Table 25.197 Operation of TAUDTTOUTm upon Occurrence of an Interrupt on Slave Channel 2

TAUDnTDL. TAUDnTDLm	Count Direction of Slave Channel 2 upon Occurrence of Interrupt	TAUDTTOUTm Set/Reset Timing
0	Down	Set after elapse of dead time
	Up	Reset immediately after interrupt occurs
1	Down	Set immediately after interrupt occurs
	Up	Reset after elapse of dead time

25.15.8.2 Equations

Pulse cycle = (TAUDnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUDnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

PWM signal width (normal phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2))× 2 – (TAUDnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (reverse phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2)) × 2 + (TAUDnCDRm (slave 3) + 1)] × count clock cycle

25.15.8.3 Block Diagram and General Timing Diagram

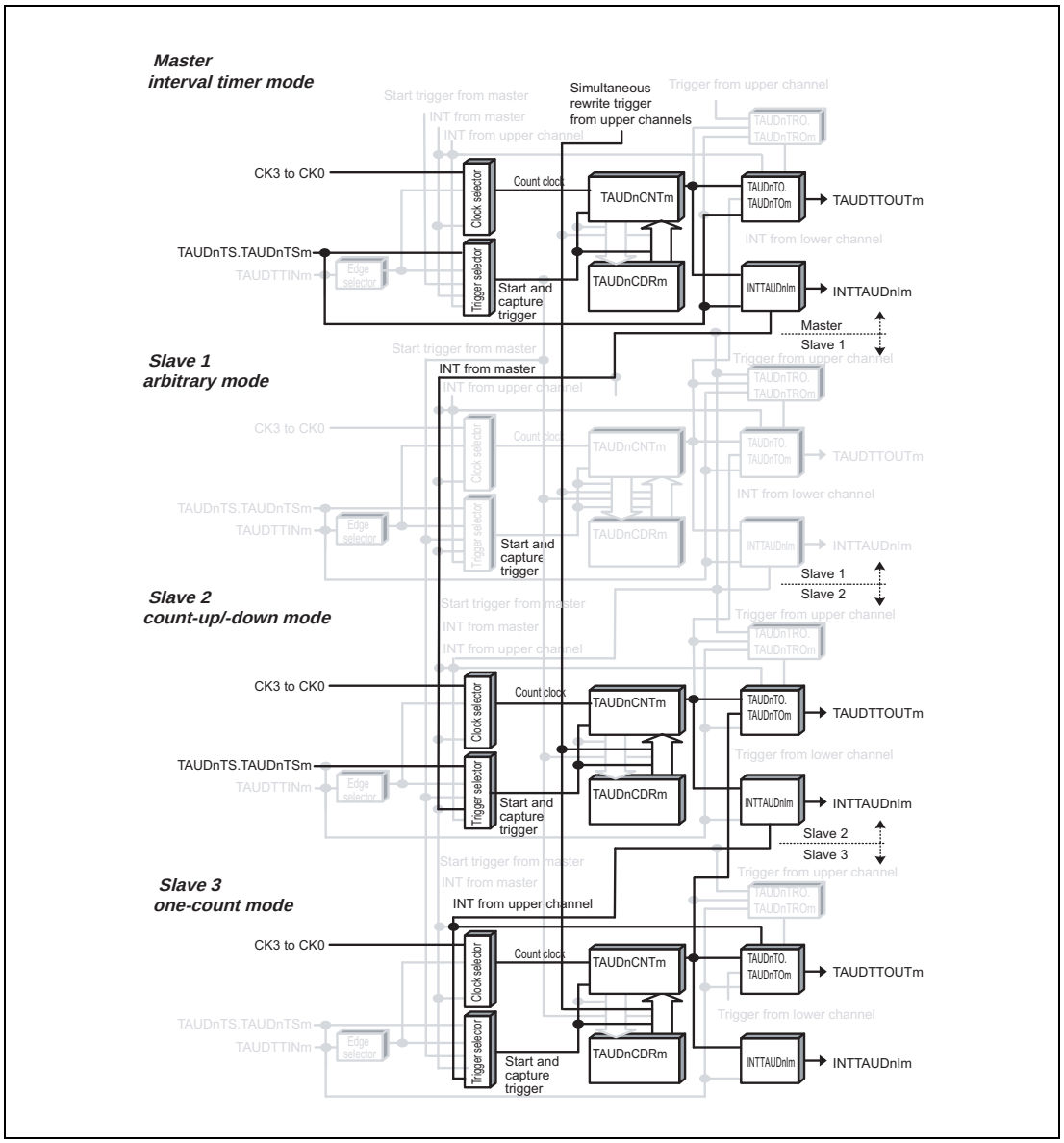


Figure 25.120 Block Diagram of Triangle PWM Output Function with Dead Time

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
- Slave channel 2:
  - INTTAUDnIm not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
  - TAUDnTDL.TAUDnTDLm = 0
  - Positive logic (TAUDnTOL.TAUDnTOLm = 0)

- Slave channel 3:
  - Enables start trigger detection during counting (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnTDL.TAUDnTDLm = 1
  - Positive logic (TAUDnTOL.TAUDnTOLm = 0)

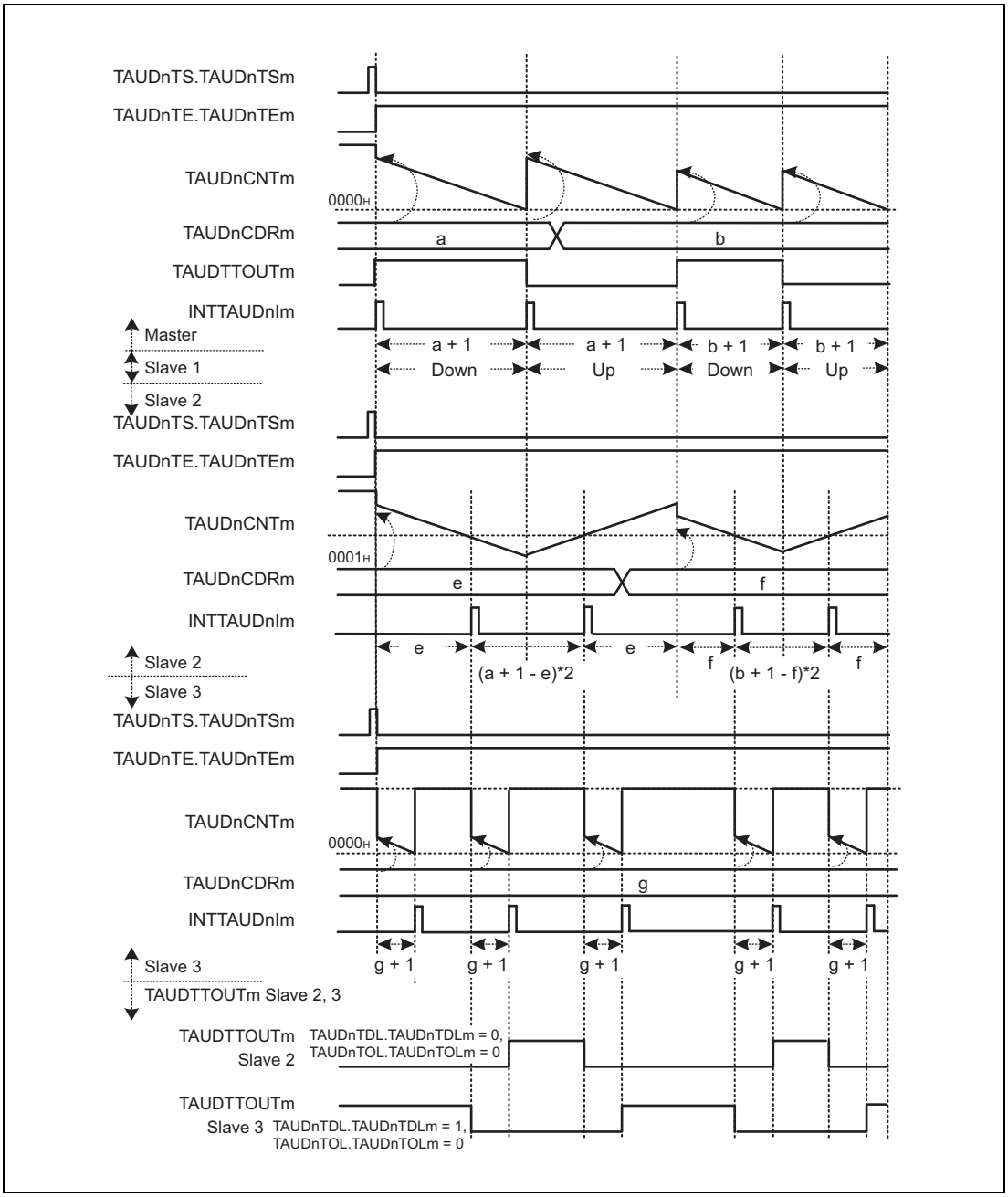


Figure 25.121 General Timing Diagram of Triangle PWM Output Function with Dead Time

25.15.8.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.198 Contents of the TAUDnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated and TAUDTTOUTm is not toggled at the beginning of operation. 1: INTTAUDnIm is generated and TAUDTTOUTm is toggled at the beginning of operation.

(2) TAUDnCMURm for the master channel

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

Table 25.199 Contents of the TAUDnCMURm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the master channel

Table 25.200 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.201 Simultaneous Rewrite Setting for the Master Channel of Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

25.15.8.5 Register Settings for Slave Channel 2

(1) TAUDnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.202 Contents of the TAUDnCMORm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm for slave channel 2

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

Table 25.203 Contents of the TAUDnCMURm Register for Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channel 2

Table 25.204 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from odd channels.

(4) Simultaneous rewrite for slave channel 2

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.205 Simultaneous Rewrite Settings for Slave Channel 2 of Triangle PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.8.6 Register Settings for Slave Channel 3

(1) TAUDnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.206 Contents of the TAUDnCMORm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	110: Dead time output signal of the TAUDTTOUTm generation unit
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for slave channel 3

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

Table 25.207 Contents of the TAUDnCMURm Register for Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channel 3

Table 25.208 Control Bit Settings in Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel operation
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set this bit to 0
TAUDnTRC.TAUDnTRCm	0: Disables the operation as a real-time output trigger channel
TAUDnTME.TAUDnTMEm	0: Disables modulation

CAUTION

Set TAUDnTDL.TAUDnTDLm exclusively from even channels.

(4) Simultaneous rewrite for slave channel 3

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.209 Simultaneous Rewrite Settings for Slave Channel 3 of Triangle PWM Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave cycle.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.8.7 Operating Procedure for Triangle PWM Output Function with Dead Time

Table 25.210 Operating Procedure for Triangle PWM Output Function with Dead Time

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.8.4, Register Settings for the Master Channel**. Slave channel 2: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.8.5, Register Settings for Slave Channel 2**. Slave channel 3: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.8.6, Register Settings for Slave Channel 3**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm (master) is generated on the master channel if TAUDnCMORm.TAUDnMD0 is set to 1.
During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel and slave channel 2 is loaded into TAUDnCNTm to perform counting down. When the counter of master channel reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCDRm value is reloaded into TAUDnCNTm (master) to continue count operation. • TAUDnCDRm value is reloaded into TAUDnCNTm (slave 2) or counting is started in opposite direction. When TAUDnCNTm of slave channel 2 reaches 0001_H: • IINTTAUDnIm (slave 2) is generated. • TAUDnCDRm value of slave channel 3 is loaded into TAUDnCNTm perform counting down. When TAUDnCNTm of slave channel 3 reaches 0000_H: • INTTAUDnIm is generated.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

25.15.8.8 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram in **Figure 25.122**.

- Slave channel 2:
  - Positive logic (TAUDnTDL.TAUDnTDLm = 0)
- Slave channel 3:
  - Negative logic (TAUDnTDL.TAUDnTDLm = 1)

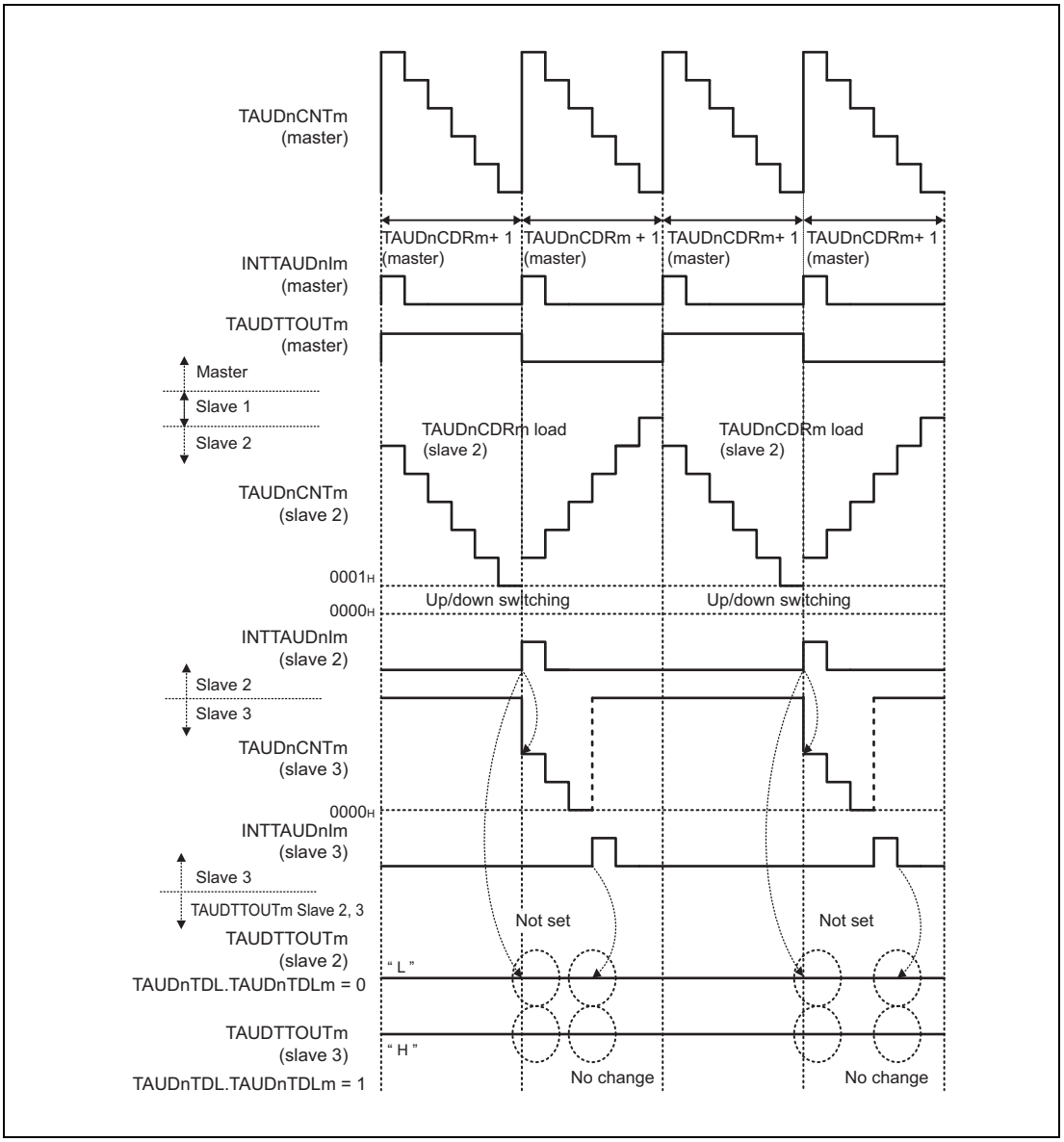


Figure 25.122 TAUDnCDRm (Slave 2) ≥ TAUDnCDRm (Master) + 1

- If TAUDnCDRm (slave 2) is greater than TAUDnCDRm (master), the counter of slave channel does not reach 0000_H while counting down. Therefore, TAUDTTOUTm signal is not set/reset and remains initial. This signal becomes a reset signal because an interrupt occurs on slave channel 2 during count-up operation.

(2) Duty cycle = 100%

The following settings apply to the general timing diagram in **Figure 25.123**.

- Slave channel 2:
  - Positive logic (TAUDnTDL.TAUDnTDLm = 0)
- Slave channel 3:
  - Negative logic (TAUDnTDL.TAUDnTDLm = 1)

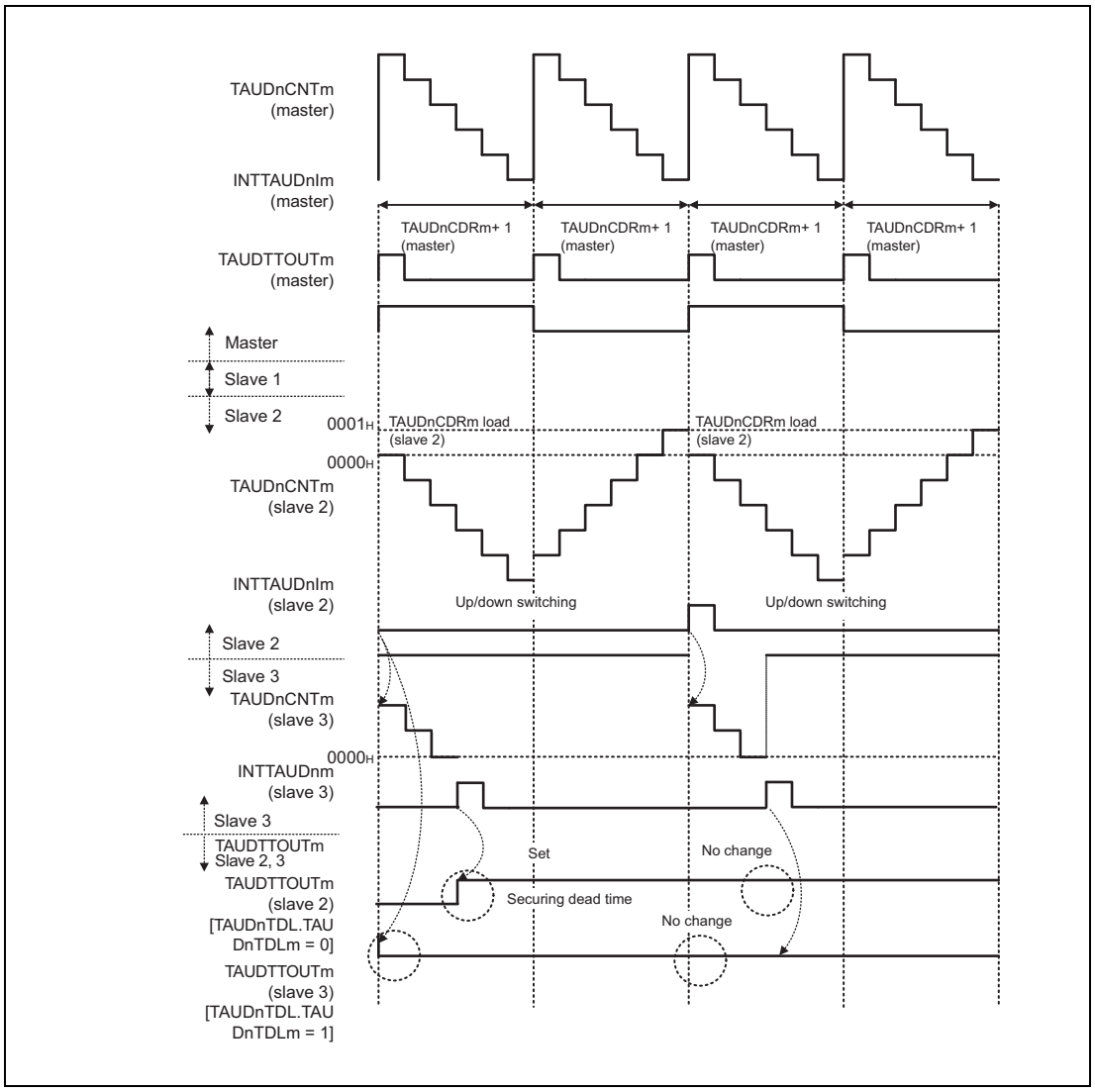


Figure 25.123 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm (slave 2) = 0000_H, the slave channel counter does not reach 0001_H while counting up. Therefore, no INTTAUDnIm is generated during count-up operation.
  - The set conditions for a channel with TAUDnTDL.TAUDnTDLm = 0 are met after elapse of dead time. TAUDTTOUTm is left in a newly set state even if a set/reset is made because no reset conditions are satisfied on such a channel.
  - Slave channel 3 in the above diagram is set when the counter starts. However, TAUDTTOUTm is left in an initial state on the slave channel with TAUDnTDL.TAUDnTDLm = 1 because no reset conditions are satisfied on that channel.

25.15.9 A/D Conversion Trigger Output Function Type 2

25.15.9.1 Overview

Summary

This function is identical to **Section 25.15.7, Triangle PWM Output Function**, except that TAUDTTOUTm is not output.

This function is enabled by setting channel output mode for the slave to independent channel output mode controlled by software.

25.15.9.2 Block Diagram and General Timing Diagram

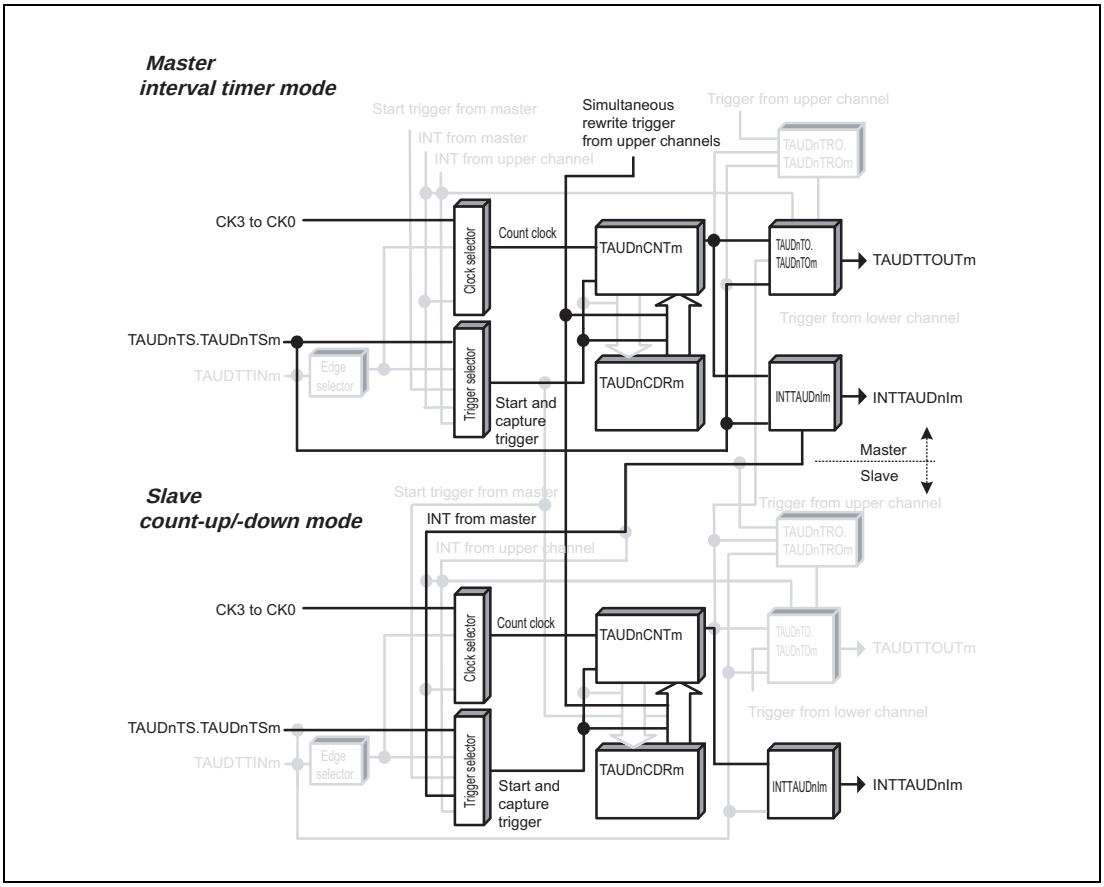


Figure 25.124 Block Diagram of A/D Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

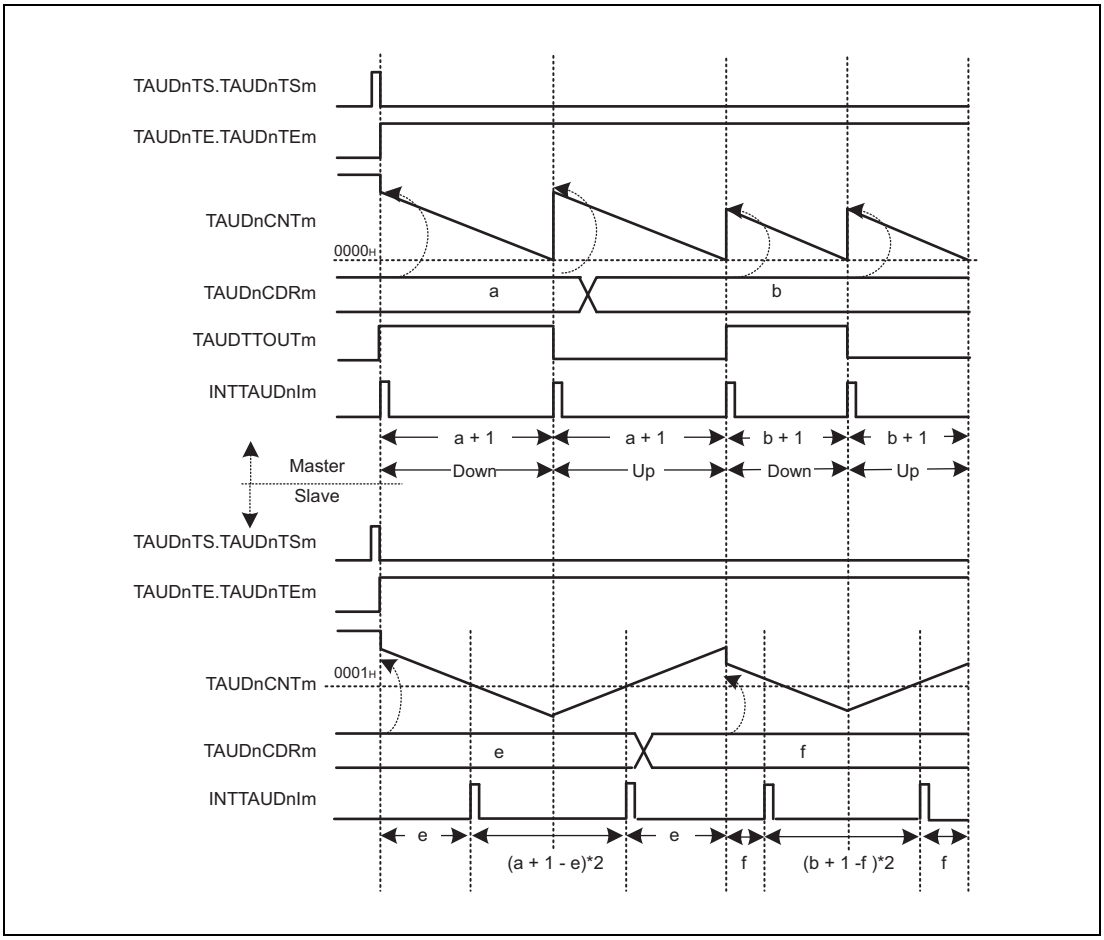


Figure 25.125 General Timing Diagram of A/D Conversion Trigger Output Function Type 2

25.15.10 Interrupt Request Signals Culling Function

25.15.10.1 Overview

Summary

This function divides the number of interrupts of the master channel by a specified value using a slave channel.

The interrupt request signals culling function is a sub function of the following functions:

- PWM Output Function (See **Section 25.15.1, PWM Output Function**)
- Triangle PWM Output Function (See **Section 25.15.7, Triangle PWM Output Function**
- Triangle PWM Output Function with Dead Time
  (See **Section 25.15.8, Triangle PWM Output Function with Dead Time**)

Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode. (See **Table 25.211, Contents of the TAUDnCMORm Register for the Master Channel of the Interrupt Request Signals Culling Function**)
- The operation mode of the slave channel must be set to Event Count Mode. (See **Table 25.214, Contents of the TAUDnCMORm Register for the Slave Channel of the Interrupt Request Signals Culling Function**)
- This function does not use TAUDTTOUTm.

Functional description

The counters (master and slave) are enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1 for both channels. This in turn sets TAUDnTE.TAUDnTEm, enabling count operation. The current value of the data register of the master channel and slave channel (TAUDnCDRm) are written to the counter (TAUDnCNTm).

- Master channel:
  When the counter of the master channel reaches 0000_H, INTTAUDnIm is generated and TAUDnCDRm value is reloaded to TAUDnCNTm.
- Slave channel:
  Every time the master channel generates an INTTAUDnIm, the counter of the slave channel decrements by one. When the counter reaches 0000_H, it awaits the next interrupt from the master channel. This causes TAUDnCNTm (slave) to reload the value of TAUDnCDRm, and an INTTAUDnIm is generated.

Forced restart is not possible for this function. The counter can be stopped by setting TAUDnTT.TAUDnTTm to 1 for the master and slave channel, which in turn sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm of master and slave channel stops but retains its value.

Conditions

This function enables simultaneous rewrite. See **Section 25.6, Simultaneous Rewrite**.

25.15.10.2 Equations

Interrupt division operator = TAUDnCDRm (slave channel)

– One INTTAUDnIm is generated for the INTTAUDnIm count of the master channel defined by TAUDnCDRm (slave channel) + 1.

25.15.10.3 Block Diagram and General Timing Diagram

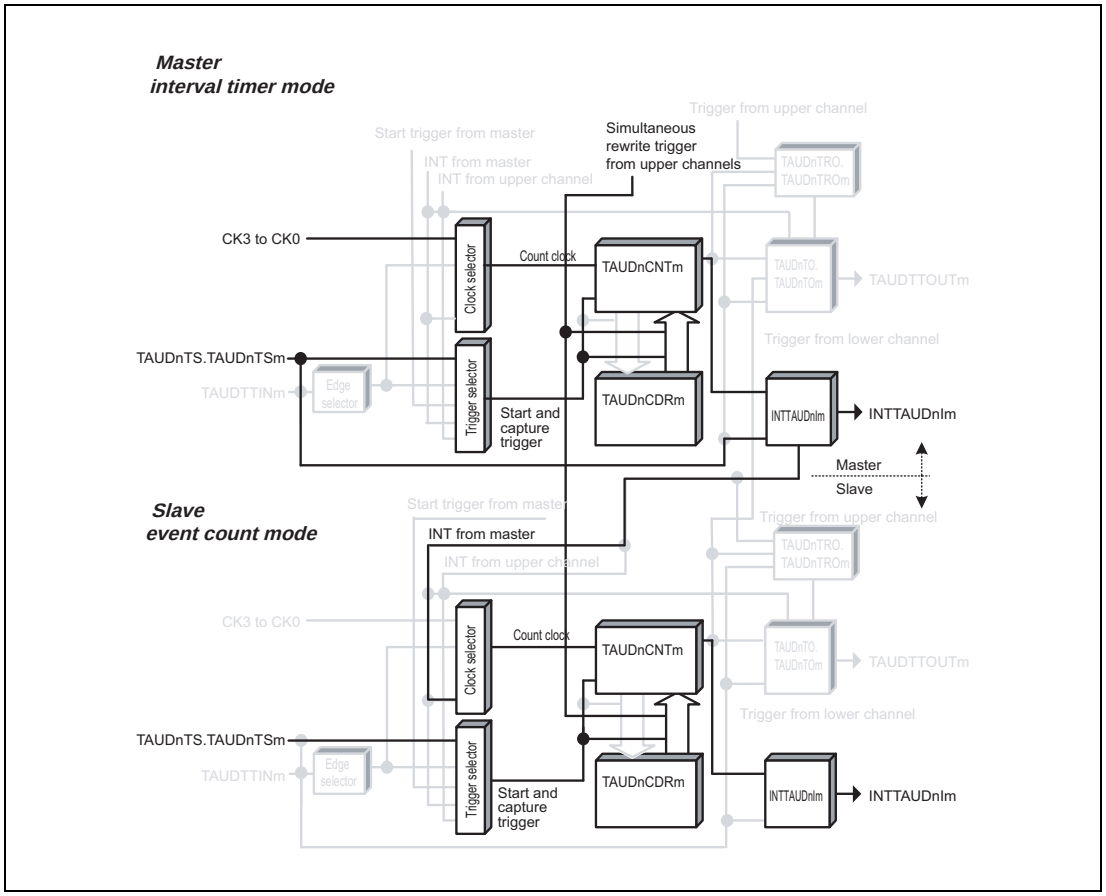


Figure 25.126 Block Diagram of Interrupt Request Signals Culling Function

The following settings apply to the general timing diagram.

Master channel:

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

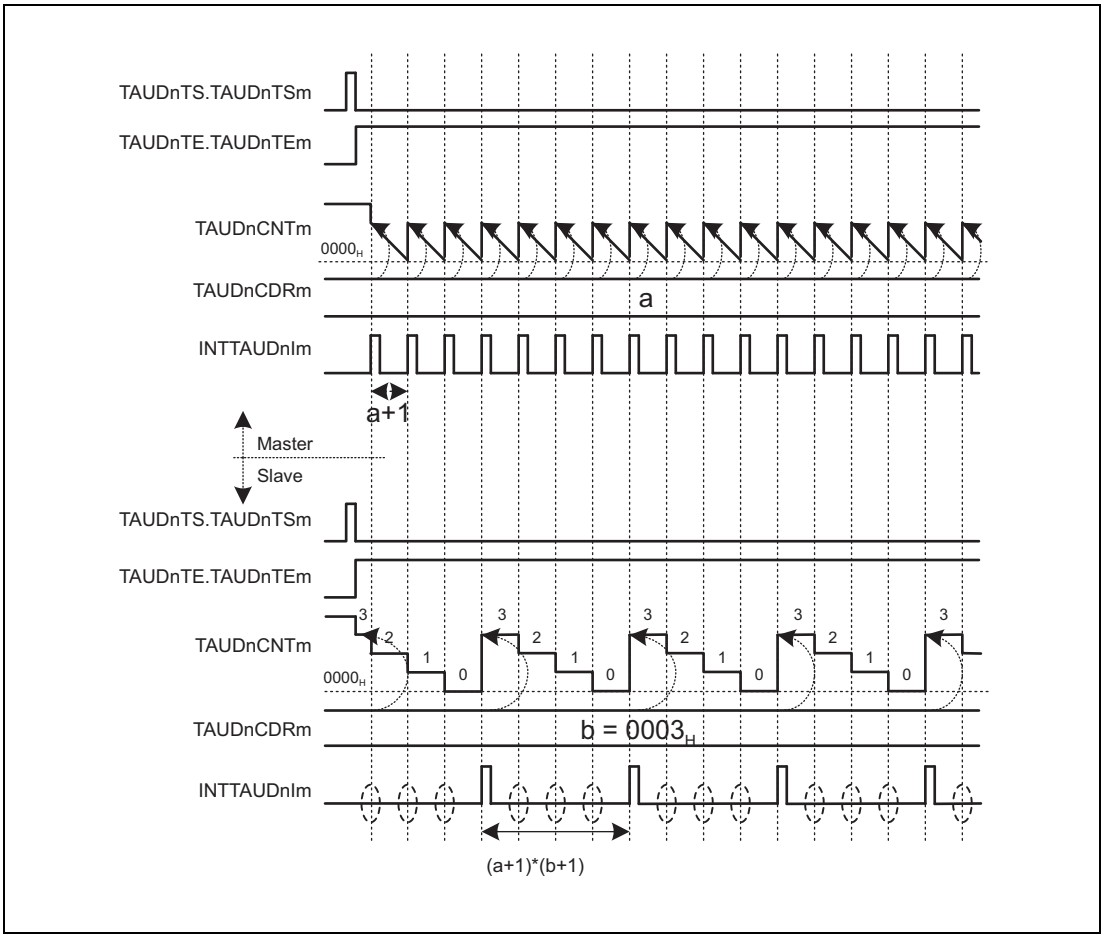


Figure 25.127 General Timing Diagram of Interrupt Request Signals Culling Function

25.15.10.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.211 Contents of the TAUDnCMORm Register for the Master Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation. 1: INTTAUDnIm generated at the beginning of operation.

(2) TAUDnCMURm for the master channel

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

Table 25.212 Contents of the TAUDnCMURm Register for the Master Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the master channel

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.213 Simultaneous Rewrite Settings for the Master Channel of Interrupt Request Signals Culling Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. 1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.10.5 Register Settings for the Slave Channel

(1) TAUDnCMORm for the slave channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.214 Contents of the TAUDnCMORm Register for the Slave Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm not generated at the beginning of operation.

(2) TAUDnCMURm for the slave channel

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

Table 25.215 Contents of the TAUDnCMURm Register for the Slave Channel of the Interrupt Request Signals Culling Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the slave channel

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite for the slave channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.216 Simultaneous Rewrite Settings for the Slave Channel of Interrupt Request Signals Culling Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count. 1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

25.15.10.6 Operating Procedure for Interrupt Request Signals Culling Function

Table 25.217 Operating Procedure for Interrupt Request Signals Culling Function

		Operation	TAUDn Status
	Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.10.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.15.10.5, Register Settings for the Slave Channel**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.
Restart Operation	Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm (master and slave channels) is set to 1 and the counters of master and slave channels start. INTTAUDnIm is generated on the master channel.
	During Operation	TAUDnCDRm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCNTm of master channel loads TAUDnCDRm value and counts down. When the counter reaches 0000_H: • INTTAUDnIm (master) is generated. • TAUDnCNTm (master) loads TAUDnCDRm value and continues count operation. • TAUDnCNTm of slave channels counts down each time INTTAUDnIm of master channel is detected. When TAUDnCNTm of the slave = 0000_H: • INTTAUDnIm (slave) is generated. • The TAUDnCDRm value is loaded in TAUDnCNTm (slave) and count operation continues.
	Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm stops and retains its current value.

25.15.10.7 Specific Timing Diagram

(1) Interrupt count (master) = interrupt count (slave)

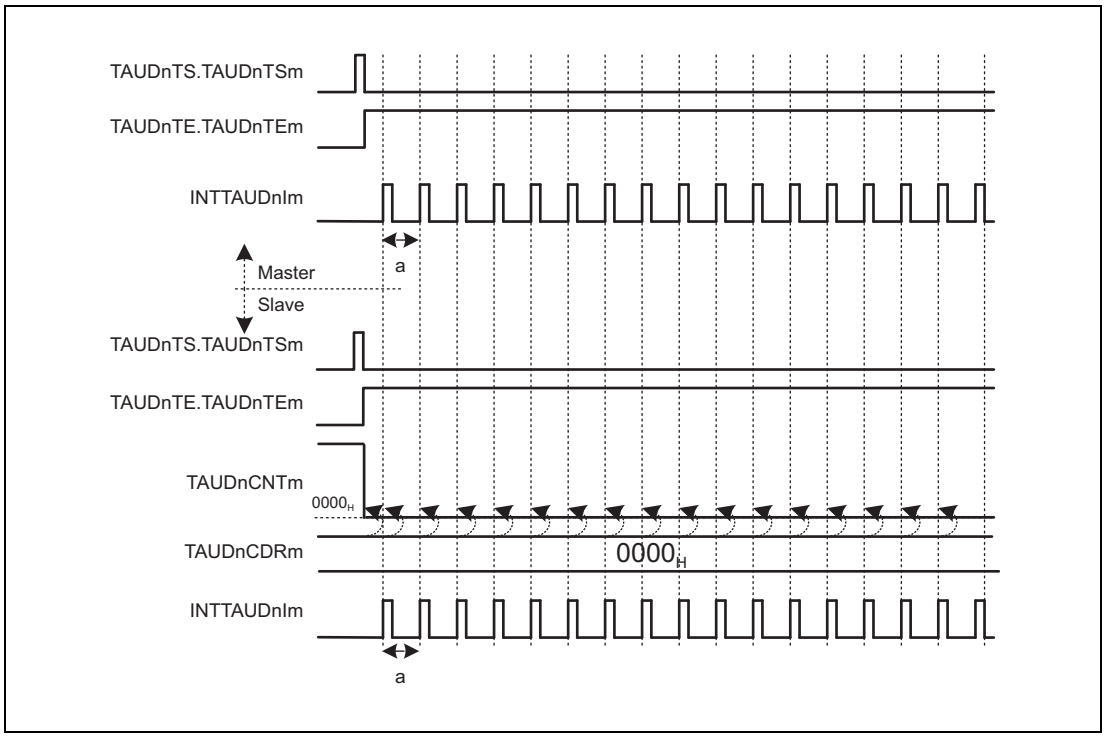


Figure 25.128 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm = 0000_H, the TAUDnCDRm value of the slave channel is loaded into TAUDnCNTm each time INTTAUDnIm of master channel is detected. In other words, TAUDnCNTm is always 0000_H.
- Therefore, an interrupt occurs on the master channel and simultaneously an interrupt occurs on slave channels.

25.16 Synchronous Non-Complementary and Complementary Modulation Output Functions

This section describes functions that generate 6-phase PWM output or triangle PWM output using a master channel and seven slave channels.

25.16.1 Non-Complementary Modulation Output Function Type 1

25.16.1.1 Overview

Summary

This function outputs a PWM signal, a high-level signal, or a low-level signal from TAUDTTOUTm depending on the values of the real-time output bits (TAUDnTRO.TAUDnTROm) and the modulation output enable bits (TAUDnTME.TAUDnTMEm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See **Table 25.219, Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1**).
- The operating mode for slave channels 1 to 7 should be set to one-count mode (See **Table 25.222, Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1**, and **Table 25.225, Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1**).
- TAUDTTOUTm is not used with the master channel of this function.
- TAUDTTOUTm of slave channel 1 is not used with this function, but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 25.7, Channel Output Modes**).
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 1 with non-complementary modulation output (See **Section 25.7, Channel Output Modes**).
- TAUDnCDRm of slave channel 1 should be set to 0000_H.

Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The value of data register (TAUDnCDRm) is loaded into the counter (TAUDnCNTm) and the counter starts to count down. When the counter reaches 0000_H, INTTAUDnIm is generated.

- Slave channel 1:
  Slave channel 1 is set as a channel that triggers real-time output (TAUDnTRC.TAUDnTRCm = 1). If an interrupt occurs on slave channel 1 (TAUDnCDRm is fixed to 0000_H), the value of real-time output bit (TAUDnTRO.TAUDnTROm) of the channel that monitors the interrupt on slave channel 1 is reflected to the TAUDTTOUTm output. After that, the counter returns to $FFFF_H$ and waits for the next interrupt of master channel.

- Slave channel 2:
  Slave channel 2 generates a PWM output. The master channel specifies a PWM output cycle and slave channel 2 specifies a duty cycle. After generating an interrupt, the counter returns to $FFFF_H$ and awaits the next interrupt from the master channel.

Slave channels 3 to 7 operate like slave channel 2.

As described in **Table 25.218, TAUDTTOUTm Output of Slave Channels for Non-Complementary Modulation Output Function Type 1 (TAUDnTOL.TAUDnTOLm = 0)**, a signal output from TAUDTTOUTm depends on the value of the real-time output bit (TAUDnTRO.TAUDnTROm) and modulation output bit (TAUDnTME.TAUDnTMEm) of slave channel.

This function cannot use a forced restart. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnTME.TAUDnTMEm = 0 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs a high-level signal.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm = 1 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs PWM (positive logic) corresponding to the channel.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDTTOUTm are inverted. The PWM signal is negative logic. Only the initial setting of TAUDnTOL.TAUDnTOLm is permitted (cannot be changed during operation).

Table 25.218 TAUDTTOUTm Output of Slave Channels for Non-Complementary Modulation Output Function Type 1 (TAUDnTOL.TAUDnTOLm = 0)

TAUDnTME.TAUDnTMEm	TAUDnTRO.TAUDnTROm	TAUDTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWM (positive logic)

- This function enables simultaneous rewrite. See **Section 25.6, Simultaneous Rewrite**.
- TAUDnCDRm value of slave channel 1 should be set to 0000_H so that a real-time output is triggered at the same time as PWM generation on slave channels 2 to 7.
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.

25.16.1.2 Equations

Slave channels 2 to 7:

Pulse period = [TAUDnCDRm (master) + 1] × count clock cycle

Duty time = [TAUDnCDRm (slave)] × count clock cycle

25.16.1.3 Block Diagram and General Timing Diagram

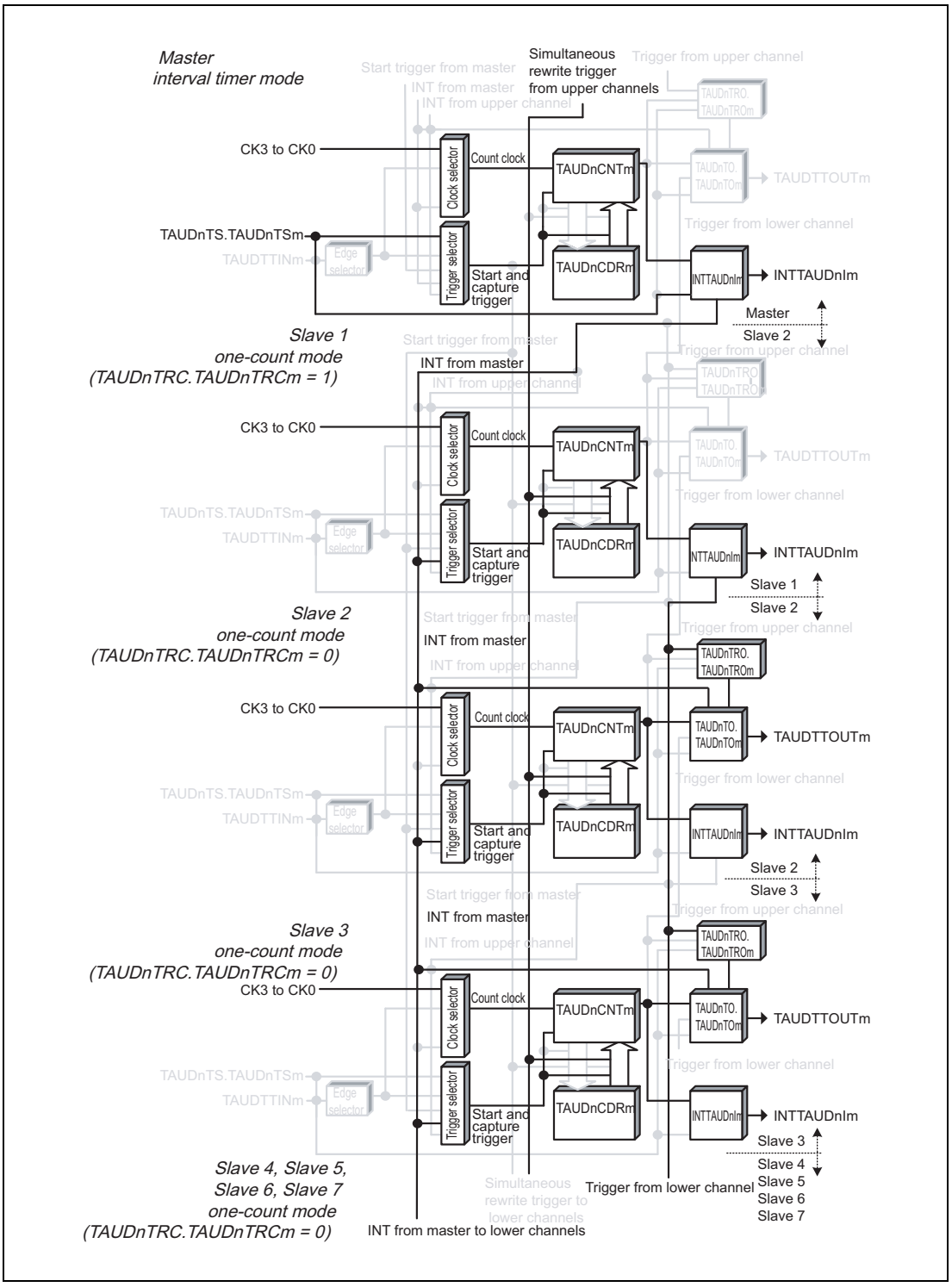


Figure 25.129 Block Diagram of Non-Complementary Modulation Output Function Type 1

The following settings apply to the general timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

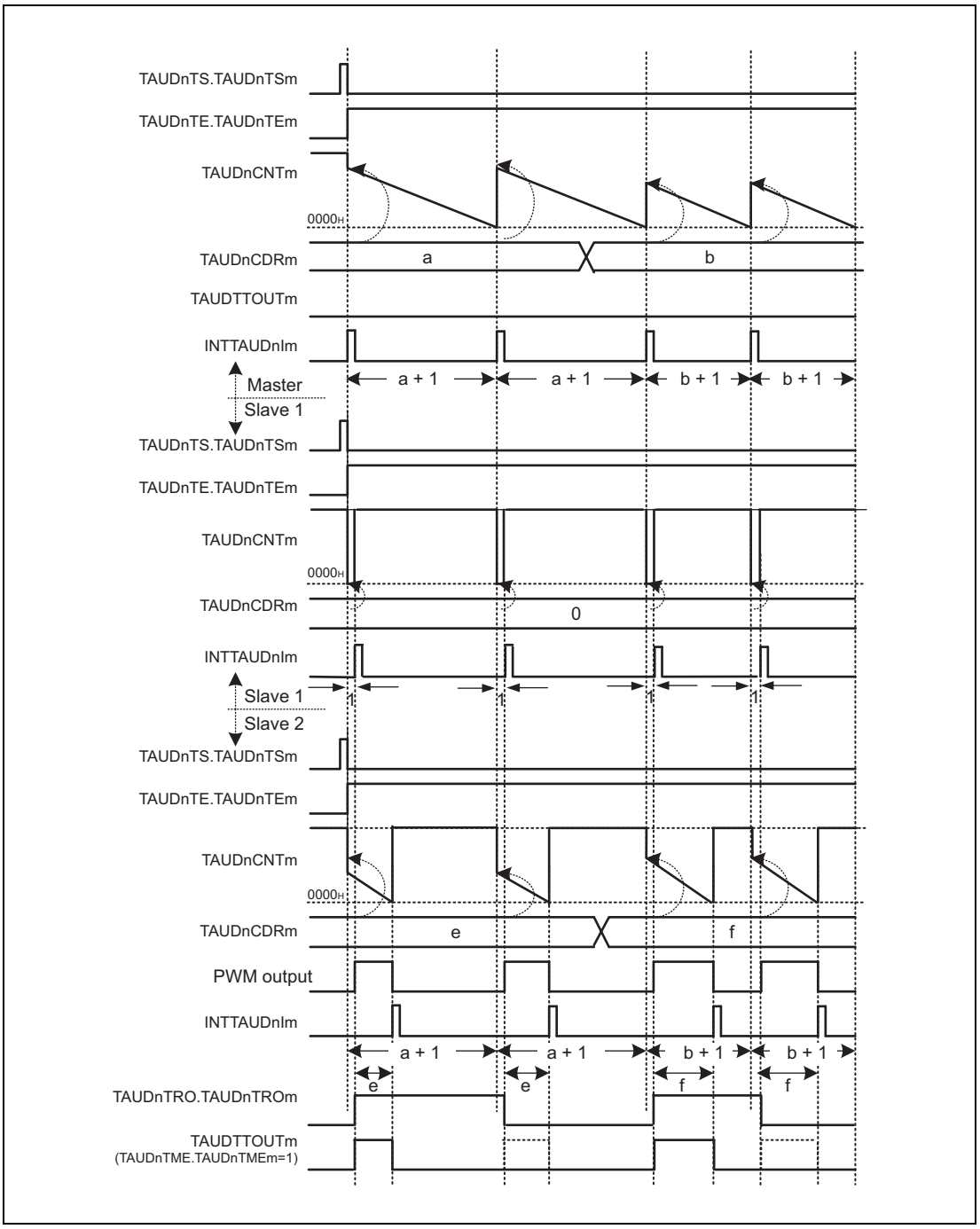


Figure 25.130 General Timing Diagram of Non-Complementary Modulation Output Function Type 1

NOTE

TAUDTTOUTm of slave channel 2 rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

25.16.1.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.219 Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnIm is generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for the master channel

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

Table 25.220 Contents of the TAUDnCMURm Register for the Master Channel of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the master channel

TAUDnTOE.TAUDnTOEm is set to 0 because channel output mode is not used with this function.

(4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.221 Simultaneous Rewrite Settings for the Master Channel of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

NOTE

Use with TAUDnRDS.TAUDnRDSm bit = 1 requires an upper channel higher than the master channel that operates as described in **Section 25.14.1, Simultaneous Rewrite Trigger Generation Function Type 1**.

Conduct operation settings under the following conditions.

- Simultaneous rewrite trigger output function type 1 setting channel: TAUDnRDCm = 1, TAUDnRDSm = 1
  In addition, TAUDnCDRm settings for this channel are as follows:
  = ((TAUDnCDR setting for the master channel targeted for simultaneous rewrite + 1) × Interrupt count) – 1
- Master channel: TAUDnRDCm = 0, TAUDnRDSm = 1
- Slave channel: TAUDnRDCm = 0, TAUDnRDSm = 1

25.16.1.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.222 Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid.

(2) TAUDnCMURm for slave channel 1

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

Table 25.223 Contents of the TAUDnCMURm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous rewrite for slave channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.224 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

25.16.1.6 Register Settings for Slave Channels 2 to 7

(1) TAUDnCMORm for slave channels 2 to 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.225 Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	100: INTTAUDnIm of master channel is a start trigger.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Start trigger during operation is valid.

(2) TAUDnCMURm for slave channels 2 to 7

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

Table 25.226 Contents of the TAUDnCMURm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channels 2 to 7

Table 25.227 Control Bit Settings in Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

(4) Simultaneous rewrite of slave channels 2 to 7

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.228 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 1

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

25.16.1.7 Operating Procedure for Non-Complementary Modulation Output Function Type 1

Table 25.229 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (1/2)

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.16.1.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.16.1.5, Register Settings for Slave Channel 1**. Slave channels 2 to 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.16.1.6, Register Settings for Slave Channels 2 to 7**. Set the value of TAUDnCDRm register of every channel. Set a pulse cycle with TAUDnCDRm of master channel, 0000_H in TAUDnCDRm of slave channel 1, and duty width with TAUDnCDRm of slave channels 2 to 7. Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 25.229 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (2/2)

Restart Operation

	Operation	TAUDn Status
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm of master and slave channels is set to 1 and the counter starts counting down.
During Operation	TAUDnCDRm, TAUDnTRO.TAUDnTROm, and TAUDnTMEm.TAUDnTMEm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel, slave channel 1 and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. When the counter of master channel reaches 0000_H: • INTTAUDnIm is generated. • TAUDnCDRm value of master channel is reloaded into TAUDnCNTm to continue counting down. • PWM output signals of slave channels 2 to 7 are set. • TAUDnCDRm value of slave channel 1 is reloaded into TAUDnCNTm to perform counting down. • TAUDnCDRm value of slave channels 2 to 7 is reloaded into TAUDnCNTm to perform counting down. • When the counter of slave channel 1 reaches 0000_H: – INTTAUDnIm is generated. – The TAUDnTRO.TAUDnTROm value of slave channels 2 to 7 is reflected to the TAUDTTOUTm output. • When the counter of slave channels 2 to 7 reaches 0000_H: – INTTAUDnIm is generated. – PWM output signals of slave channels 2 to 7 are set. TAUDTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high-level signal or low-level signal depending on the values of real-time output bits (TAUDnTRO.TAUDnTROm) and modulation output bit (TAUDnTME.TAUDnTMEm) of a pair of slave channels.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

25.16.1.8 Specific Timing Diagrams

The following settings apply to the specific timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

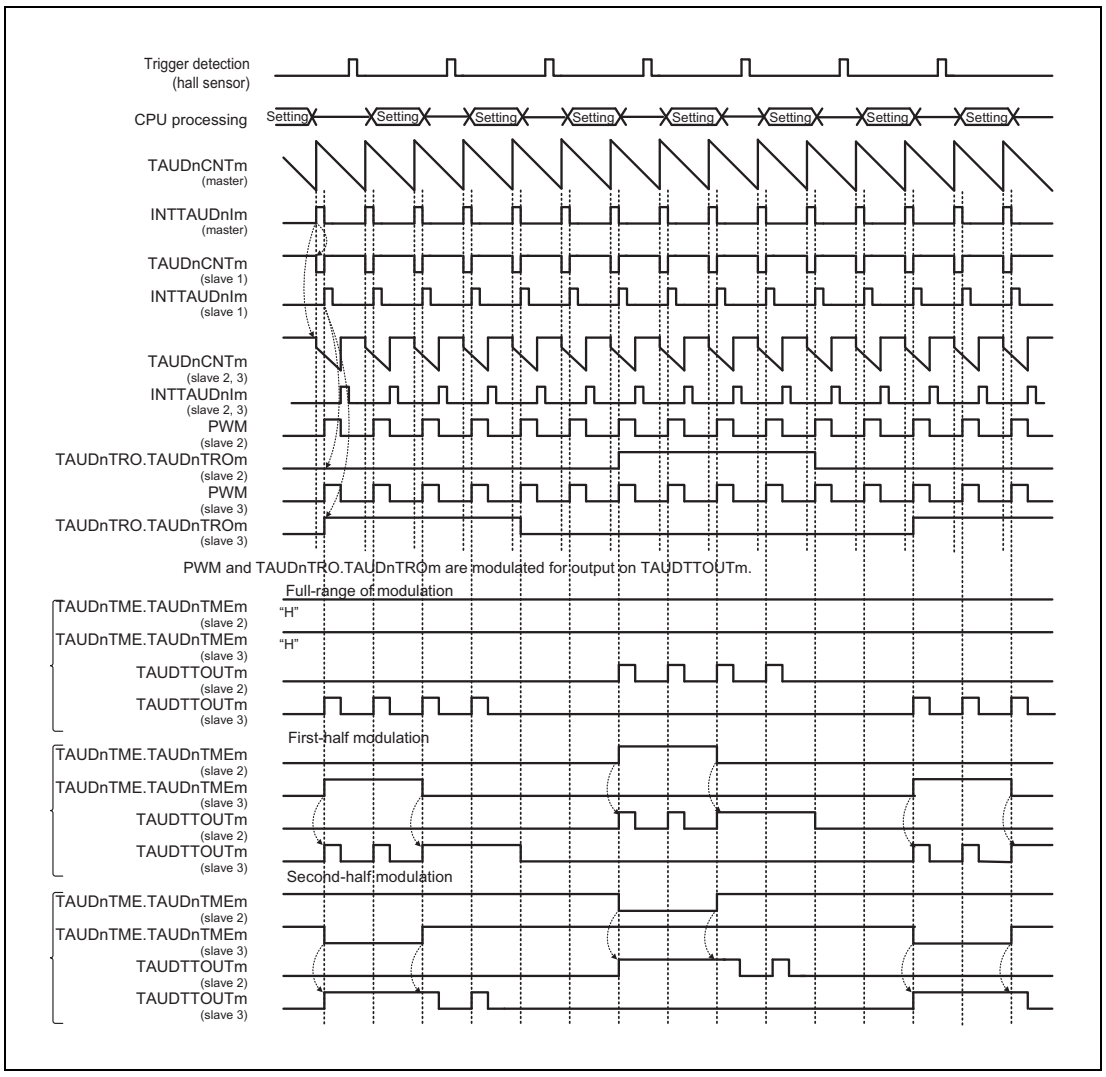


Figure 25.131 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 1

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

The "Setting" symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, and TAUDnTRO.TAUDnTROm can be changed.

TAUDnTME.TAUDnTMEm setting is reflected by detecting the count start timing and master channel cycle. According to the modified setting, modulation waveforms are output from TAUDTTOUTm.

A TAUDnTRO.TAUDnTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

25.16.2 Non-Complementary Modulation Output Function Type 2

25.16.2.1 Overview

Summary

This function outputs a triangular PWM output signal, a high-level signal, or low-level signal from TAUDTTOUTm depending on the real-time output bit value (TAUDnTRO.TAUDnTROm) and the modulation output enable bit value (TAUDnTME.TAUDnTMEm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See **Table 25.231, Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 2**).
- The operating mode for slave channel 1 should be set to event count mode (See **Table 25.235, Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2**).
- The operating mode for slave channels 2 to 7 should be set to count-up/-down mode (See **Table 25.238, Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2**).
- The output mode for the master channel should be set to independent channel output mode 1. (See **Section 25.7, Channel Output Modes**.)
- This function does not use TAUDTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 25.7, Channel Output Modes**).
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with non-complementary modulation output (See **Section 25.7, Channel Output Modes**).

Functional description

The master/slave channel counter is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The value of data register (TAUDnCDRm) is loaded into the counter (TAUDnCNTm).

- Master channel:
  The counter of master channel starts to count down. When the counter reaches 0000_H, INTTAUDnIm is generated.
- Slave channel 1:
  When slave channel 1 detects an interrupt from the master channel, the TAUDnCNTm value is decremented. When an interrupt from the master channel is detected (TAUDnCDRm + 1) times, INTTAUDnIm is generated. Then, the TAUDnCDRm value is loaded into TAUDnCNTm to continue operation subsequently.
  Since slave channel 1 is set as a real-time output trigger channel (TAUDnTRC.TAUDnTRCm = 1), if an interrupt occurs on slave channel 1, the real-time output bit (TAUDnTRO.TAUDnTROm) of the channel which monitors an interrupt on the corresponding channel is reflected to the TAUDTTOUTm output.

- Slave channel 2:
  Once an interrupt is detected from the master channel, TAUDnCNTm counts in the reverse direction. When an interrupt is detected during count-up operation, TAUDnCDRm value is reloaded and then the counter starts to count down.
  If TAUDnCNTm = 0001_H, an interrupt occurs and a PWM output signal is set/reset.

The combined use of the master channel and slave channel 2 generates a PWM output signal. The master channel generates a PWM output cycle and slave channel 2 generate a duty cycle.

Slave channels 3 to 7 operate like slave channel 2.

A signal that is output from TAUDTTOUTm depends on a real-time output bit value (TAUDnTRO.TAUDnTROm) and a modulation output bit value (TAUDnTME.TAUDnTMEm) of the slave channel, as described in **Table 25.230, TAUDTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Function Type 2 (TAUDnTOL.TAUDnTOLm = 0)**.

This function cannot make a forced restart. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnTME.TAUDnTMEm = 0 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs a high-level signal.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm = 1 on slave channels 2 to 7 (TAUDnTOL.TAUDnTOLm = 0):
  - If the channel's TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm outputs PWM (positive logic) corresponding to the channel.
  - If the channel's TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDTTOUTm are inverted. The PWM signal is negative logic. Only the initial setting of TAUDnTOL.TAUDnTOLm is permitted (cannot be changed during operation).

Table 25.230 TAUDTTOUTm Output of Slave Channels in Non-Complementary Modulation Output Function Type 2 (TAUDnTOL.TAUDnTOLm = 0)

TAUDnTME.TAUDnTMEm	TAUDnTRO.TAUDnTROm	TAUDTTOUTm Output
0	0	Low level
	1	High level
1	0	Low level
	1	PWM (positive logic)

- This function enables simultaneous rewrite. See **Section 25.6, Simultaneous Rewrite**.
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7, TAUDnTO.TAUDnTOm is set

to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.

25.16.2.2 Equations

Slave channels 2 to 7:

Carrier cycle (down/up) = [TAUDnCDRm (master) + 1] × 2 × count clock cycle

Duty time = [TAUDnCDRm (master) + 1 - TAUDnCDRm (slave)] × 2 × count clock cycle

25.16.2.3 Block Diagram and General Timing Diagram

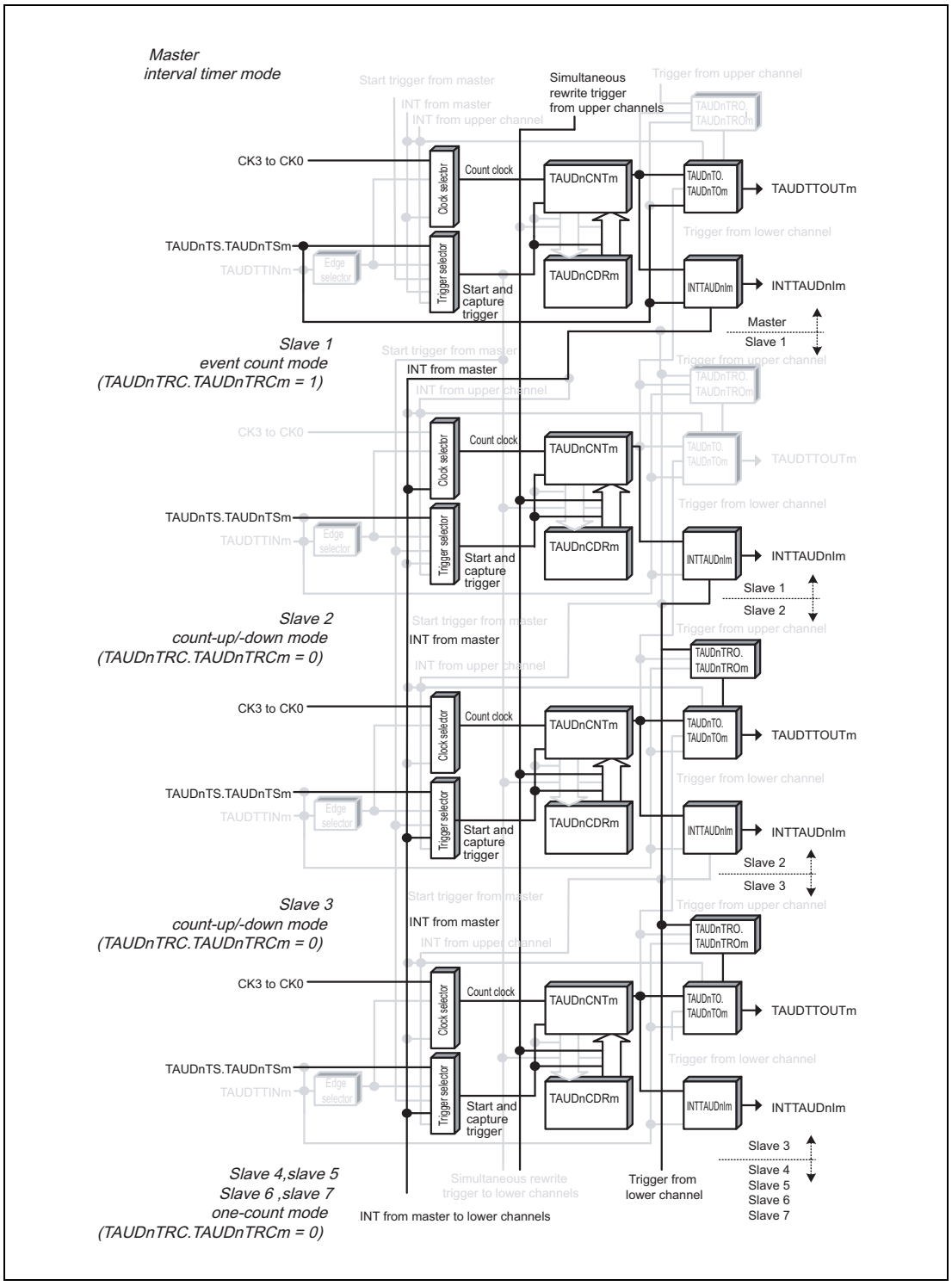


Figure 25.132 Block Diagram of Non-Complementary Modulation Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

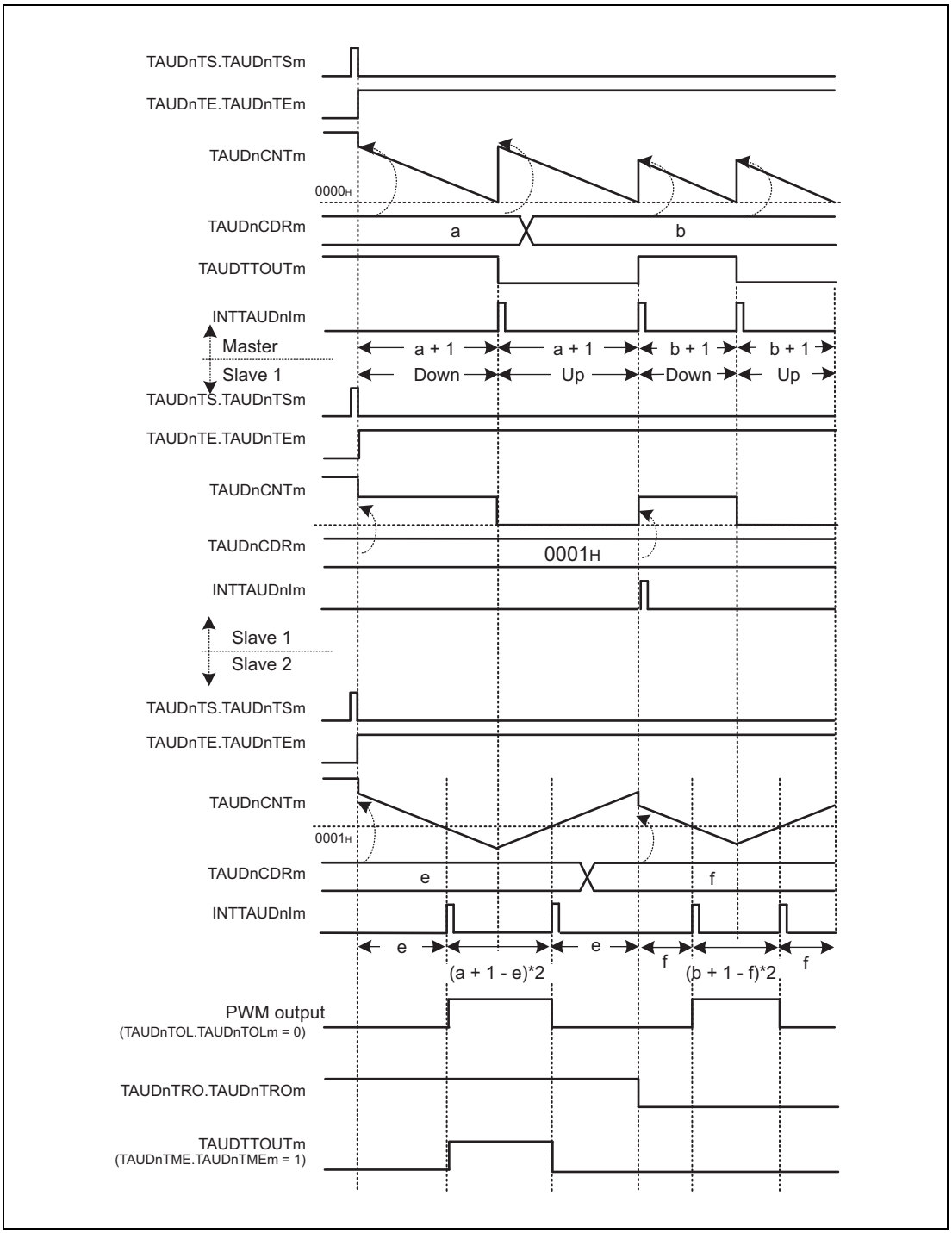


Figure 25.133 General Timing Diagram of Non-Complementary Modulation Output Function Type 2

25.16.2.4 Register Settings of the Master Channel

(1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.231 Contents of the TAUDnCMORm Register for the Master Channel of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time. 1: INTTAUDnIm is generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for the master channel

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

Table 25.232 Contents of the TAUDnCMURm Register for the Master channel of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the master channel

Table 25.233 Control Bit Settings for the Master Channel in Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (toggle mode with TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.234 Simultaneous Rewrite Settings for the Master Channel of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

25.16.2.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.235 Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software. 011: Triggers simultaneous rewrite.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for slave channel 1

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

Table 25.236 Contents of the TAUDnCMURm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous rewrite for slave channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.237 Simultaneous Rewrite Settings for Slave Channel 1 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors the master channel for simultaneous rewrite triggers regardless of the value of this bit if TAUDnRDS.TAUDnRDSm = 0.

25.16.2.6 Register settings for slave channels 2 to 7

(1) TAUDnCMORm for slave channels 2 to 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.238 Contents of the TAUDnCMORm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: The up/down output trigger signal of the master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for slave channels 2 to 7

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

Table 25.239 Contents of the TAUDnCMURm Register for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Output mode for slave channels 2 to 7

Table 25.240 Control Bit Settings in Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: The upper channel generates the real-time output trigger for channel m
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

(4) Simultaneous rewrite for slave channels 2 to 7

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.241 Simultaneous Rewrite Settings for Slave Channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

25.16.2.7 Operating Procedure for Non-Complementary Modulation Output Function Type 2

Table 25.242 Operating Procedure for Non-Complementary Modulation Output Function Type 2 (1/2)

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.16.2.4, Register Settings of the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.16.2.5, Register Settings for Slave Channel 1**. Slave channels 2 to 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.16.2.6, Register settings for slave channels 2 to 7**. Set the value of TAUDnCDRm register of every channel. Set pulse cycle in TAUDnCDRm of master channel, and in TAUDnCDRm of slave channel 1, set the number of interrupts from master channel to be ignored before slave channel 1 generates a real-time output trigger. Set duty width in TAUDnCDRm of slave channels 2 to 7. Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 25.242 Operating Procedure for Non-Complementary Modulation Output Function Type 2 (2/2)

	Operation	TAUDn Status
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm of master and slave channels is set to 1 and the counter starts counting down.
During Operation	TAUDnCDRm, TAUDnTRO.TAUDnTROm, and TAUDnTMEm.TAUDnTMEm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	The TAUDnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. The TAUDnCDRm value of slave channel 1 is loaded and the counter waits for an interrupt from the master channel. When the counter of master channel reaches 0000_H: • INTTAUDnIm is generated. • TAUDnCDRm value is reloaded into TAUDnCNTm to continue counting down. • The TAUDnCNTm value of slave channel 1 decrements by 1 and the counter waits for a next interrupt from the master channel. • TAUDnCNTm of slave channels 2 to 7 reloads the TAUDnCDRm value, or performs counting in opposite direction. • At the same timing when the TAUDnCDRm value is loaded, the TAUDnTME.TAUDnTMEm value of slave channels 2 to 7 is reflected to the TAUDTTOUTm output. • When slave channel 1 detects an interrupt from the master channel for the (TAUDnCDRm + 1) times: – INTTAUDnIm is generated. – The TAUDnTRO.TAUDnTROm value of slave channels 2 to 7 is reflected to the TAUDTTOUTm output. • When the counter of slave channels 2 to 7 reaches 0001_H: – INTTAUDnIm is generated. – PWM output signals of slave channels 2 to 7 are set/reset.
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

Restart Operation

25.16.2.8 Specific Timing Diagrams

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

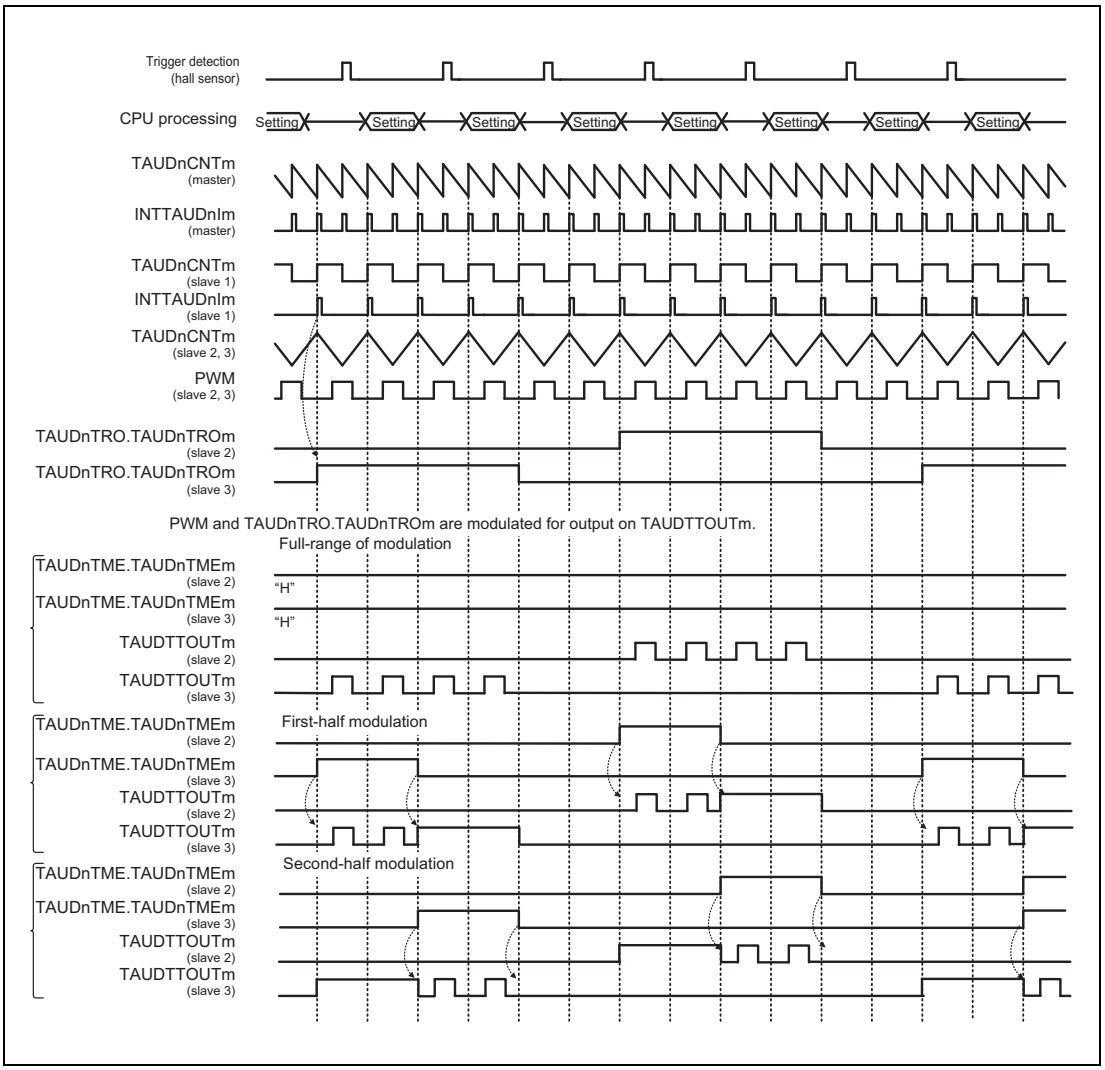


Figure 25.134 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 2

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

The "Setting" symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, and TAUDnTRO.TAUDnTROm can be changed.

TAUDnTME.TAUDnTMEm setting is reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).

TAUDnTRO.TAUDnTROm bit value is set by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

25.16.3 Complementary Modulation Output Function

25.16.3.1 Overview

Summary

This function outputs a triangle PWM output signal, a high-level signal, or low-level signal from TAUDTTOUTm with dead time added, depending on the real-time output bit value (TAUDnTRO.TAUDnTROm) and the modulation output bit value (TAUDnTME.TAUDnTMEm) of a pair of slave channels, and an output level bit value (TAUDnTDL.TAUDnTDLm). Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to interval timer mode (See **Table 25.244, Contents of the TAUDnCMORm Register for the Master Channel of the Complementary Modulation Output Function**).
- The operating mode for slave channel 1 should be set to event count mode (See **Table 25.248, Contents of the TAUDnCMORm Register for Slave Channel 1 of the Complementary Modulation Output Function**).
- The operating mode for slave channels 2, 4 and 6 should be set to count-up/-down mode (See **Table 25.251, Contents of the TAUDnCMORm Register for Slave Channels 2, 4, and 6 of the Complementary Modulation Output Function**).
- The operating mode for slave channels 3, 5 and 7 should be set to one-count mode (See **Table 25.255, Contents of the TAUDnCMORm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output Function**).
  In addition, as the number of occurrences of an interrupt for slave channels 3, 5 and 7 within the carrier cycle is not uniquely determined, do not use the interrupt as an interrupt source.
- The output mode for master channels should be set to independent channel output mode 1 (See **25.7, Channel Output Modes**).
- This function does not use TAUDTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 25.7, Channel Output Modes**).
- The channel output mode for slave channels 2 to 7 should be set to synchronous channel output mode 2 with complementary modulation output (See **Section 25.7, Channel Output Modes**).

Functional description

- Master channel:
  The counter of the master channel is enabled by setting the channel trigger bit (TAUDnTS.TAUDnTSm) to 1. This sets TAUDnTE.TAUDnTEm = 1, enabling count operation. The value of data register (TAUDnCDRm) of the master channel is loaded into the counter (TAUDnCNTm) and the counter starts to count down from this value.
  When the counter of master channel reaches 0000_H, INTTAUDnIm is generated. This decrements the counter value of slave channel 1 by 1 and the counter of slave channel 2 starts to count in the opposite direction.
- Slave channel 1:
  When the counter reaches 0000_H, slave channel 1 waits for the next interrupt from the master

channel. And the TAUDnCDRm value is reloaded into TAUDnCNTm (slave 1) and INTTAUDnIm is generated.
Slave channel 1 is set as a real-time output trigger channel (TAUDnTRC.TAUDnTRCm = 1). The value of real-time output bit (TAUDnTRO.TAUDnTROm) of each channel is applied to the channel that detects the occurrence of an interrupt on slave channel 1 by an interrupt. The real-time output bit value can be changed in any timing by application software but a new value is not applied until an interrupt occurs on slave channel 1.

- Slave channel 2:
  When the slave channel 2 counter reaches 0001_H, the slave channel 3 counter starts counting down. When the slave channel 3 counter reaches 0000_H, an interrupt occurs.
- Slave channels 2 and 3:
  The combined use of the master channel and slave channels 2 and 3 generates a PWM output signal. The master channel generates a PWM output cycle, slave channel 2 generates a duty cycle, and slave channel 3 generates dead time.
- Slave channels 4 to 7:
  Slave channels 4 and 6 operate like slave channel 2. Slave channels 5 and 7 operate like slave channel 3.

A signal that is output from TAUDTTOUTm depends on a real-time output bit value (TAUDnTRO.TAUDnTROm), a modulation output bit value (TAUDnTME.TAUDnTMEm), and an output level bit value (TAUDnTDL.TAUDnTDLm) of the slave channel, as described in **Table 25.243, TAUDTTOUTm Output (TAUDnTOL.TAUDnTOLm = 0) for a Pair of Slave Channels of Complementary Modulation Output Function**.

It is, however, prohibited that a high-level signal is output from both channel 2 and channel 3 (in order to prevent a motor driver short circuit).

Forced restart is not possible for this function. The counter can be stopped by setting TAUDnTT.TAUDnTTm of master and slave channels to 1. This sets TAUDnTE.TAUDnTEm to 0. TAUDnCNTm and TAUDTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUDnTS.TAUDnTSm to 1.

Conditions

- If TAUDnTME.TAUDnTMEm of a pair of channels is set to 1 (TAUDnTOL.TAUDnTOLm = 0):
  - If TAUDnTRO.TAUDnTROm of one channel is set to 1, TAUDTTOUTm outputs the corresponding PWM of the channel.
  - If TAUDnTRO.TAUDnTROm of both channels is set to 0, TAUDTTOUTm of a pair outputs a low-level signal.
- If TAUDnTME.TAUDnTMEm of a pair of channels is set to 0 (TAUDnTOL.TAUDnTOLm = 0):
  - If TAUDnTRO.TAUDnTROm is set to 1, TAUDTTOUTm of the channel outputs a high-level signal.
  - If TAUDnTRO.TAUDnTROm is set to 0, TAUDTTOUTm of the channel outputs a low-level signal.
- If TAUDnTOL.TAUDnTOLm is set to 1, high-level and low-level signals output from TAUDTTOUTm are inverted. The PWM signal is negative logic.

Table 25.243 TAUDTTOUTm Output (TAUDnTOL.TAUDnTOLm = 0) for a Pair of Slave Channels of Complementary Modulation Output Function

TAUDnTME.TAUDnTME2	TAUDnTME.TAUDnTME3	TAUDnTRO.TAUDnTRO2	TAUDnTRO.TAUDnTRO3	TAUDnTDL.TAUDnTDL2	TAUDnTDL.TAUDnTDL3	TAUDTTOUT2 Output	TAUDTTOUT3 Output
0	0	0	0	X	X	Low level	Low level
		0	1	1	0	Low level	High level
		1	0	0	1	High level	Low level
		1	1	X	X	Setting prohibited	Setting prohibited
1	1	0	0	X	X	Low level	Low level
		0	1	1	0	~PWM	PWM
		1	0	0	1	PWM	~PWM
		1	1	X	X	Setting prohibited	Setting prohibited

NOTES

1. In the above table, PWM indicates a positive PWM signal and ~PWM indicates an inverted PWM signal (positive logic). PWM and ~PWM are set by TAUDnTDL.TAUDnTDLm.
2. Any settings not listed above are prohibited.

- If TAUDnTME.TAUDnTMEm is continuously set to 1 while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, full modulation is applied.
- If TAUDnTME.TAUDnTMEm is set to 1 at the first half of the period while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, first-half modulation is applied.
- If TAUDnTME.TAUDnTMEm is set to 1 at the second half of the period while TAUDnTRO.TAUDnTROm of one of paired channels is set to 1, second-half modulation is applied.
- Whether dead time is added to a normal or reverse phase PWM signal when two channels become high-level signal outputs simultaneously depends on a TAUDnTDL.TAUDnTDLm bit value.
  - If TAUDnTDL.TAUDnTDLm = 0, dead time is added to a normal phase PWM signal.
  - If TAUDnTDL.TAUDnTDLm = 1, dead time is added to a reverse phase PWM signal.
  - The operation defined by a TAUDnTDL.TAUDnTDLm bit value should be conducted by application software during operation. To modify TAUDnTDL.TAUDnTDLm, rewrite it during the period when TAUDnTRO.TAUDnTROm is 00_B.
- The TAUDnCDRm value of slave channel 1 should be set to the value to generate INTTAUDnIm of slave channel 1 at a carrier cycle (peak interrupt timing).
- If TAUDnTOL.TAUDnTOLm is set to 0 on slave channels 2 to 7:
  - If TAUDnTDL.TAUDnTDLm is set to 0, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.
  - If TAUDnTDL.TAUDnTDLm is set to 1, TAUDnTO.TAUDnTOm is set to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.
- If TAUDnTOL.TAUDnTOLm is set to 1 on slave channels 2 to 7:
  - If TAUDnTDL.TAUDnTDLm is set to 0, TAUDnTO.TAUDnTOm is set to 1 (high) before TAUDnTE.TAUDnTEm is set to 0.
  - If TAUDnTDL.TAUDnTDLm is set to 1, TAUDnTO.TAUDnTOm is set to 0 (low) before TAUDnTE.TAUDnTEm is set to 0.

- This function enables simultaneous rewrite. See **Section 25.6, Simultaneous Rewrite**.

25.16.3.2 Equations

Pulse period = (TAUDnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUDnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUDnCDRm (master) + 1) × 2 × count clock cycle

For slave channels 2 and 3:

PWM signal width (positive phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2) × 2) – (TAUDnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (negative phase) = [(TAUDnCDRm (master) + 1 – TAUDnCDRm (slave 2) × 2) + (TAUDnCDRm (slave 3) + 1)] × count clock cycle

For slave channels 4 to 7:

Slave channels 4 and 6 are calculated in the same way as slave channel 2, whereas slave channels 5 and 7 are calculated as slave channel 3.

25.16.3.3 Block Diagram and General Timing Diagram

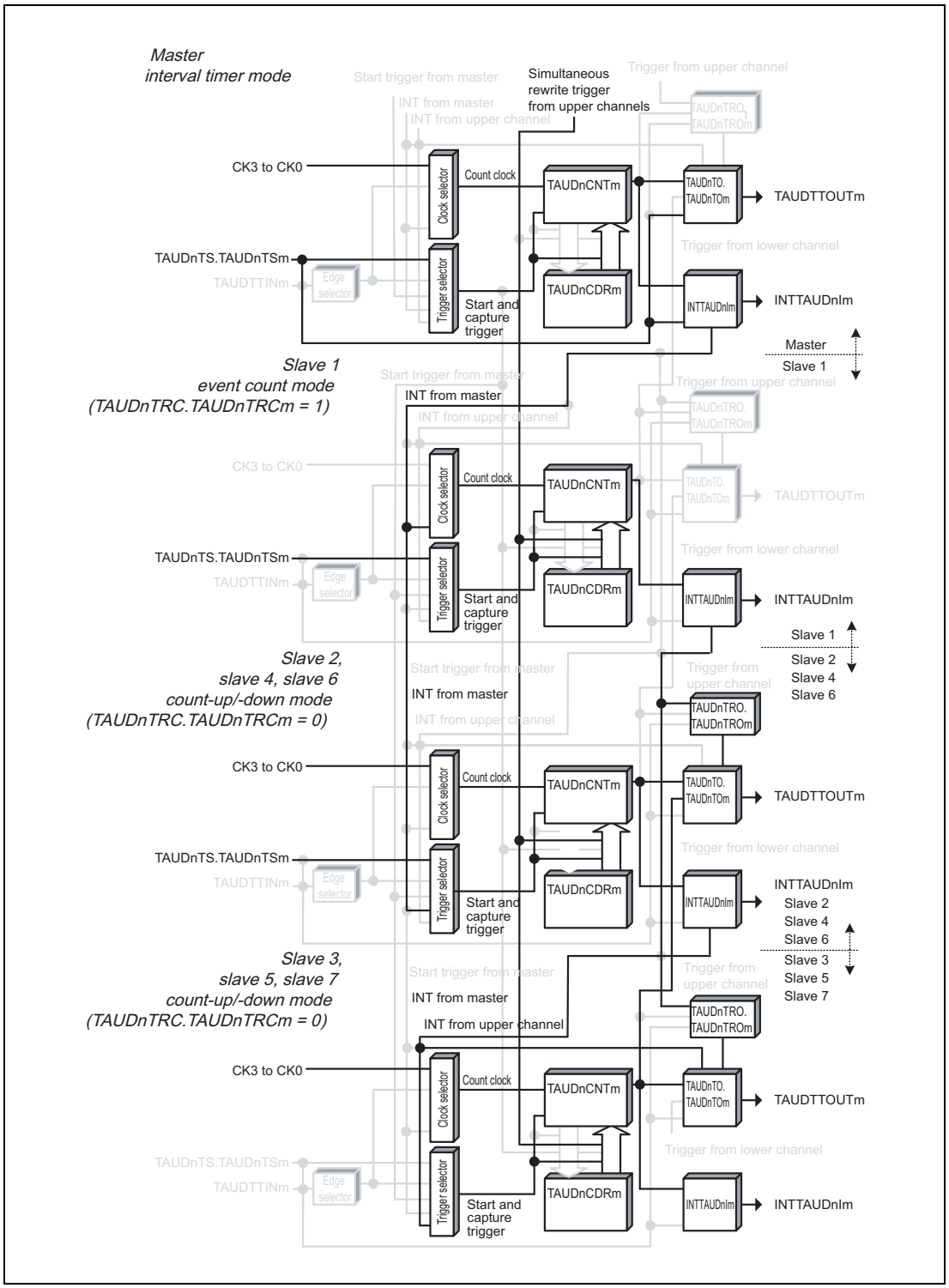


Figure 25.135 Block Diagram of Complementary Modulation Output Function

The following settings apply to the general timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channel 1: TAUDnCDRm = 0001_H

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

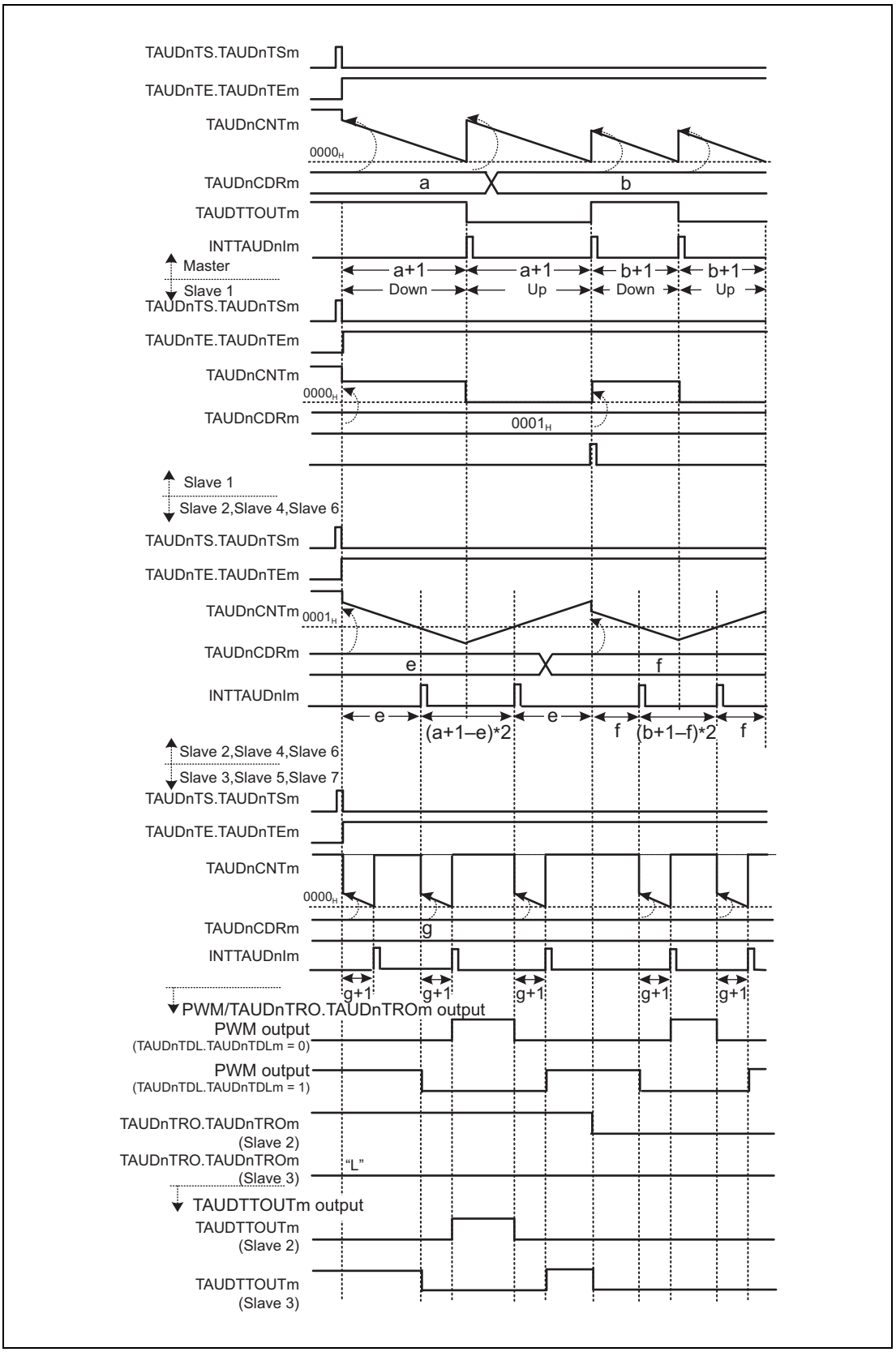


Figure 25.136 General Timing Diagram of Complementary Modulation Output Function

25.16.3.4 Register Settings for the Master Channel

(1) TAUDnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.244 Contents of the TAUDnCMORm Register for the Master Channel of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	1: Master channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	0: INTTAUDnIm is not generated and TAUDTTOUTm is not toggled at the beginning of operation or at a restart time. 1: INTTAUDnIm is generated to toggle TAUDTTOUTm at the beginning of an operation.

(2) TAUDnCMURm for the master channel

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

Table 25.245 Contents of the TAUDnCMURm Register for the Master Channel of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the master channel

Table 25.246 Control Bit Settings in Independent Channel Output Mode 1

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	0: Independent channel output
TAUDnTOC.TAUDnTOCm	0: Operating mode 1 (Toggle mode if TAUDnTOM.TAUDnTOMm = 0)
TAUDnTOL.TAUDnTOLm	0: The setting is disabled in toggle mode (the value after reset).
TAUDnTDE.TAUDnTDEm	0: Disables dead time operation
TAUDnTDM.TAUDnTDMm	0: When dead time operation is disabled (TAUDnTDE.TAUDnTDEm = 0), set these bits to 0
TAUDnTDL.TAUDnTDLm	
TAUDnTRE.TAUDnTREm	0: Disables real-time output
TAUDnTRO.TAUDnTROm	0: When real-time output is disabled (TAUDnTRE.TAUDnTREm = 0), set these bits to 0
TAUDnTRC.TAUDnTRCm	
TAUDnTME.TAUDnTMEm	0: Disables modulation

(4) Simultaneous rewrite for the master channel

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.247 Simultaneous Rewrite Settings for the Master Channel of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

25.16.3.5 Register Settings for Slave Channel 1

(1) TAUDnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDn MAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.248 Contents of the TAUDnCMORm Register for Slave Channel 1 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	11: INTTAUDnIm of the master channel is used as the count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	000: Trigger the counter using software. 011: Triggers simultaneous rewrite.
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0011: Event count mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for slave channel 1

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

Table 25.249 Contents of the TAUDnCMURm Register for Slave Channel 1 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode

TAUDnTOE.TAUDnTOEm is set to 0 because the channel output mode is not used on slave channel 1 with this function. However, this mode can be used in independent channel output mode controlled by software.

CAUTION

TAUDnTRC.TAUDnTRCm should be set to 1 because slave channel 1 is used as a real-time output trigger channel.

(4) Simultaneous rewrite for slave channel 1

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.250 Simultaneous Rewrite Settings for Slave Channel 1 of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

25.16.3.6 Register settings for slave channels 2, 4, and 6

(1) TAUDnCMORm for slave channels 2, 4, and 6

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.251 Contents of the TAUDnCMORm Register for Slave Channels 2, 4, and 6 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	111: Up/down output trigger signal of master channel
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	1001: Count-up/-down mode
0	TAUDnMD0	0: INTTAUDnIm is not generated at the beginning of operation or at a restart time.

(2) TAUDnCMURm for slave channels 2, 4, and 6

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

Table 25.252 Contents of the TAUDnCMURm Register for Slave Channels 2, 4, and 6 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Output mode for slave channels 2, 4, and 6

Table 25.253 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time output trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

CAUTION

At the PWM output, set TAUDnTDL.TAUDnTDLm exclusively from odd channels.

(4) Simultaneous rewrite for slave channels 2, 4, and 6

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.254 Simultaneous Rewrite Settings for Slave Channels 2, 4, and 6 of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: A simultaneous rewrite trigger signal is generated when master channel starts to count and the corresponding slave channel is at the peak of a triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

25.16.3.7 Register settings for slave channels 3, 5, and 7

(1) TAUDnCMORm for slave channels 3, 5, and 7

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUDnCKS[1:0]		TAUDnCCS[1:0]		TAUDnMAS	TAUDnSTS[2:0]			TAUDnCOS[1:0]		—	TAUDnMD[4:1]				TAUDnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 25.255 Contents of the TAUDnCMORm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
15, 14	TAUDnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUDnCKS[1:0] bits of the master and slave channels must be identical.
13, 12	TAUDnCCS[1:0]	00: Uses an operation clock as a count clock
11	TAUDnMAS	0: Slave channel
10 to 8	TAUDnSTS[2:0]	110: Dead time trigger
7, 6	TAUDnCOS[1:0]	00: Unused. Set to 00.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUDnMD[4:1]	0100: One-count mode
0	TAUDnMD0	1: Enables start trigger detection while counting.

(2) TAUDnCMURm for slave channels 3, 5, and 7

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

Table 25.256 Contents of the TAUDnCMURm Register for Slave Channel 3, 5, and 7 of the Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Output mode for slave channels 3, 5, and 7

Table 25.257 Control Bit Settings in Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit Name	Setting
TAUDnTOE.TAUDnTOEm	1: Enables independent channel output mode
TAUDnTOM.TAUDnTOMm	1: Synchronous channel output
TAUDnTOC.TAUDnTOCm	1: Operating mode 2
TAUDnTOL.TAUDnTOLm	0: Positive logic 1: Negative logic
TAUDnTDE.TAUDnTDEm	1: Enables dead time operation.
TAUDnTDM.TAUDnTDMm	0: Adds dead time if an interrupt is detected on an even upper channel and the conditions set by TAUDnTDL.TAUDnTDLm are satisfied.
TAUDnTDL.TAUDnTDLm	0: Adds dead time to normal phase. 1: Adds dead time to reverse phase.
TAUDnTRE.TAUDnTREm	1: Enables real-time output.
TAUDnTRO.TAUDnTROm	0: Real-time output is low. 1: Real-time output is high.
TAUDnTRC.TAUDnTRCm	0: Upper channel generates a real-time trigger for channel m.
TAUDnTME.TAUDnTMEm	0: Disables modulation 1: Enables modulation

CAUTION

At the PWM output, set TAUDnTDL.TAUDnTDLm exclusively from even channels.

(4) Simultaneous rewrite for slave channels 3, 5, and 7

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 25.258 Simultaneous Rewrite Settings for Slave Channels 3, 5, and 7 of Complementary Modulation Output Function

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Monitors master channel for simultaneous rewrite triggers. 1: Monitors upper channel other than the channel group for simultaneous rewrite triggers.
TAUDnRDM.TAUDnRDMm	1: Simultaneous rewrite trigger signal is generated when master channel counter is started and the corresponding slave channel is at the peak of triangular wave.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel. Monitors master channel for simultaneous rewrite triggers, regardless of the value of this bit, when TAUDnRDS.TAUDnRDSm = 0.

25.16.3.8 Operating Procedure for Complementary Modulation Output Function

Table 25.259 Operating Procedure for Complementary Modulation Output Function (1/2)

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.16.3.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.16.3.5, Register Settings for Slave Channel 1**. Slave channels 2, 4, and 6: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.16.3.6, Register settings for slave channels 2, 4, and 6**. Slave channels 3, 5, and 7: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 25.16.3.7, Register settings for slave channels 3, 5, and 7**. Set the value of TAUDnCDRm register of every channel. Set a pulse cycle using TAUDnCDRm of master channel, and an interrupt count of master channel ignored using TAUDnCDRm of slave channel 1. Also set a duty width in TAUDnCDRm of slave channels 2, 4, and 6, and a dead time delay on slave channels 3, 5, and 7. Set TAUDnTRC.TAUDnTRCm to 1 on slave channel 1.	Channel operation is stopped.

Table 25.259 Operating Procedure for Complementary Modulation Output Function (2/2)

Restart Operation

	Operation	TAUDn Status
Start Operation	Set TAUDnTS.TAUDnTSm of master and slave channels to 1 simultaneously. TAUDnTS.TAUDnTSm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm of master and slave channels is set to 1 and the counter starts counting down.
During Operation	TAUDnCDRm, TAUDnTRO.TAUDnTROm, TAUDnTME.TAUDnTMEm, and TAUDnTDL.TAUDnTDLm can be changed at any time. TAUDnCNTm and TAUDnRSF.TAUDnRSFm can be read at any time. TAUDnRDT.TAUDnRDTm can be changed during operation.	TAUDnCDRm value of master channel and slave channels 2 to 7 is loaded into TAUDnCNTm to perform counting down. TAUDnCDRm value of slave channel 1 is loaded and the counter waits for a master channel interrupt. When the counter of master channel reaches 0000_H: • INTTAUDnIm is generated. • TAUDnCDRm value is reloaded into TAUDnCNTm to continue counting down. • TAUDnCNTm value of slave channel 1 decrements by 1 and the counter waits for the next master channel interrupt. • TAUDnCNTm of slave channels 2, 4, and 6 reloads the TAUDnCDRm value, or performs counting in opposite direction. • At the same timing when the TAUDnCDRm value of slave channels 2, 4, and 6 is loaded, the TAUDnTME.TAUDnTMEm value of slave channels 2 to 7 is reflected to the TAUDTTOUTm output. • The counter of slave channel 1 waits for the next interrupt from the master channel when reaching 0000_H. When the interrupt is detected: – TAUDnCDRm value is reloaded into TAUDnCNTm and the counter waits for the next master channel interrupt. – INTTAUDnIm is generated. – TAUDnTRO.TAUDnTROm is changeable. • When the counter of slave channels 2, 4, and 6 reaches 0001_H: – INTTAUDnIm is generated. – PWM output of slave channel m is set/reset (when the specified condition of the channel output mode is matched). – TAUDnCDRm value of slave channels 3, 5, and 7 is loaded into TAUDnCNTm to perform counting down. • When the counter of slave channels 3, 5, and 7 reaches 0000_H: – INTTAUDnIm is generated. – PWM output of slave channel m is set/reset (when the specified condition of the channel output mode is matched).
Stop Operation	Set TAUDnTT.TAUDnTTm of master and slave channels to 1 simultaneously. TAUDnTT.TAUDnTTm is a trigger bit, which is automatically cleared to 0.	TAUDnTE.TAUDnTEm is cleared to 0 and the counter stops. TAUDnCNTm and TAUDTTOUTm stop and retain their current values.

25.16.3.9 Specific Timing Diagrams

The following settings apply to the timing diagram.

- Master channel: INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)
- Slave channel 1: TAUDnCDRm = 0001_H
- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

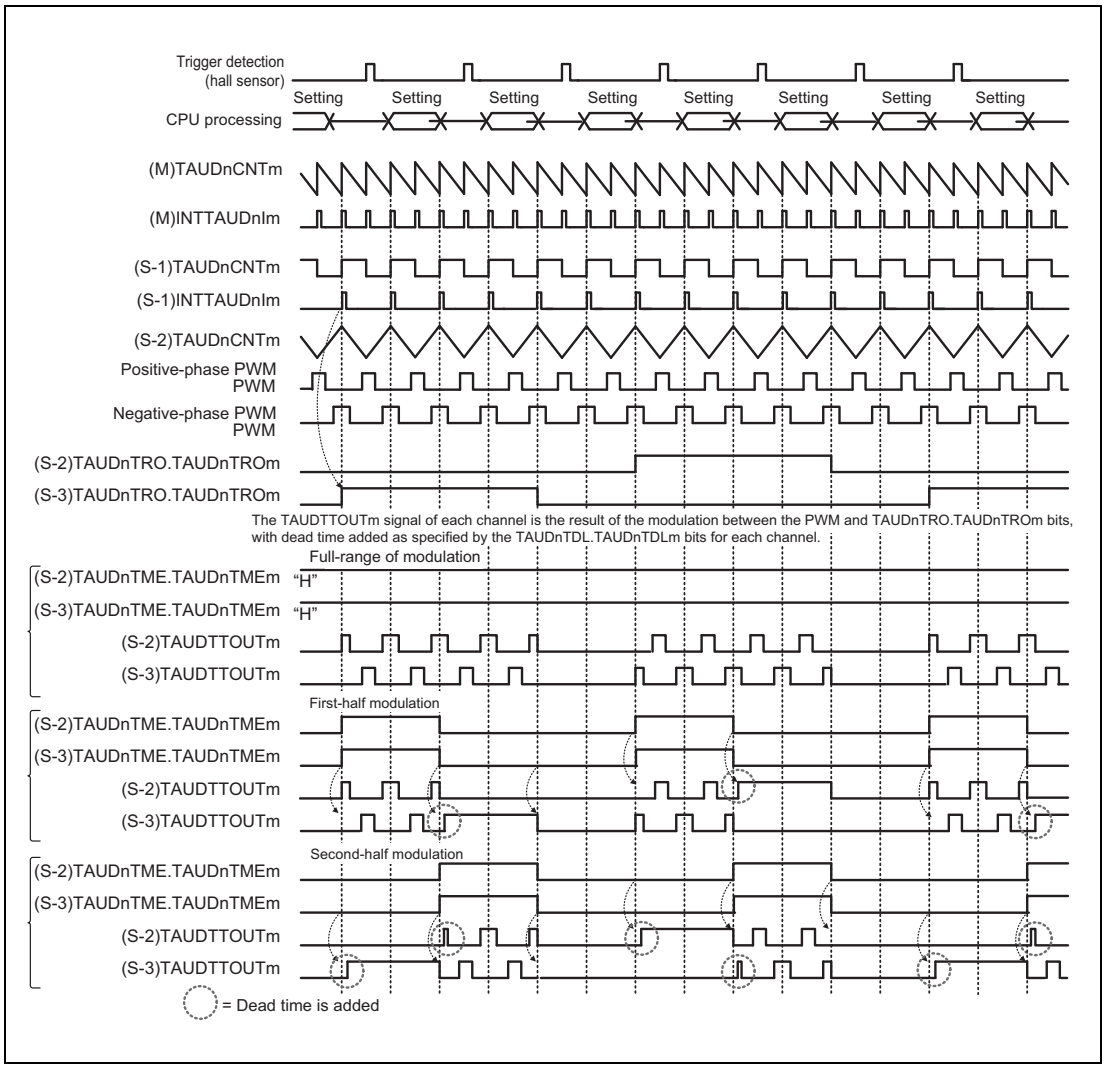


Figure 25.137 Specific Timing Diagram of Complementary Modulation Output Function

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUDnTME.TAUDnTMEm bits of lower slave channels during operation.

A modulated PWM output signal and TAUDnTRO.TAUDnTROm bit value are output from slave channels 2 and 3.

TAUDnTME.TAUDnTMEm and TAUDnTDL.TAUDnTDLm settings are reflected by detecting the count start timing and triangle PWM carrier cycle (peak interrupt timing).

TAUDnTRO.TAUDnTROm bit value is specified by software, but a new setting is applied only when an interrupt occurs on slave channel 1.

NOTE

Dead time is added to suppress simultaneous change of PWM edges of normal and reverse phases.

The “Setting”symbol indicates a time period when the values of TAUDnCDRm, TAUDnTME.TAUDnTMEm, TAUDnTRO.TAUDnTROm, and TAUDnTDL.TAUDnTDLm can be changed.

Section 26 Timer Array Unit J (TAUJ)

This section contains a generic description of the timer array unit J (TAUJ).

The first part of this section describes the RH850/F1M specific features such as the number of units and the register base addresses. The remainder of the section describes the functions and registers of the TAUJ.

26.1 Features of RH850/F1M TAUJ

26.1.1 Number of Units

This microcontroller has the following number of TAUJ units.

Table 26.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	2		
Name	TAUJn (n = 0, 1)		

TAUJn has the following number of channels of timers.

Table 26.2 TAUJn Unit Configurations and Channels

Unit Name (Channel Name) TAUJn	Channels per Unit	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
TAUJ0	4	√	√	√
TAUJ1	4	√	√	√

Table 26.3 Indices

Index	Description
n	Throughout this section, the individual TAUJ units are identified by the index “n”; for example, TAUJnTOM is the TAUJn channel output mode register.
m	The TAUJ has 4 channels. Throughout this section, the individual channels are identified by the index “m” (m = 0 to 3), thus a certain channel is denoted as CHm. The even numbered channels (m = 0, 2) are denoted as CHm_even. The odd numbered channels (m = 1, 3) are denoted as CHm_odd.

26.1.2 Register Base Address

TAUJn base addresses are listed in the following table.

TAUJn register addresses are given as offsets from the base addresses.

Table 26.4 Register Base Addresses

Base Address Name	Base Address
<TAUJ0_base>	FFE5 0000_H
<TAUJ1_base>	FFE5 1000_H

26.1.3 Clock Supply

The TAUJn clock supply is shown in the following table.

Table 26.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
TAUJ0	PCLK	CKSCLK_ATAUJ
	Register access clock	CKSCLK_ATAUJ
TAUJ1	PCLK	CKSCLK_IPERI1
	Register access clock	CKSCLK_IPERI1

26.1.4 Interrupt Requests

TAUJn interrupt requests are listed in the following table.

Table 26.6 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
TAUJ0			
INTTAUJ0I0	Channel 0 interrupt	80	21
INTTAUJ0I1	Channel 1 interrupt	81	80
INTTAUJ0I2	Channel 2 interrupt	82	81
INTTAUJ0I3	Channel 3 interrupt	83	22
TAUJ1			
INTTAUJ1I0	Channel 0 interrupt	168	46
INTTAUJ1I1	Channel 1 interrupt	169	100
INTTAUJ1I2	Channel 2 interrupt	170	47
INTTAUJ1I3	Channel 3 interrupt	171	101

26.1.5 Reset Sources

TAUJn reset sources are listed in the following table. TAUJn is initialized by these reset sources.

Table 26.7 Reset Sources

Unit Name	Reset Source
TAUJ0	All reset sources except the transition to DeepSTOP mode (AWORES)
TAUJ1	All reset sources (ISORES)

26.1.6 External Input/Output Signals

External input/output signals of TAUJn are listed below.

Table 26.8 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal Name
TAUJ0		
TAUJTTIN0, TAUJTTIN1	Channel 0, 1 input	TAUJ0I0, TAUJ0I1
TAUJTTIN2	Channel 2 input	TAUJ0I2 or RTCA0OUT*1
TAUJTTIN3	Channel 3 input	TAUJ0I3 or RTCA0OUT*1
TAUJTTOUT0 to TAUJTTOUT3	Channel 0 to 3 output	TAUJ0O0 to TAUJ0O3
TAUJ1		
TAUJTTIN0 to TAUJTTIN3	Channel 0 to 3 input	TAUJ1I0 to TAUJ1I3
TAUJTTOUT0 to TAUJTTOUT3	Channel 0 to 3 output	TAUJ1O0 to TAUJ1O3

Note 1. For details, see **Section 26.1.8, TAUJ0 Input Selection**.

26.1.7 Internal Input/Output Signals

The internal input/output signals of TAUJn are listed below.

Table 26.9 Internal Input/Output Signals

Unit Signal Name	Description	Connected to
TAUJnTSSTm*1	Simultaneous channel start trigger input	PIC

Note 1. n = 1 only. TAUJ0TSSTm is not connected to PIC.

26.1.8 TAUJ0 Input Selection

The 1-Hz pulse output (RTCA0OUT) from RTCA0 can be input to TAUJTTIN2 and TAUJTTIN3 as shown in the following figure.

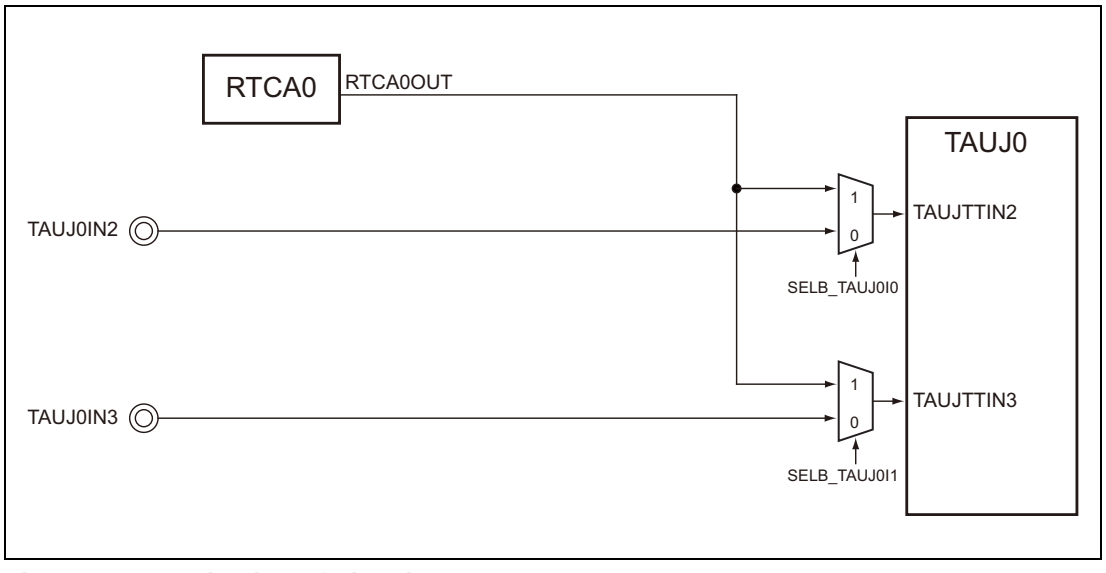


Figure 26.1 Selection of Signals Input to TAUJ0

The following table shows how to select signals input to the TAUJ.

Table 26.10 TAUJ0 Input Selections

Input Signal	Function	Settings
TAUJTTIN2	Port TAUJ0I2	SELB_TAUJ0I.SELB_TAUJ0I0 = 0
	RTCA0OUT (Real-Time Clock 1 Hz output)	SELB_TAUJ0I.SELB_TAUJ0I0 = 1
TAUJTTIN3	Port TAUJ0I3	SELB_TAUJ0I.SELB_TAUJ0I1 = 0
	RTCA0OUT (Real-Time Clock 1 Hz output)	SELB_TAUJ0I.SELB_TAUJ0I1 = 1

26.1.8.1 SELB_TAUJ0I — TAUJTTINm Input Signal Selection Register

This register selects the TAUJ0 input signals.

Access: This register can be read or written in 8-bit units.

Address: FFE5 4000$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SELB_TAUJ0I1	SELB_TAUJ0I0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 26.11 SELB_TAUJ0I Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SELB_TAUJ0I1	Selection of TAUJTTIN3 input signal: 0: Port TAUJ0I3 1: RTCA0OUT
0	SELB_TAUJ0I0	Selection of TAUJTTIN2 input signal: 0: Port TAUJ0I2 1: RTCA0OUT

26.2 Overview

26.2.1 Functional Overview

The TAUJ has the following functions:

- Independent channel operation function (operated using a single channel)
- Synchronous channel operation function (operated using a master channel and multiple slave channels)

The TAUJ is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 4 channels, each equipped with a 32-bit counter TAUJnCNTm and a 32-bit data register TAUJnCDRm to hold the count start value or compare value.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in two operating modes, either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, it can be operated independent of all other channels.

The synchronous operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

26.2.2 Terms

In this section, the following terms are used.

Independent channel operation function/synchronous operation channel operation function

TAUJ has 4 channels, and provides an independent channel operation function that individual channels operate independently and a synchronous channel operation function that is implemented by using a combination of channels.

- The independent channel operation function can be used any channel independently of all other channels.
- The synchronous channel operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

Channel group

In the synchronous channel operation function, all channels that depend on each other are referred to as a “channel group”.

A channel group has one master channel and one or more slave channels.

Upper/lower channel

Based on the channel number m, a channel with a smaller channel number or higher channel number can be referred to as “upper” or “lower” channel:

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

For instance, as to channel 2, channel 1 is an upper channel and channel 3 is a lower channel. Channel 0 is the highest channel and channel 3 is the lowest channel.

26.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 26.12 Functional List of TAUJ Operations

Operation Function	Example
Independent Channel Operation Functions	**Section 26.12**
Interval Timer Function	**Section 26.12.1**
TAUJTTINm Input Interval Timer Function	**Section 26.12.2**
TAUJTTINm Input Pulse Interval Measurement Function	**Section 26.12.3**
TAUJTTINm Input Signal Width Measurement Function	**Section 26.12.4**
TAUJTTINm Input Position Detection Function	**Section 26.12.5**
TAUJTTINm Input Period Count Detection Function	**Section 26.12.6**
Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)	**Section 26.12.7**
Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)	**Section 26.12.8**
Synchronous Channel Operation Functions	**Section 26.13**
PWM Output Function	**Section 26.13.1**

26.2.4 TAUJ I/O and Interrupt Request Signals

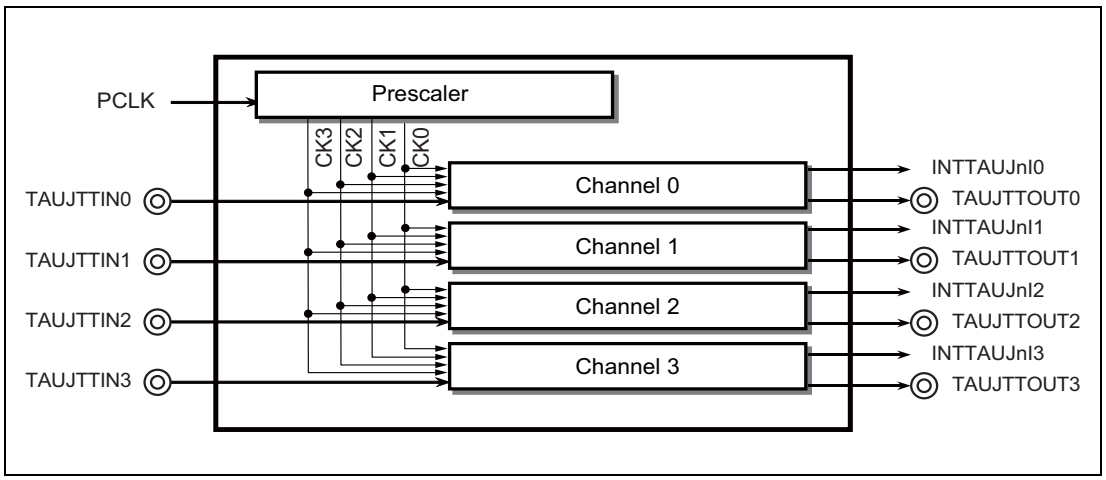


Figure 26.2 TAUJ I/O and Interrupt Request Signals

26.2.5 Block Diagram

The following figure shows the main components of the TAUJ.

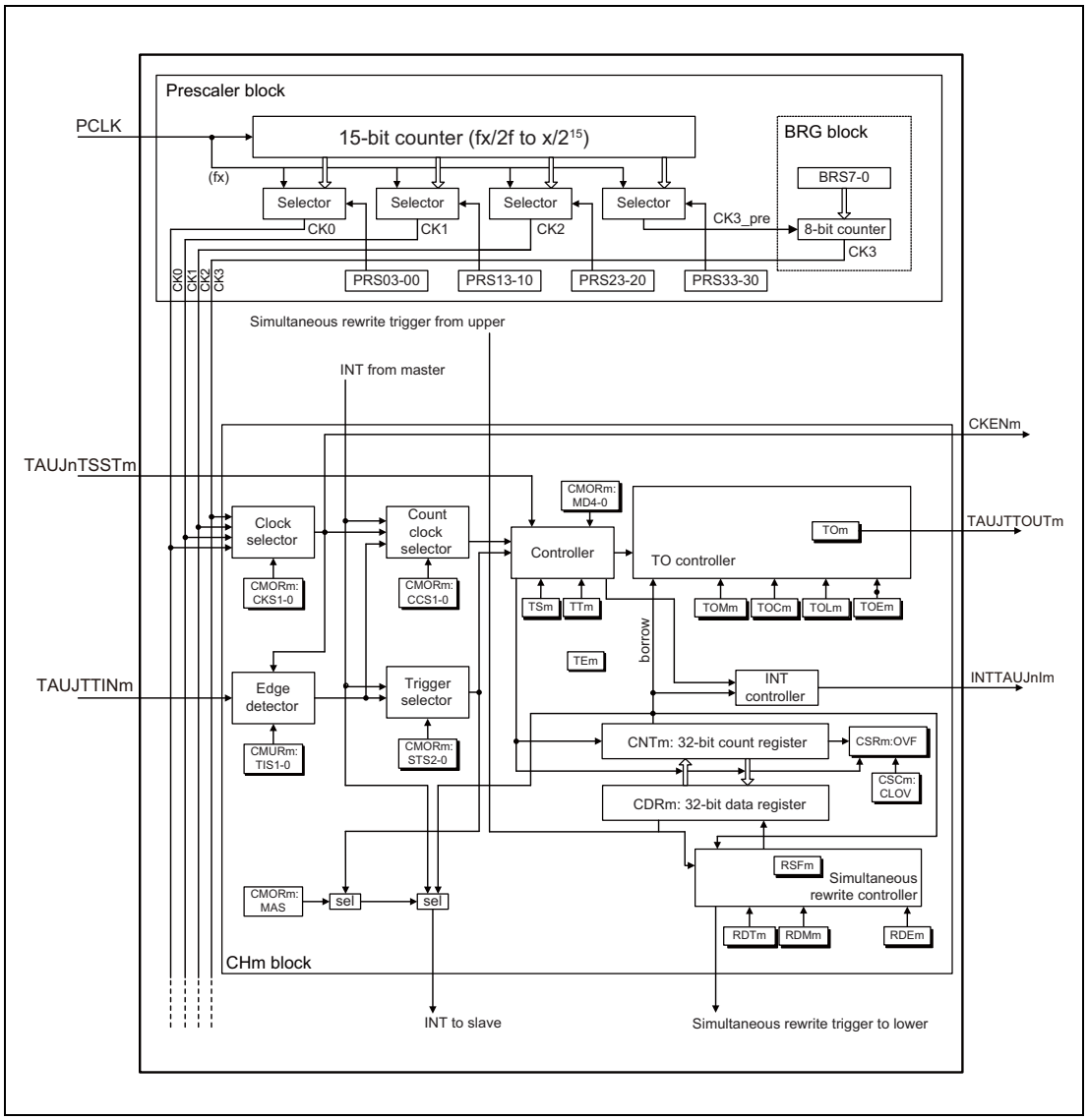


Figure 26.3 Block Diagram of the TAUJ

The prefix “TAUJn” has been omitted from the register names for the sake of clarity in the above figure.

26.2.6 Description of Block Diagram

The following describes the functional blocks.

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK2 are derived by dividing PCLK in the prescaler by a division factor of 2^0 to 2^{15}. The fourth count clock, CK3, is derived by dividing PCLK by a division factor that is not a power of 2 by using the baud rate generator.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source.

– One of CK0 to CK3 clocks (selected by the clock selector)

Controller

The controller controls the main operations of the counter.

- Operating mode (selected with the TAUJnCMORm.TAUJnMD[4:0] bits)
- Counter start enable (TAUJnTS.TAUJnTSm) and counter stop (TAUJnTT.TAUJnTTm)

When counter start is enabled, status flag TAUJnTE.TAUJnTEm is set.

Trigger selector

The counter starts automatically when it is enabled (TAUJnTE.TAUJnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

– Synchronous channel start trigger input TAUJnTSSTm
– Valid edge of the TAUJTTINm input signal
– INTTAUJnIm from the master channel

Simultaneous rewrite controller

Simultaneous rewrite control is enabled in synchronous operating modes. The data registers of all channels in a channel group (TAUJnCDRm) can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUJnTO controller

The output control of every channel enables the generation of various output signals such as PWM signals.

26.3 Registers

26.3.1 List of Registers

TAUJ registers are listed in the following table.

For details about <TAUJn_base>, see **Section 26.1.2, Register Base Address**.

Table 26.13 List of Registers

Module Name	Register Name	Symbol	Address
TAUJn prescaler registers			
TAUJn	TAUJn prescaler clock select register	TAUJnTPS	<TAUJn_base> + 90_H
TAUJn	TAUJn prescaler baud rate setting register	TAUJnBRS	<TAUJn_base> + 94_H
TAUJn control registers			
TAUJn	TAUJn channel data register m	TAUJnCDRm	<TAUJn_base> + m × 4_H
TAUJn	TAUJn channel counter register m	TAUJnCNTm	<TAUJn_base> + 10_H + m × 4_H
TAUJn	TAUJn channel mode OS register m	TAUJnCMORm	<TAUJn_base> + 80_H + m × 4_H
TAUJn	TAUJn channel mode user register m	TAUJnCMURm	<TAUJn_base> + 20_H + m × 4_H
TAUJn	TAUJn channel status register m	TAUJnCSRm	<TAUJn_base> + 30_H + m × 4_H
TAUJn	TAUJn channel status clear trigger register m	TAUJnCSCm	<TAUJn_base> + 40_H + m × 4_H
TAUJn	TAUJn channel start trigger register	TAUJnTS	<TAUJn_base> + 54_H
TAUJn	TAUJn channel enable status register	TAUJnTE	<TAUJn_base> + 50_H
TAUJn	TAUJn channel stop trigger register	TAUJnTT	<TAUJn_base> + 58_H
TAUJn output registers			
TAUJn	TAUJn channel output enable register	TAUJnTOE	<TAUJn_base> + 60_H
TAUJn	TAUJn channel output register	TAUJnTO	<TAUJn_base> + $5C_H$
TAUJn	TAUJn channel output mode register	TAUJnTOM	<TAUJn_base> + 98_H
TAUJn	TAUJn channel output configuration register	TAUJnTOC	<TAUJn_base> + $9C_H$
TAUJn	TAUJn channel output active level register	TAUJnTOL	<TAUJn_base> + 64_H
TAUJn reload data registers			
TAUJn	TAUJn channel reload data enable register	TAUJnRDE	<TAUJn_base> + $A0_H$
TAUJn	TAUJn channel reload data mode register	TAUJnRDM	<TAUJn_base> + $A4_H$
TAUJn	TAUJn channel reload data trigger register	TAUJnRDT	<TAUJn_base> + 68_H
TAUJn	TAUJn channel reload status register	TAUJnRSF	<TAUJn_base> + $6C_H$
TAUJn emulation register			
TAUJn	TAUJn emulation register	TAUJnEMU	<TAUJn_base> + $A8_H$

26.3.2 Details of TAUJn Prescaler Registers

26.3.2.1 TAUJnTPS — TAUJn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3_PRE for all channels of the PCLK prescalers. CK3 is generated by dividing CK3_PRE by the factor specified in TAUJnBRS.

Access: This register can be read or written in 16-bit units.

Address: <TAUJn_base> + 90_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnPRS3[3:0]				TAUJnPRS2[3:0]				TAUJnPRS1[3:0]				TAUJnPRS0[3:0]			
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.14 TAUJnTPS Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	TAUJnPRS3 [3:0]	Specifies a CK3_PRE clock. The CK3_PRE clock is an input clock of the BRG unit which supplies CK3 operation clocks to all channels.

TAUJnPRS3[3:0]	CK3_PRE clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK3 are stopped (TAUJnTE.TAUJnTEm = 0).

Table 26.14 TAUJnTPS Register Contents (2/3)

Bit Position	Bit Name	Function
11 to 8	TAUJnPRS2 [3:0]	Specifies a CK2 clock.

TAUJnPRS2[3:0]	CK2 clock
0000_B	$PCLK/2^0$
0001_B	$PCLK/2^1$
0010_B	$PCLK/2^2$
0011_B	$PCLK/2^3$
0100_B	$PCLK/2^4$
0101_B	$PCLK/2^5$
0110_B	$PCLK/2^6$
0111_B	$PCLK/2^7$
1000_B	$PCLK/2^8$
1001_B	$PCLK/2^9$
1010_B	$PCLK/2^{10}$
1011_B	$PCLK/2^{11}$
1100_B	$PCLK/2^{12}$
1101_B	$PCLK/2^{13}$
1110_B	$PCLK/2^{14}$
1111_B	$PCLK/2^{15}$

The above bits are rewritable only when all the counters using CK2 are stopped (TAUJnTE.TAUJnTEm = 0).

7 to 4	TAUJnPRS1 [3:0]	Specifies a CK1 clock.

TAUJnPRS1[3:0]	CK1 clock
0000_B	$PCLK/2^0$
0001_B	$PCLK/2^1$
0010_B	$PCLK/2^2$
0011_B	$PCLK/2^3$
0100_B	$PCLK/2^4$
0101_B	$PCLK/2^5$
0110_B	$PCLK/2^6$
0111_B	$PCLK/2^7$
1000_B	$PCLK/2^8$
1001_B	$PCLK/2^9$
1010_B	$PCLK/2^{10}$
1011_B	$PCLK/2^{11}$
1100_B	$PCLK/2^{12}$
1101_B	$PCLK/2^{13}$
1110_B	$PCLK/2^{14}$
1111_B	$PCLK/2^{15}$

The above bits are rewritable only when all the counters using CK1 are stopped (TAUJnTE.TAUJnTEm = 0).

Table 26.14 TAUJnTPS Register Contents (3/3)

Bit Position	Bit Name	Function
3 to 0	TAUJnPRS0 [3:0]	Specifies a CK0 clock.

TAUJnPRS0[3:0]	CK0 clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK0 are stopped (TAUJnTE.TAUJnTEm = 0).

NOTE

TAUJn clock input PCLK is defined in the first part of this section, **Section 26.1.3, Clock Supply**.

26.3.2.2 TAUJnBRS — TAUJn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUJnTPS. TAUJnPRS3[3:0].

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 94_H

Value after reset: 00_H

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

Table 26.15 TAUJnBRS Register Contents

Bit Position	Bit Name	Function
7 to 0	TAUJnBRS [7:0]	Specifies a CK3_PRE clock division factor for generating CK3.

TAUJnBRS[7:0]	CK3 clock
$0000\ 0000_B$	CK3_PRE / 1
$0000\ 0001_B$	CK3_PRE / 2
$0000\ 0010_B$	CK3_PRE / 3
$0000\ 0011_B$	CK3_PRE / 4
:	:
$1111\ 1110_B$	CK3_PRE / 255
$1111\ 1111_B$	CK3_PRE / 256

26.3.3 Details of TAUJn Control Registers

26.3.3.1 TAUJnCDRm — TAUJn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUJnCMORm.TAUJnMD[4:1].

Access: This register can be read or written in 32-bit units.
- When this register functions as a capture register, only reading is possible. Write operation is ignored.
- When this register functions as a compare register, reading and writing is possible.

Address: <TAUJn_base> + 0_H + m × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAUJnCDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.16 TAUJnCDRm Register Contents

Bit Position	Bit Name	Function
31 to 0	TAUJnCDR [31:0]	Data register for capture/compare values

26.3.3.2 TAUJnCNTm — TAUJn Channel Counter Register

This is a channel m counter register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <TAUJn_base> + 10_H + m × 4_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAUJnCNT[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 26.17 TAUJnCNTm Register Contents

Bit Position	Bit Name	Function
31 to 0	TAUJnCNT [31:0]	32-bit counter value

The read value depends on a counter, an operating mode change, or TAUJnTS.TAUJnTSm/ TAUJnTT.TAUJnTTm bit value.

The initial counter read value depends on the operating mode and how the counter is stopped.

- By a reset
- By a counter stop trigger (TAUJnTT.TAUJnTTm = 1)

The following table lists the initial counter read values after the counter is stopped (TAUJnTE.TAUJnTEm = 0) and re-enabled (TAUJnTS.TAUJnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUJnTS.TAUJnTSm = 1) with the counter waiting for a start trigger.

Table 26.18 TAUJnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUJnCNTm		
		Start Value*1	After Stop Trigger	After One Count
Interval timer mode	Count down	FFFF FFFF$_H$	Stop value	—
Capture mode	Count up	0000 0000$_H$	Stop value	—
One-count mode	Count down	FFFF FFFF$_H$	Stop value	Stop value
Capture and one-count mode	Count up	0000 0000$_H$	Stop value	Capture value + 1 (TAUJnCDRm)
Count capture mode	Count up	0000 0000$_H$	Stop value	—
Gate count mode	Count down	FFFF FFFF$_H$	Stop value	Stop value
Capture and gate count mode	Count up	0000 0000$_H$	Stop value	Stop value

Note 1. The value set for TAUJnCNTm when operating mode is changed after a reset is deasserted.

26.3.3.3 TAUJnCMORm — TAUJn Channel Mode OS Register

This register controls channel m operation.

Access: This register can be read or written in 16-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + 80_H + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 26.19 TAUJnCMORm Register Contents (1/3)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Selects an operation clock, which is used with the TAUJTTINm input edge detection circuit. Setting of TAUJnCMORm.TAUJnCCS[1:0] bits also allows the operation clock to serve as the TAUJnCNTm count clock.

TAUJn CKS1	TAUJn CKS0	Selection of Operation Clock
0	0	CK0
0	1	CK1
1	0	CK2
1	1	CK3

Bit Position	Bit Name	Function
13, 12	TAUJnCCS[1:0]	Selects a count clock for TAUJnCNTm counter.

TAUJn CCS1	TAUJn CCS0	Selection of Operation Clock
0	0	Operation clock specified by TAUJnCMORm.TAUJnCKS[1:0].
0	1	Setting prohibited
1	0	
1	1	

Bit Position	Bit Name	Function
11	TAUJnMAS	Specifies whether the channel is a master or slave channel during synchronous channel operation. 0: Slave 1: Master This bit setting is valid only for even channels (CHm_even). Odd channels (CHm-odd) are fixed to 0.

Table 26.19 TAUJnCMORm Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUJnSTS[2:0]</td><td>Selects an external start trigger.
<table>
<tr><th>TAUJn STS2</th><th>TAUJn STS1</th><th>TAUJn STS0</th><th>Selection of Operation Clock</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of TAUJTTINm input signal, which is specified by TAUJnCMURm.TAUJnTIS[1:0].</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of TAUJTTINm input signal is used as a start trigger and the reverse edge as a stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INT of master channel</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td></td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>TAUJnCOS[1:0]</td><td>Specifies the timing for updating capture register TAUJnCDRm and overflow flag TAUJnCSRm.TAUJnOVF of channel m.
These bits are only valid if channel m is for capture function (capture mode and capture & one-count mode).
<table>
<tr><th>TAUJn COS1</th><th>TAUJn COS0</th><th>TAUJnCDRm</th><th>TAUJnCSRm.TAUJnOVF</th></tr>
<tr><td>0</td><td>0</td><td>Updated when valid edge of TAUJTTINm input is detected.</td><td>Updated (cleared or set) when valid edge of TAUJTTINm input is detected.
• Set TAUJnCSRm.TAUJnOVF if a counter overflow has occurred since the last valid edge was detected.
• Clear TAUJnCSRm.TAUJnOVF if no counter overflow has occurred since the last valid edge was detected.</td></tr>
<tr><td>0</td><td>1</td><td></td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated when valid edge of TAUJTTINm input is detected and when a counter overflow occurs.
• Detection of valid edge of TAUJTTINm input: The counter value is written into TAUJnCDRm.
• Occurrence of overflow: FFFF FFFF$_H$ is loaded into TAUJnCDRm. Detection of the next valid edge of TAUJTTINm is ignored.</td><td>No setting</td></tr>
<tr><td>1</td><td>1</td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr>
</table></td></tr>
<tr><td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing, write the value after reset.</td></tr>
</table>

Table 26.19 TAUJnCMORm Register Contents (3/3)

Bit Position	Bit Name	Function
4 to 0	TAUJnMD[4:0]	Specifies an operating mode.

TAUJn MD4	TAUJn MD3	TAUJn MD2	TAUJn MD1	TAUJn MD0	Functional Description
0	0	0	0	1/0	Interval timer mode
0	0	0	1	1/0	Setting prohibited
0	0	1	0	1/0	Capture mode
0	0	1	1	0	Setting prohibited
0	1	0	0	1/0	One-count mode
0	1	0	1	1/0	Setting prohibited
0	1	1	0	0	Capture and one-count mode
0	1	1	1	1/0	Setting prohibited
1	0	0	0	0	Setting prohibited
1	0	0	1	0	Setting prohibited
1	0	1	0	1/0	Setting prohibited
1	0	1	1	1/0	Count capture mode
1	1	0	0	0	Gate count mode
1	1	0	1	0	Capture and gate count mode

Mode	Role of TAUJnMD0 Bit
Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUJnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUJnIm is not generated. 1: INTTAUJnIm is generated.
One-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. **CAUTION** In one-count mode, INTTAUJnIm signal is not output at the beginning of count operation.
Capture and one-count mode Gate count mode Capture and gate count mode	This bit should be set to 0. **CAUTION** INTTAUJnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.

26.3.3.4 TAUJnCMURm — TAUJn Channel Mode User Register

This register specifies a type of valid edge detection used for TAUJTTINm input.

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 20_H + m × 4_H

Value after reset: 00_H

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

Table 26.20 TAUJnCMURm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	Specifies a valid edge of TAUJTTINm input signal.

TAUJn TIS1	TAUJn TIS0	Functional Description
0	0	Falling edge
0	1	Rising edge
1	0	Detection of falling and rising edges (selection of low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge
1	1	Detection of falling and rising edges (selection of high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge

Edge detection of TAUJTTINm input signal is based on the operation clock selected by TAUJnCMORm.TAUJnCKS[1:0].

26.3.3.5 TAUJnCSRm — TAUJn Channel Status Register

This register indicates the overflow status of channel m.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUJn_base> + 30_H + m × 4_H

Value after reset: $0x_H$

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

Table 26.21 TAUJnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	Reserved	When read, an undefined value is returned.
0	TAUJnOVF	Indicates the counter overflow status: 0: No overflow occurs 1: Overflow occurs This bit is used only in the following modes: • Capture mode • Capture and one-count mode The function of this bit depends on the setting of control bits TAUJnCMORm.TAUJnCOS[1:0].

26.3.3.6 TAUJnCSCm — TAUJn Channel Status Clear Trigger Register

This register is a trigger register for clearing the overflow flag TAUJnCSRm.TAUJnOVF of channel m.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 40_H + m × 4_H

Value after reset: 00_H

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

Table 26.22 TAUJnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAUJnCLOV	0: No function 1: Clears the overflow flag TAUJnCSRm.TAUJnOVF

26.3.3.7 TAUJnTS — TAUJn Channel Start Trigger Register

This register enables the counter operation for each channel.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 54_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTS03	TAUJnTS02	TAUJnTS01	TAUJnTS00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 26.23 TAUJnTS Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnTSm	Enables the counter operation for channel m: 0: No function 1: Enables the counter operation and sets TAUJnTE.TAUJnTEm = 1. Only the counter operation is enabled even if TAUJnTE.TAUJnTEm = 1. Whether the counter is started or not depends on the selected operating mode.

26.3.3.8 TAUJnTE — TAUJn Channel Enable Status Register

This register indicates whether a counter operation is enabled.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUJn_base> + 50_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTE03	TAUJnTE02	TAUJnTE01	TAUJnTE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 26.24 TAUJnTE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	TAUJnTEm	Indicates whether channel m's counter operation is enabled. 0: Counter operation is disabled 1: Counter operation is enabled This bit is set to 1 when trigger input of TAUJnTSSTm (synchronous channel start trigger signal) is detected or when TAUJnTS.TAUJnTSm is set to 1. This bit is reset to 0 when TAUJnTT.TAUJnTTm is set to 1.

26.3.3.9 TAUJnTT — TAUJn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 58_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTT03	TAUJnTT02	TAUJnTT01	TAUJnTT00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 26.25 TAUJnTT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnTTm	Stops channel m's counter operation. 0: No function 1: Stops the counter operation and resets TAUJnTE.TAUJnTEm. TAUJnCNTm, TAUJnTO.TAUJnTOm, and TAUJTTOUTm retain the values provided before the counter is stopped.

26.3.4 Details of TAUJn Simultaneous Rewrite Register

26.3.4.1 TAUJnRDE — TAUJn Channel Reload Data Enable Register

This register enables and disables simultaneous rewrite of the data register TAUJnCDRm. It also enables and disables simultaneous rewrite of the data register TAUJnTOLm for the PWM output function.

Access: This register can be read or written in 8-bit units. It can only be written when TAUJnTE.TAUJnTEm = 0.

Address: <TAUJn_base> + $A0_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDE03	TAUJnRDE02	TAUJnRDE01	TAUJnRDE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 26.26 TAUJnRDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnRDEm	Enables/disables simultaneous rewrite of the data register of channel m: 0: Disables simultaneous rewrite 1: Enabled simultaneous rewrite

26.3.4.2 TAUJnRDM — TAUJn Channel Reload Data Mode Register

This register selects when the signal that controls simultaneous rewrite is generated.

Access: This register can be read or written in 8-bit units. It can only be written when TAUJnTE.TAUJnTEm = 0.

Address: <TAUJn_base> + $A4_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDM03	TAUJnRDM02	TAUJnRDM01	TAUJnRDM00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 26.27 TAUJnRDM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnRDMm	Specifies when the signal that triggers simultaneous rewrite is generated: 0: When the master channel counter starts counting 1: No function These bits only apply when TAUJnRDE.TAUJnRDEm = 1.

26.3.4.3 TAUJnRDT — TAUJn Channel Reload Data Trigger Register

This register triggers the simultaneous rewrite enabling state.

Access: This register is a write-only register that can be written in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 68_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDT03	TAUJnRDT02	TAUJnRDT01	TAUJnRDT00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 26.28 TAUJnRDT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3 to 0	TAUJnRDTm	Triggers the simultaneous rewrite enabling state. 0: No function 1: Simultaneous rewrite enabling state is triggered. The simultaneous rewrite enabling flag (TAUJnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger. These bits only apply when: • TAUJnRDE.TAUJnRDEm = 1

26.3.4.4 TAUJnRSF — TAUJn Channel Reload Status Register

This flag register indicates the simultaneous rewrite status.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <TAUJn_base> + $6C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRSF03	TAUJnRSF02	TAUJnRSF01	TAUJnRSF00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 26.29 TAUJnRSF Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	TAUJnRSFm	Indicates the simultaneous rewrite status. 0: Indicates that simultaneous rewrite has been completed due to the generation of simultaneous rewrite trigger. 1: Indicates that the system waits for a simultaneous rewrite trigger in the simultaneous rewrite enabling state (TAUJnRDFm = 1).

26.3.5 Details of TAUJn Output Registers

26.3.5.1 TAUJnTOE — TAUJn Channel Output Enable Register

This register enables and disables independent channel output mode controlled by software.

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 60_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOE03	TAUJnTOE02	TAUJnTOE01	TAUJnTOE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 26.30 TAUJnTOE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOEm	Enables/disables independent channel output function: 0: Disables independent timer output function (controlled by software) 1: Enables independent timer output function

26.3.5.2 TAUJnTO — TAUJn Channel Output Register

This register specifies and reads the level of TAUJTTOUTm.

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + $5C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTO03	TAUJnTO02	TAUJnTO01	TAUJnTO00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 26.31 TAUJnTO Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOm	Specifies and reads the level of TAUJTTOUTm: 0: Low 1: High Only TAUJnTOm bits for which Independent Channel Output function is disabled (TAUJnTOEm = 0) can be written.

26.3.5.3 TAUJnTOM — TAUJn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read or written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + 98_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOM03	TAUJnTOM02	TAUJnTOM01	TAUJnTOM00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 26.32 TAUJnTOM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOMm	Specifies the channel output mode: 0: Independent channel output mode 1: Synchronous channel output mode The output mode depends on the settings of channel output control (TAUJnTOE.TAUJnTOEm) bits.

26.3.5.4 TAUJnTOC — TAUJn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUJnTOMm.

Access: This register can be read or written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + $9C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOC03	TAUJnTOC02	TAUJnTOC01	TAUJnTOC00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 26.33 TAUJnTOC Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 26.33 TAUJnTOC Register Contents

Bit Position	Bit Name	Function
3 to 0	TAUJnTOCm	Specifies the output mode: 0: Operation mode 1 (= Toggle mode) 1: No function This bit must be set to 0 for all output modes except independent channel output mode controlled by software. The output mode also depends on TAUJnTOM.TAUJnTOMm, as shown in the following table.

TAUJn TOMm	TAUJn TOCm	Functional Description
0	0	Toggle mode: Toggling proceeds when INTTAUJnIm occurs.
0	1	No function
1	0	Synchronous channel operation mode 1: Set when INT occurs on the master channel and reset when INT occurs on the slave channel.
1	1	No function

26.3.5.5 TAUJnTOL — TAUJn Channel Output Level Register

This register specifies the output logic of the channel output bit (TAUJnTO.TAUJnTOm).

Access: This register can be read or written in 8-bit units.

Address: <TAUJn_base> + 64_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOL03	TAUJnTOL02	TAUJnTOL01	TAUJnTOL00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 26.34 TAUJnTOL Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	TAUJnTOLm	Specifies the output logic of the channel m output bit (TAUJnTO.TAUJnTOm): 0: Positive logic (active high) 1: Negative logic (active low) These bits apply in all channel output modes except independent channel output mode controlled by software and independent channel output mode 1.

26.3.5.6 TAUJnEMU — TAUJn Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read or written in 8-bit units.
A write should be performed when counters are stopped (TAUJnTE.TAUJnTEm = 0) and EPC.SVSTOP = 0.

Address: <TAUJn_base> + $A8_H$

Value after reset: 00_H

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

Table 26.35 TAUJnEMU Register Contents

Bit Position	Bit Name	Function
7	TAUJnSVSDIS	When EPC.SVSTOP bit = 0: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint), regardless of the value of this bit (1 or 0). When EPC.SVSTOP bit = 1: 0: The count clock is stopped when the debugger takes control of the microcontroller (as in the breakpoint). 1: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

26.4 Operating Procedure

The following lists the general operation procedure for the TAUJn:

After a reset is deasserted, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUJTTOUTm is also initialized and outputs a low level.

1. Set the TAUJnTPS and TAUJnBRS registers to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUJn function:
   - Set the operation mode
   - Set any other control bits
3. Enable the counter by setting the TAUJnTS.TAUJnTSm bit to 1.
   The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.
4. If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUJnTT.TAUJnTTm bit to 1. The counter can be forcibly restarted by setting the TAUJnTS.TAUJnTSm bit to 1.
5. Stop the function by setting the TAUJnTT.TAUJnTTm bit to 1.

NOTES

1. A detailed description of the required control bits and the operation of the individual functions are given in **Section 26.12, Independent Channel Operation Functions** and **Section 26.13, Synchronous Channel Operation Functions**.
2. The function can be changed while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

26.5 Concepts of Synchronous Channel Operation Function

The synchronous channel operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

These rules are detailed in **Section 26.5.1, Rules of Synchronous Channel Operation Function**.

The synchronous channel operation function are detailed in the following section.

- **Section 26.13, Synchronous Channel Operation Functions**

26.5.1 Rules of Synchronous Channel Operation Function

Number of master and slave channels

- Only even channels (CH0, CH2) can be set as master channels. Any channel other than CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and multiple slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 can be set as slave channel.
- If two master channels are used, slave channels cannot cross the master.
  Example: If CH0 and CH2 are master channels, CH1 can be set as slave channel for CH0, but CH3 cannot.

Operation clock

- The same operation clock should be set for the master channel and the synchronized slave channels. This is achieved by setting the same value in the TAUJnCMORm.TAUJnCKS[1:0] bits of the master and slave channels.

The basic concepts of master/slave usage and operation clocks are illustrated in **Figure 26.4**.

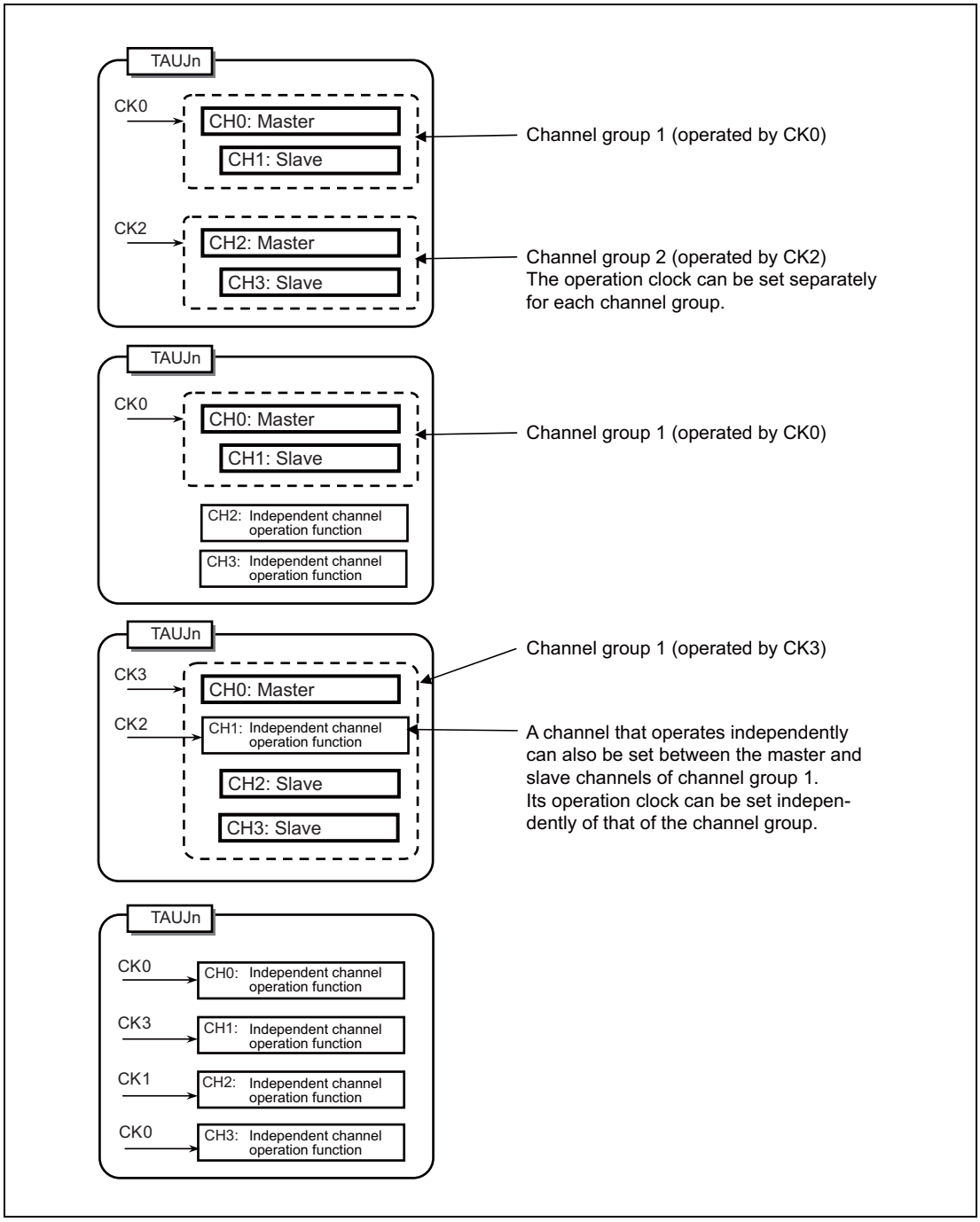


Figure 26.4 Grouping of Channels and Assignment of Operation Clocks

26.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously, both within a TAUJ unit and between TAUJ units.

26.5.2.1 Simultaneous Start and Stop within a TAUJ Unit

- To simultaneously start synchronized channels, the TAUJnTS.TAUJnTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUJnTT.TAUJnTTm bits of the channels should be set at the same time.

Setting 1 in the TAUJnTS.TAUJnTSm bits sets the corresponding TAUJnTE.TAUJnTEm bits to 1, enabling counting. The count start timing of the counter depends on the operating mode.

26.5.2.2 Simultaneous Start between TAUJ Units

Counters in different TAUJ units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

26.6 Simultaneous Rewrite

26.6.1 How to Control Simultaneous Rewrite

The following figure shows the general procedure for simultaneous rewrite. The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

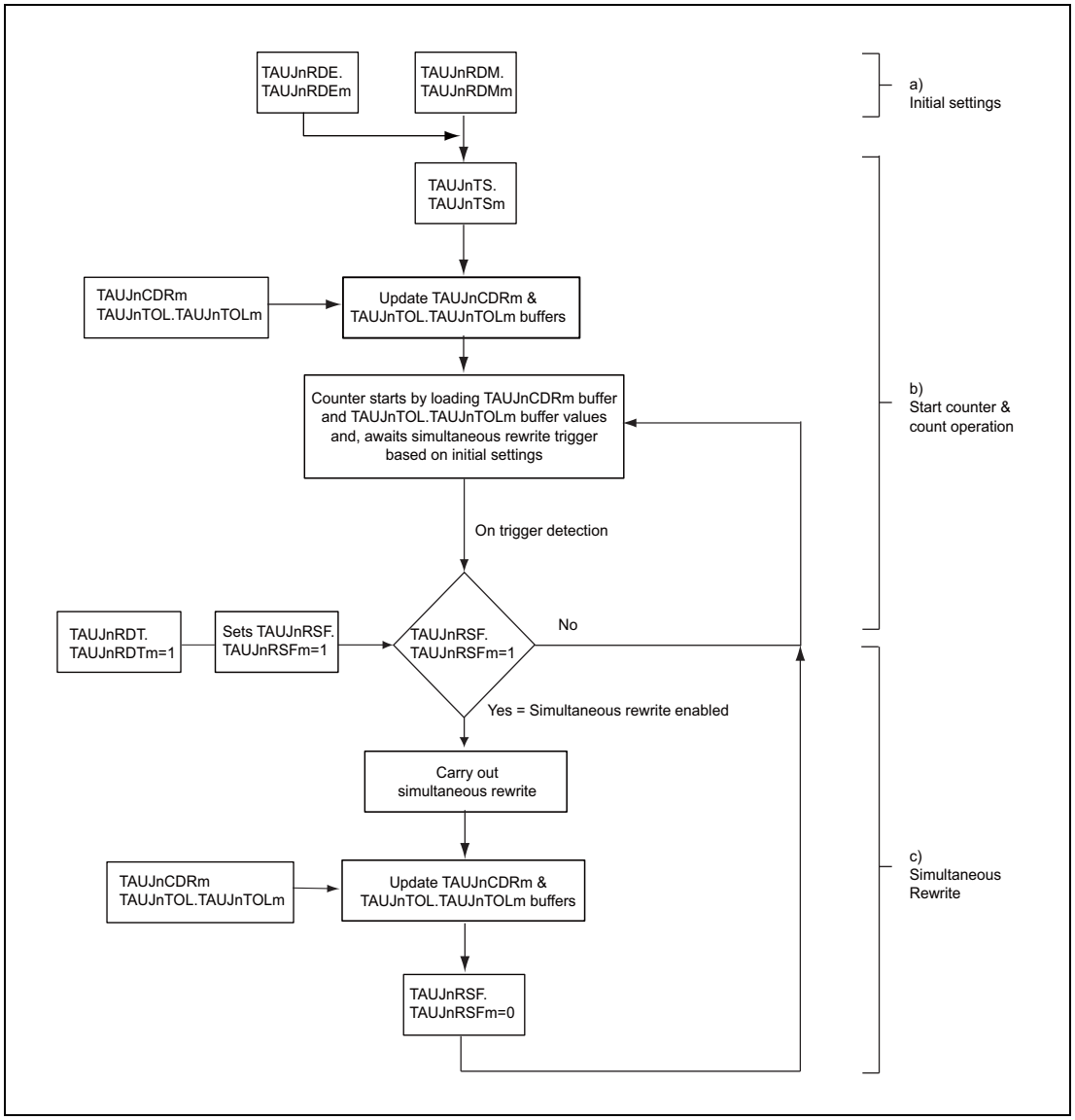


Figure 26.5 General Procedure for Simultaneous Rewrite

26.6.1.1 Initial Settings

- To enable simultaneous rewrite in channel m, set TAUJnRDE.TAUJnRDEm = 1.
- To select simultaneous rewrite when the master channel starts counting, set TAUJnRDM.TAUJnRDMm.

26.6.1.2 Start Counter and Count Operation

- To start all the TAUJnCNTm counters in the channel group, set the corresponding TAUJnTS.TAUJnTSm bits to 1. The values of TAUJnTOL.TAUJnTOLm and the data registers (TAUJnCDRm) are loaded into the corresponding TAUJnTOL.TAUJnTOLm buffer (TAUJnTOL.TAUJnTOLm buf) and data buffer registers (TAUJnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUJnRDT.TAUJnRDTm) to 1 sets the reload flag (TAUJnRSF.TAUJnRSFm) to 1, enabling simultaneous rewrite. TAUJnRSF.TAUJnRSFm remains set to 1 until simultaneous rewrite is completed.
- When a specified trigger for simultaneous rewrite is detected, the TAUJnRSF.TAUJnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1). If enabled, simultaneous rewrite is carried out. Otherwise simultaneous rewrite is not carried out and the system waits for detection of the next simultaneous rewrite trigger.

26.6.1.3 Simultaneous Rewrite

- When the simultaneous rewrite trigger is detected and simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1), the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and the values are applied the next time the counter starts or restarts.
- When the simultaneous rewrite is completed, the TAUJnRSF.TAUJnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

26.6.2 Other General Rules for Simultaneous Rewrite

The following rules also apply.

- TAUJnRDE.TAUJnRDEm and TAUJnRDM.TAUJnRDMm cannot be changed while the counter is in operation (TAUJnTE.TAUJnTEm = 1).
- TAUJnTOL.TAUJnTOLm can be rewritten only during operation using the PWM output function. For all other functions, TAUJnTOL.TAUJnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUJTTOUTm outputs an invalid waveform.

26.6.3 Simultaneous Rewrite Procedure

The simultaneous rewrite procedure with PWM output function is described in the following figure.

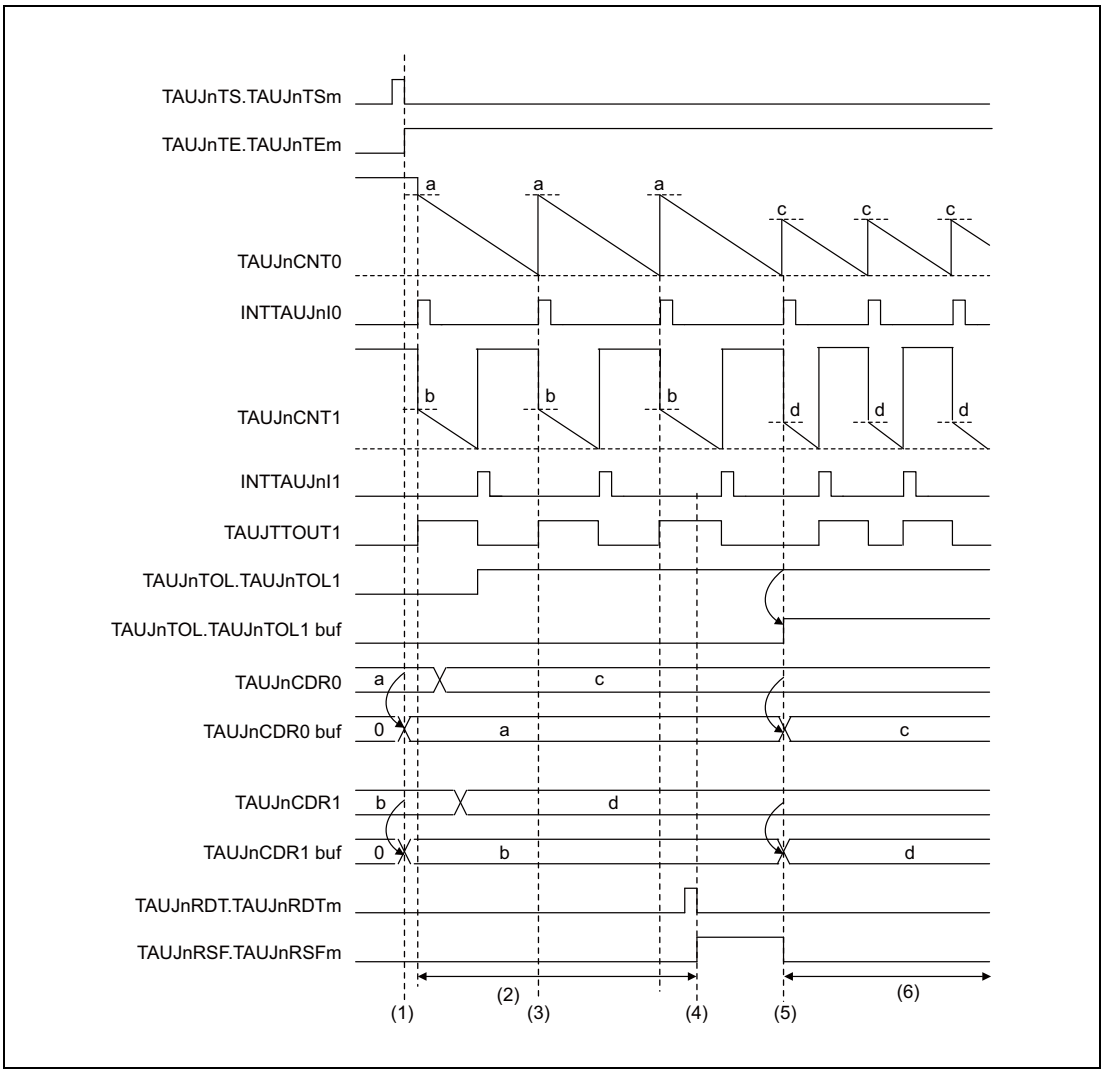


Figure 26.6 Simultaneous Rewrite with PWM Output Function

Setting:

CH0 is a master channel of PWM output function, and CH1 is a slave channel of PWM output function. Simultaneous rewrite is applied when the master channel starts counting.

Description:

(1) When TAUJnTS.TAUJnTSm = 1 is set, the value of TAUJnCDRm is copied to the TAUJnCDRm buffer and the value of TAUJnTOL.TAUJnTOLm is copied to the TAUJnTOL.TAUJnTOLm buffer.

(2) The TAUJnCDRm and TAUJnTOL.TAUJnTOLm registers can be written at any time.

(3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUJnRSF.TAUJnRSFm = 0).

(4) The reload data trigger bit (TAUJnRDT.TAUJnRDTm) is set to 1 which sets the status flag (TAUJnRSF.TAUJnRSFm = 1), enabling simultaneous rewrite.

(5) Simultaneous rewrite is triggered when CH0 restarts counting, because simultaneous rewrite is enabled. The TAUJnCDRm value is loaded into the TAUJnCDRm buffer and the TAUJnTOL.TAUJnTOLm value is loaded into the TAUJnTOL.TAUJnTOLm buffer.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUJnCDRm and TAUJnTOL.TAUJnTOLm can be changed again.

26.7 Channel Output Modes

The output of the TAUJTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUJnTOE.TAUJnTOEm = 0)
  When controlled by software, the value written in the output register bit (TAUJnTO.TAUJnTOm) is sent to the output pin (TAUJTTOUTm).
- By TAUJ signals (TAUJnTOE.TAUJnTOEm = 1)
  When controlled by TAUJ signals, the output level of TAUJTTOUTm is set or reset or toggled by internal signals. The value of TAUJnTO.TAUJnTOm is updated accordingly to reflect the value of TAUJTTOUTm.
  - Independently (TAUJnTOM.TAUJnTOMm = 0)
    In case of independent operation, the output of the TAUJTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUJnTOM.TAUJnTOMm = 0).
  - Synchronously (TAUJnTOM.TAUJnTOMm = 1)
    In case of synchronous operation, the output of the TAUJTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUJnTOM.TAUJnTOMm = 1).

The TAUJnTO.TAUJnTOm bit can always be read to determine the current value of TAUJTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in **Table 26.36, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 26.7.2, Channel Output Modes Controlled Independently by TAUJn Signals**
- **Section 26.7.3, Channel Output Modes Controlled Synchronously by TAUJn Signals**

Batch operation of TAUJnTOm bit

Whether a set value is reflected to the TAUJnTOm bit or not is controlled by the TAUJnTOE.TAUJnTOEm bit.

The TAUJnTOm setting is written only to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 0 when a write to the TAUJnTO register is attempted. No TAUJnTOm setting is reflected to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 1.

NOTE

TAUJnTO.TAUJnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or negative logic of the output is specified by control bit TAUJnTOL.TAUJnTOLm.

The value of TAUJnTOL.TAUJnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function. Otherwise, changes to TAUJnTOL.TAUJnTOLm result in an undefined TAUJTTOUTm signal output.

See **Section 26.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 26.36**.

Table 26.36 Channel Output Modes

Channel Output Mode	TAUJnTOE.TAUJnTOEm	TAUJnTOM.TAUJnTOMm
By software		
Independent channel output mode controlled by software	0	x
By TAUJ signals, independently		
Independent channel output mode 1	1	0
By TAUJ signals, synchronously		
Synchronous channel output mode 1	1	1

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

NOTE

The following bits cannot be changed during count operation (TAUJnTE.TAUJnTEm = 1):

- TAUJnTOM.TAUJnTOMm
- TAUJnTOC.TAUJnTOCm

26.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUJTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUJnTOE.TAUJnTOEm = 0).

(1) Set TAUJnTO.TAUJnTOm to specify the initial level of the TAUJTTOUTm output.

(2) Set channel output mode according to **Table 26.36, Channel Output Modes**, and the output logic using the TAUJnTOL.TAUJnTOLm bit.

(3) Start the counter (TAUJnTS.TAUJnTSm = 1).

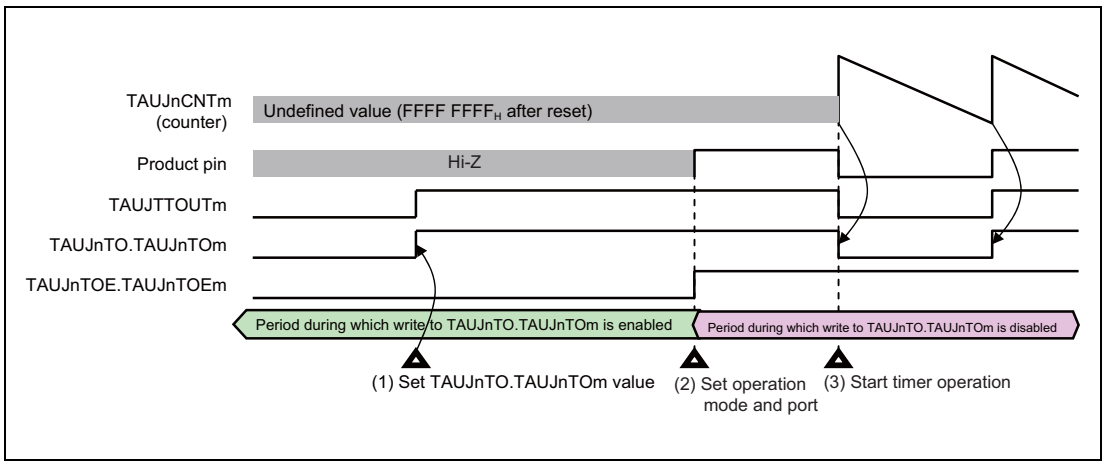


Figure 26.7 General Procedure for Specifying a TAUJTTOUTm Channel Output Mode

26.7.2 Channel Output Modes Controlled Independently by TAUJn Signals

This section lists the channel output modes that are controlled independently by TAUJn signals. The control bits used to specify a mode are listed in **Table 26.36, Channel Output Modes**.

26.7.2.1 Independent Channel Output Mode 1

Set/reset conditions

In this output mode, TAUJTTOUTm toggles when INTTAUJnIm is detected. The value of TAUJnTOL.TAUJnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 26.36, Channel Output Modes**.

26.7.3 Channel Output Modes Controlled Synchronously by TAUJn Signals

This section lists the channel output modes that are controlled synchronously by TAUJn signals. The control bits used to specify a mode are listed in **Table 26.36, Channel Output Modes**.

26.7.3.1 Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUJnIm of master channel serves as a set signal and INTTAUJnIm of the slave channel as a reset signal. If INTTAUJnIm of the master channel, and INTTAUJnIm of the slave channel are generated at the same time, INTTAUJnIm of the slave channel (reset signal) has priority over INTTAUJnIm (set signal) of the master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 26.36, Channel Output Modes**.

26.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUJnTS.TAUJnTSm is set to 1 in each operating mode.

In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

26.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-Up/-Down Mode, and Count Capture Mode

The counter starts operating with the next count clock cycle after TAUJnTS.TAUJnTSm is set to 1. The value of data register is also loaded when the counter starts.

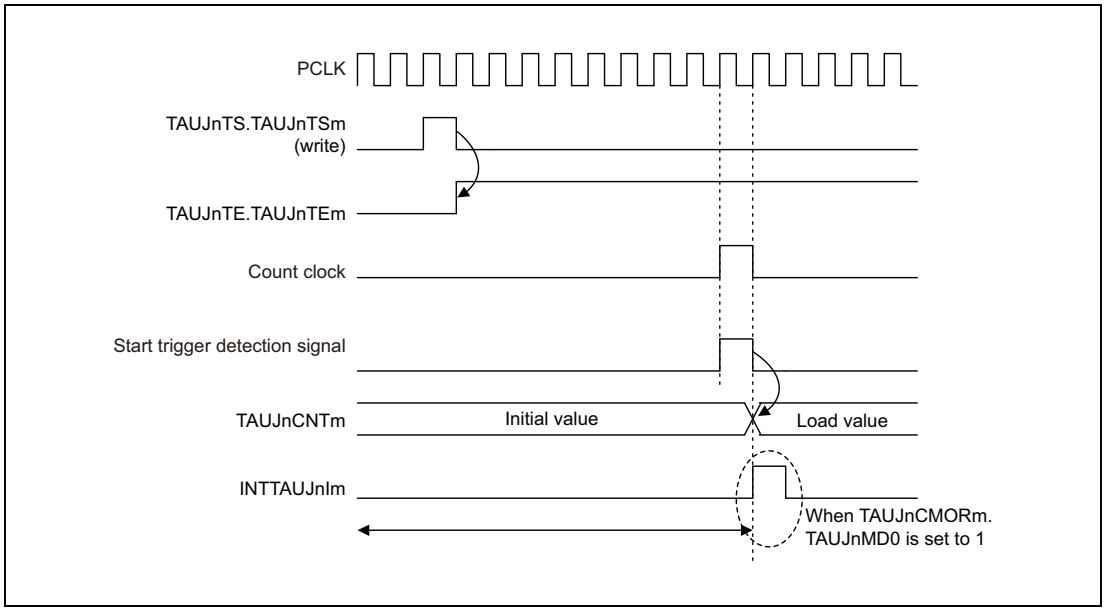


Figure 26.8 Start Timing in Interval Timer Mode, Capture Mode, and Count Capture Mode

26.8.2 Other Operating Modes

In other operating modes, count clock cycle is irrelevant to start of counter operation. The counter operation start timing is triggered only upon detection of a valid edge of TAUJTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which is irrelevant to start of counter operation, determine the frequency with which all operations take place.

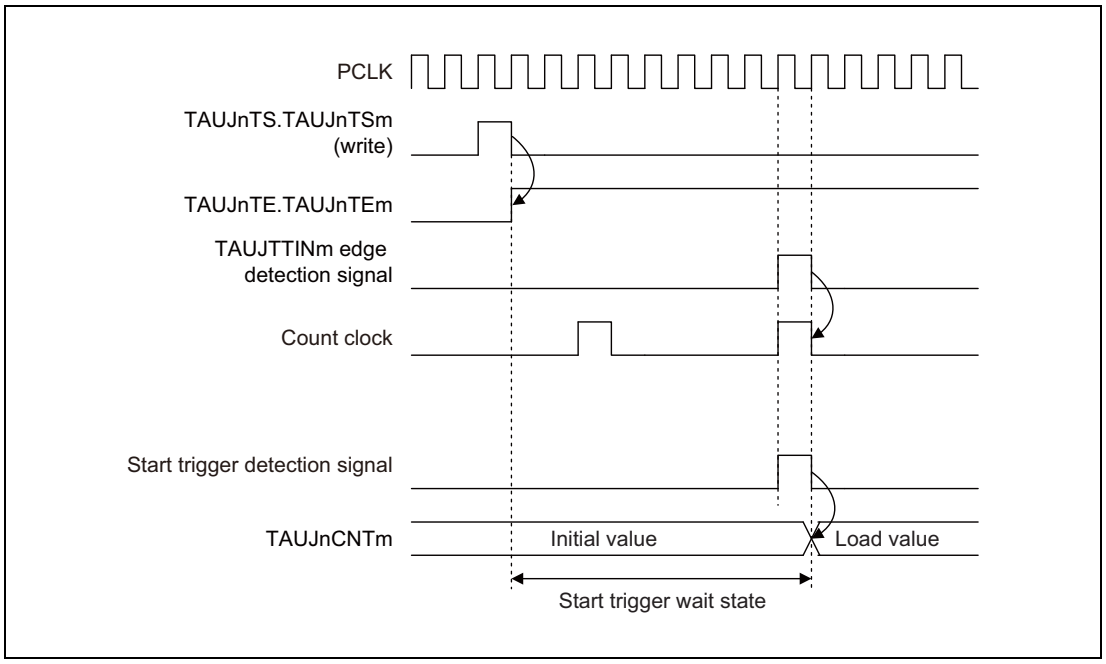


Figure 26.9 Count Start Timing in Other Operating Modes

26.9 TAUJTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUJnIm is generated using the TAUJnCMORm.TAUJnMD0 bit. The generation of INTTAUJnIm when the MD0 bit starts counting and the effect to TAUJTTOUTm depend on the selected function. For details, refer to the description of TAUJnCMORm.TAUJnMD0 of each function.

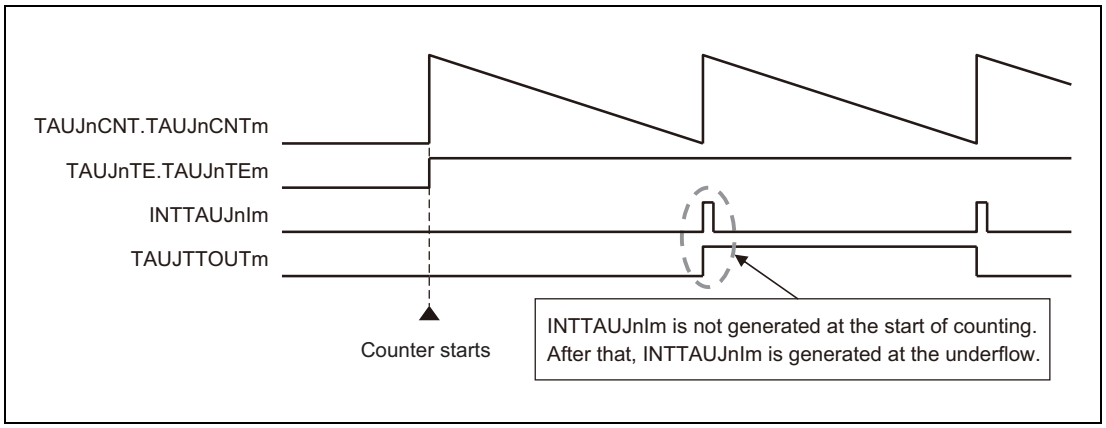


Figure 26.10 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJMD0 = 0)

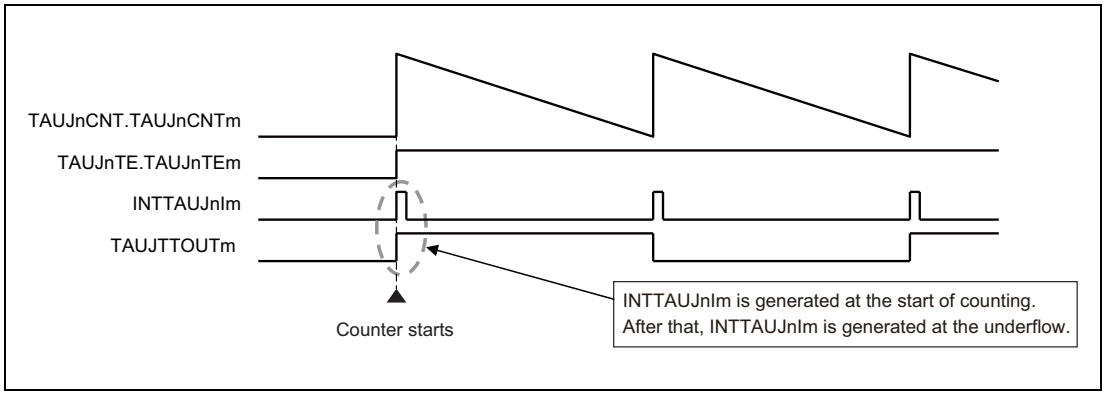


Figure 26.11 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJMD0 = 1)

26.10 Interrupt Generation upon Overflow

In certain independent functions, an interrupt is not generated when the counter value reaches FFFF FFFF$_H$ and an overflow occurs during count-up. This section describes how to generate an interrupt by combining channel operations in a mode that counts up and in a mode that counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000 0000$_H$ at the same time as the first channel overflows (TAUJnCNTm = FFFF FFFF$_H$).
- Set TAUJnCDRm of the second channel to FFFF FFFF$_H$.
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same TAUJTTINm input.
- The trigger detection settings (TAUJnCMORm.TAUJnSTS[2:0] and TAUJnCMURm.TAUJnTIS[1:0]) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000 0000$_H$ at exactly the same time as the up-counter of the first channel overflows (TAUJnCNTm = FFFF FFFF$_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

26.10.1 Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function

When the capture trigger is input simultaneously to TAUJTTINm of both channels, INTTAUJnIm of the interval timer function can detect the overflow when TAUJnCNTm of the TAUJTTINm input position detection function exceeds FFFF $FFFF_H$.

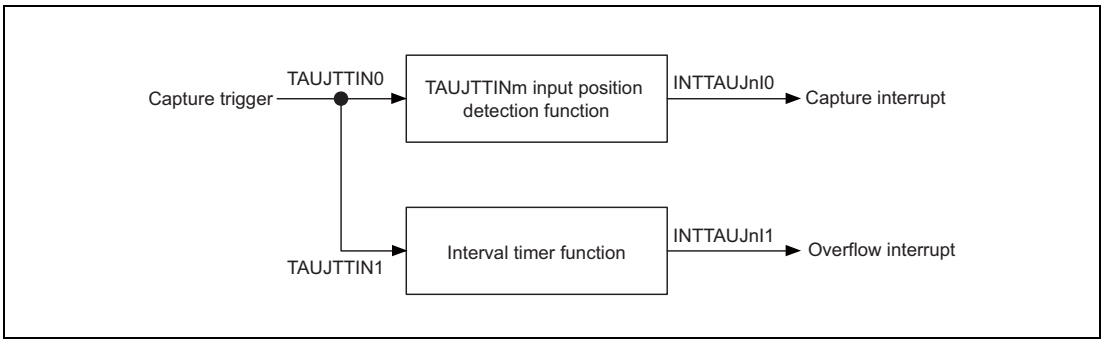


Figure 26.12 Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function

Timing diagram

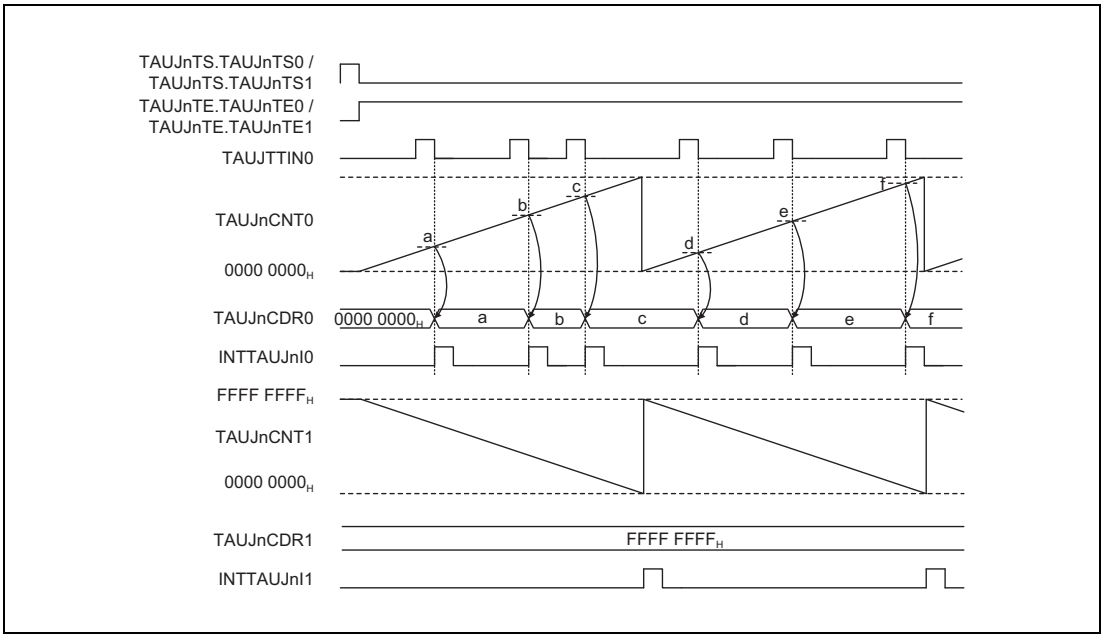


Figure 26.13 Interrupt Generation via Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function

26.11 TAUJTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

The following figure shows when edge detection takes place.

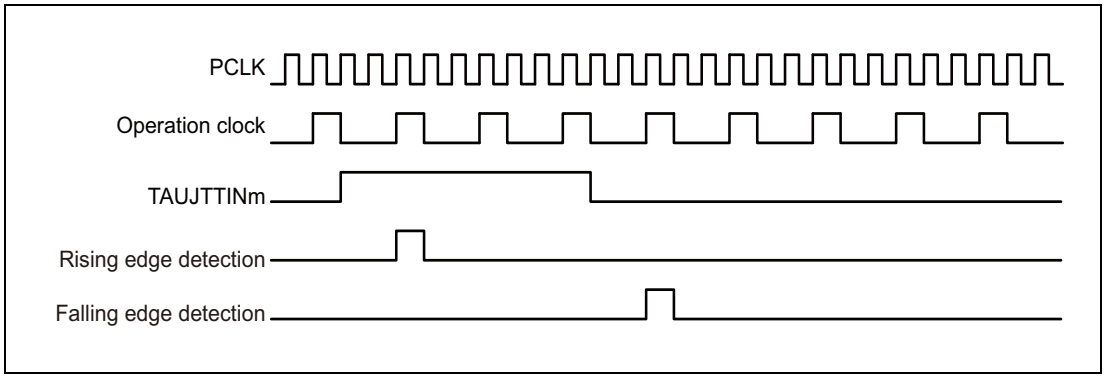


Figure 26.14 Basic Edge Detection Timing

Figure 26.14 shows an image of operation timing. In the actual operation, delay time occurs due to noise filter and synchronization circuit between the TAUJnIm pin and TAUJn.

26.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the TAUJ. For a general overview of independent channel operation functions, see **Section 26.2, Overview**.

26.12.1 Interval Timer Function

26.12.1.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals. When an interrupt is generated, the TAUJTTOUTm signal toggles, resulting in a square wave.

Functional description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counter starts to count down from this value.

When the counter reaches $0000\ 0000_H$, INTTAUJnIm is generated and the TAUJTTOUTm signal toggles. TAUJnCNTm then loads the TAUJnCDRm value and subsequently continues operation.

The value of TAUJnCDRm can be rewritten at any time, and the changed value of TAUJnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJTTOUTm stop but retain their values. The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.

Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUJTTOUTm does not toggle. This results in a reverted TAUJTTOUTm signal compared to when TAUJnCMORm.TAUJnMD0 is set to 1.

26.12.1.2 Equations

INTTAUJnIm cycle = count clock cycle × (TAUJnCDRm + 1)

TAUJTTOUTm square wave cycle = count clock cycle × (TAUJnCDRm+1) × 2

26.12.1.3 Block Diagram and General Timing Diagram

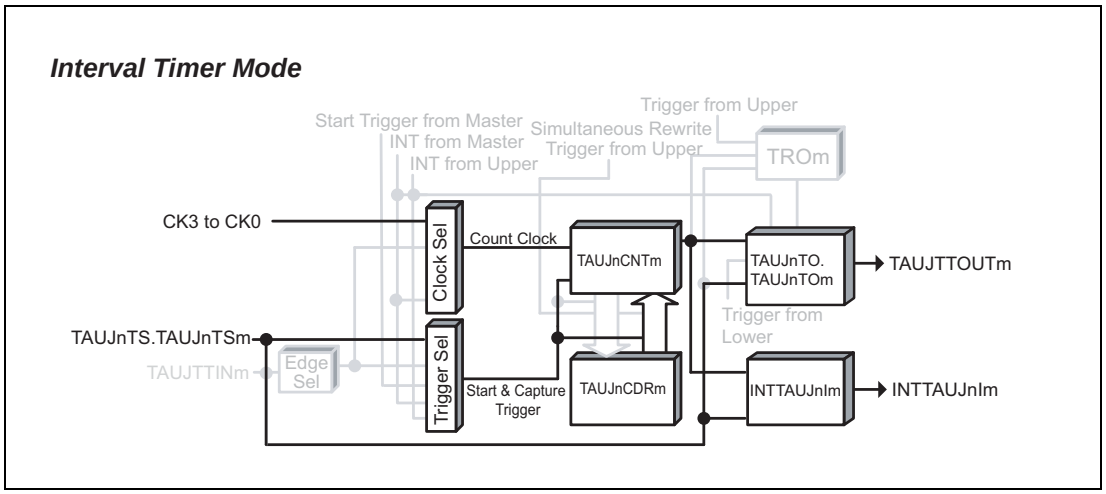


Figure 26.15 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1).

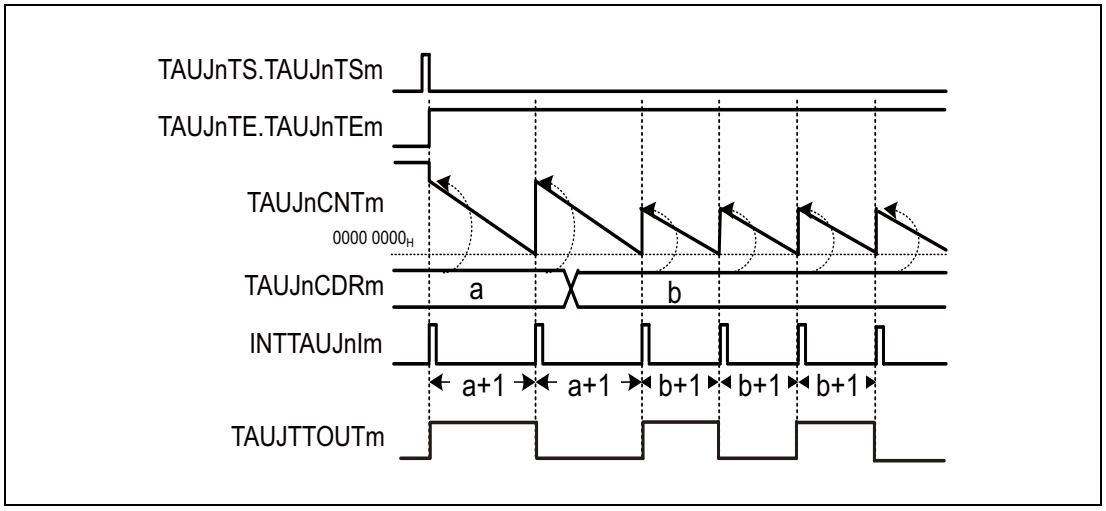


Figure 26.16 General Timing Diagram for Interval Timer Function

26.12.1.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 26.37 Contents of the TAUJnCMORm register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 000_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated and TAUJTTOUTm does not toggle when operation starts or restarts. 1: Generates INTTAUJnIm and toggles TAUJTTOUTm when operation starts or restarts.

(2) TAUJnCMURm

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

Table 26.38 Contents of the TAUJnCMURm register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode

Table 26.39 Control Bit Settings in Independent Channel Output Mode 1

Bit name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 0_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	Write 0_B.

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJTTOUTm can then be controlled independently of the interrupts. For details refer to **Section 26.7, Channel Output Modes**.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the interval timer function. Therefore, these registers must be set to 0.

Table 26.40 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

26.12.1.5 Operating Procedure for Interval Timer Function

Table 26.41 Operating Procedure for Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 26.37, Contents of the TAUJnCMORm register for Interval Timer Function** and **Table 26.38, Contents of the TAUJnCMURm register for Interval Timer Function**. Set the value of the TAUJnCDRm register. Set the channel output mode by setting the control bits as described in **Table 26.39, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJTTOUTm toggles.
During operation	The TAUJnCDRm register value can be changed at any time. The TAUJnCNTm register can be read at all times.	TAUJnCNTm counts down. When the counter reaches 0000 0000_H: • TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. • INTTAUJnIm is generated and TAUJTTOUTm toggles.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.

Restart operation

26.12.1.6 Specific Timing Diagrams

(1) TAUJnCDRm = 0000 0000_H, count clock = PCLK/2

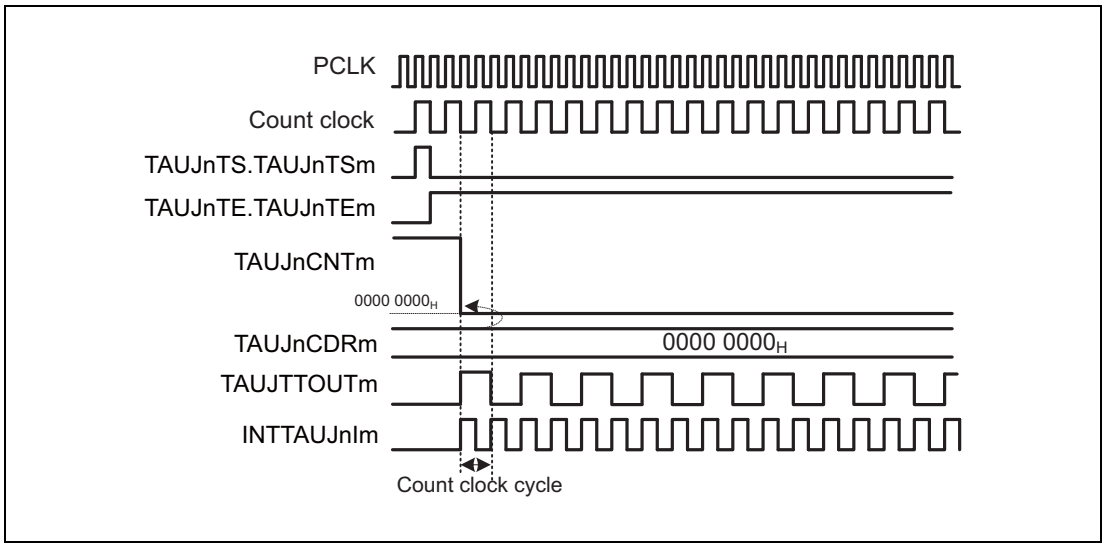


Figure 26.17 AUJnCDRm = 0000 0000_H, Count Clock = PCLK/2

- If TAUJnCDRm = 0000 0000_H and the count clock = PCLK/2, the TAUJnCDRm value is loaded to TAUJnCNTm every count clock, meaning that TAUJnCNTm is always 0000 0000_H.
- INTTAUJnIm is generated every count clock, resulting in TAUJTTOUTm toggling every count clock.

(2) TAUJnCDRm = 0000 0000$_H$, count clock = PCLK

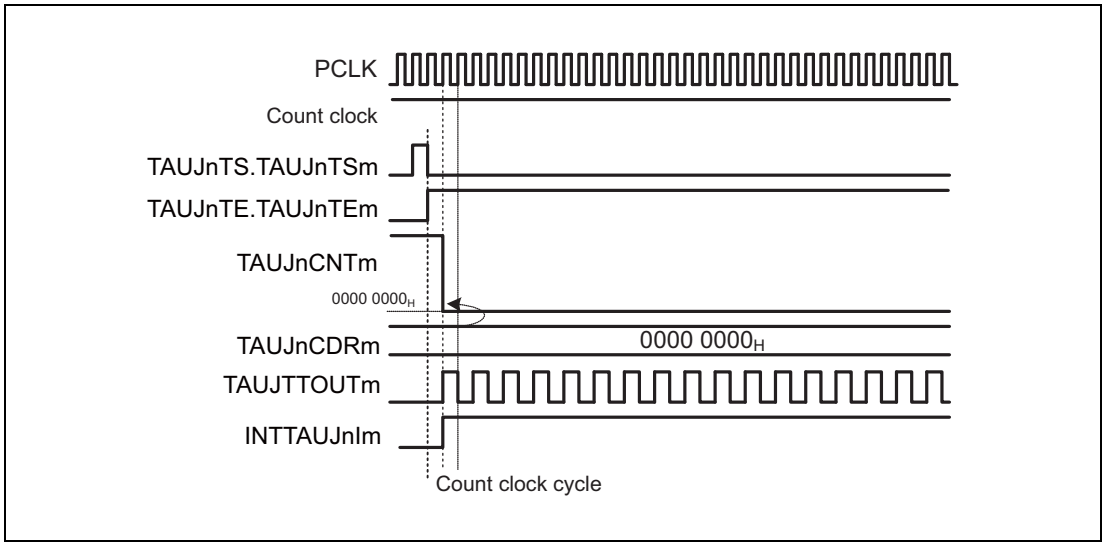


Figure 26.18 TAUJnCDRm = 0000 0000$_H$, Count Clock = PCLK

- If TAUJnCDRm = 0000 0000$_H$ and the count clock = PCLK, the TAUJnCDRm value is loaded to TAUJnCNTm every PCLK clock, meaning that TAUJnCNTm is always 0000 0000$_H$.
- INTTAUJnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated.
  TAUJTTOUTm is toggled every PCLK clock.

(3) Operation stop and restart

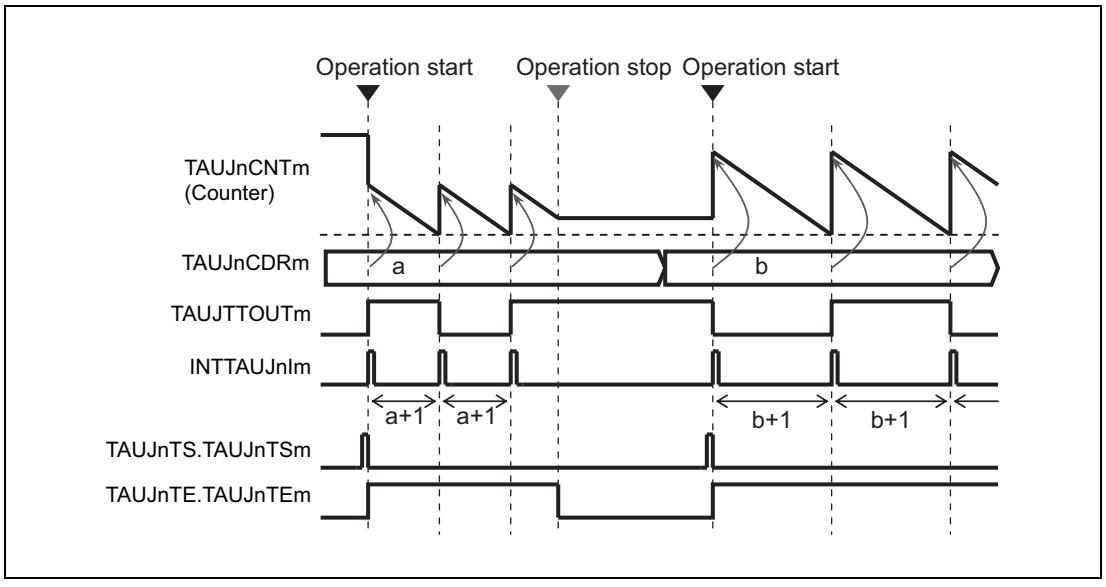


Figure 26.19 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 1)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1.

(4) Forced restart

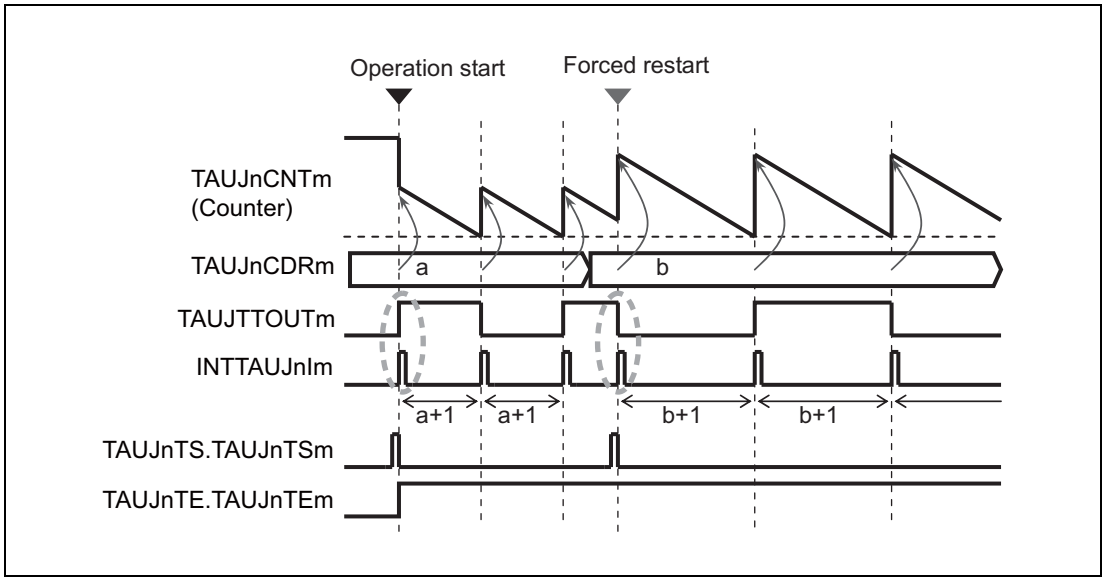


Figure 26.20 Forced Restart Operation (TAUJnCMORm.TAUJnMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.
- If the TAUJnCMORm.TAUJnMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- When a forced restart is made, the TAUJnCDRm value is reflected to TAUJnCNTm and counting starts. Execute a forced restart to reflect the changed TAUJnCDRm value immediately.

26.12.2 TAUJTTINm Input Interval Timer Function

26.12.2.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals or when a valid TAUJTTINm input edge is detected. When an interrupt is generated, the TAUJTTOUTm signal toggles, resulting in a square wave. Output of square waves is only supported for TAUJ0.

Description

This function operates in an identical manner to the interval timer function (see **Section 26.12.1, Interval Timer Function**), except that this function is restarted by a valid TAUJTTINm input edge. The type of edge used as the trigger is specified using the TAUJnCMURm.TAUJnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

26.12.2.2 Equations

INTTAUJnIm cycle = count clock cycle × (TAUJnCDRm + 1)

TAUJTTOUTm square wave cycle = count clock cycle × (TAUJnCDRm + 1) × 2

26.12.2.3 Block Diagram and General Timing Diagram

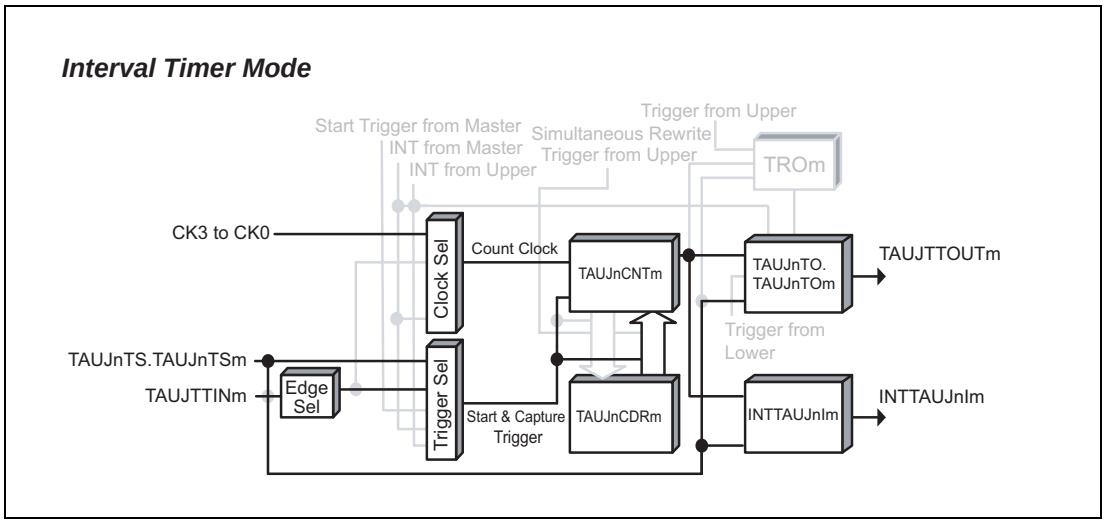


Figure 26.21 Block Diagram for TAUJTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1).
- Rising edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 01_B)

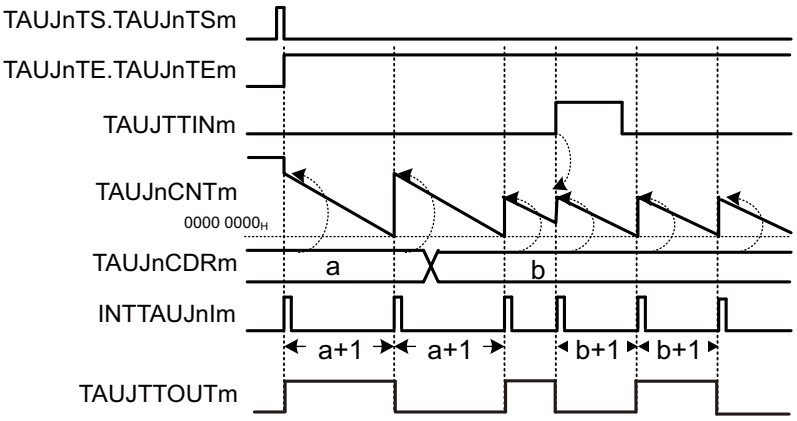


Figure 26.22 General Timing Diagram for TAUJTTINm Input Interval Timer Function

26.12.2.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 26.42 Contents of the TAUJnCMORm register for TAUJTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 001_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated and TAUJTTOUTm does not toggle when operation starts. 1: Generates INTTAUJnIm and toggles TAUJTTOUTm when operation starts.

(2) TAUJnCMURm

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

Table 26.43 Contents of the TAUJnCMURm register for TAUJTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

Table 26.44 Control Bit Settings for Independent Channel Output Mode 1

Bit name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 0_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	Write 0_B.

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJTTOUTm can then be controlled independently of the interrupts. For details refer to **Section 26.7, Channel Output Modes**.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input interval timer function. Therefore, these registers must be set to 0.

Table 26.45 Simultaneous Rewrite Settings for TAUJTTINm Input Interval Timer Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

26.12.2.5 Operating Procedure for TAUJTTINm Input Interval Timer Function

Table 26.46 Operating Procedure for TAUJTTINm Input Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 26.42, Contents of the TAUJnCMORm register for TAUJTTINm Input Interval Timer Function** and **Table 26.43, Contents of the TAUJnCMURm register for TAUJTTINm Input Interval Timer Function**. Set the value of the TAUJnCDRm register Set the channel output mode by setting the control bits as described in **Table 26.44, Control Bit Settings for Independent Channel Output Mode 1**.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJTTOUTm toggles.
During operation	The values of the TAUJnCMURm.TAUJnTIS[1:0] and TAUJnCDRm registers can be changed at any time. The TAUJnCNTm register can be read at all times. Detection of TAUJTTINm edge	TAUJnCNTm counts down. When the counter reaches 0000 0000_H: • TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. • INTTAUJnIm is generated and TAUJTTOUTm toggles. When a TAUJTTINm input valid edge is detected during count operation, TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.

Restart operation

26.12.2.6 Specific Timing Diagrams

The timing diagrams in **Section 26.12.1, Interval Timer Function** apply, and in addition the counter can also be restarted by a valid TAUJTTINm input edge.

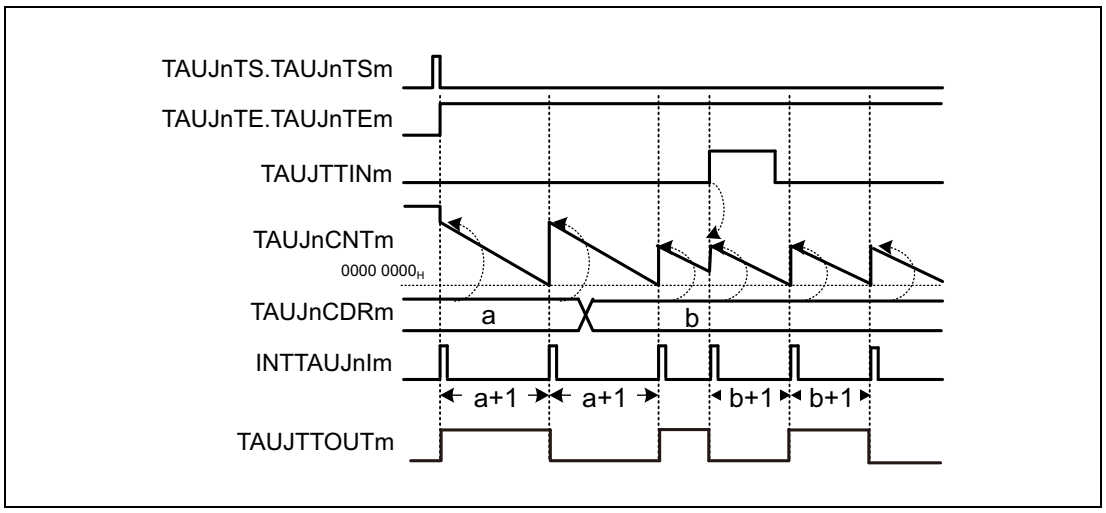


Figure 26.23 Counter Triggered By Rising TAUJTTINm input edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B), TAUJnCMORm.TAUJnMD0 = 1

If a valid TAUJTTINm input edge is detected, an interrupt is generated which causes TAUJTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B).

26.12.3 TAUJTTINm Input Pulse Interval Measurement Function

26.12.3.1 Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUJnCSRm.TAUJnOVF to measure the interval of the TAUJTTINm input signals.

Prerequisites

TAUJTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJTTINm edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter resets to 0000 0000_H and subsequently continues operation.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJTTINm edge is detected, it overflows to 0000 0000_H. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0].

Table 26.47 Effects of an Overflow

TAUJnCMORm.COS[1:0]	When Overflow Occurs		When a Valid TAUJTTINm Input is then Detected	
	TAUJnCDRm	TAUJnCSRm.TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm.TAUJnOVF
00	Unchanged	0	TAUJnCNTm loaded to TAUJnCDRm	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm set to 0, TAUJnCDRm unchanged	Unchanged
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, the overflow by (TAUJnCSRm.TAUJnOVF = 1) can only be cleared by setting TAUJnCSCm.TAUJnCLOV = 1.

The combination of the values of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the interval of the TAUJTTINm signal. However, if an overflow occurs multiple times before a valid TAUJTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

The function can be stopped by setting TAUJnTT.TAUJnTTm = 1, which in turn sets TAUJnTE.TAUJnTEm = 0. TAUJnCNTm stops but retains its value. While the function is stopped, TAUJTTINm input valid edge detection and TAUJnCNTm capture are not performed.

Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, refer to **26.9, TAUJTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts**.

NOTE

When TAUJnCMORm.TAUJnCOS[1] = 1, the value of TAUJnCNTm is not loaded to TAUJnCDRm when the first valid TAUJTTINm input edge occurs after an overflow. However, an interrupt is generated.

26.12.3.2 Equations

TAUJTTINm input pulse interval = count clock cycle ×
[(TAUJnCSRm.TAUJnOVF × (FFFF $FFFF_H$ + 1)) + TAUJnCDRm capture value + 1]

26.12.3.3 Block Diagram and General Timing Diagram

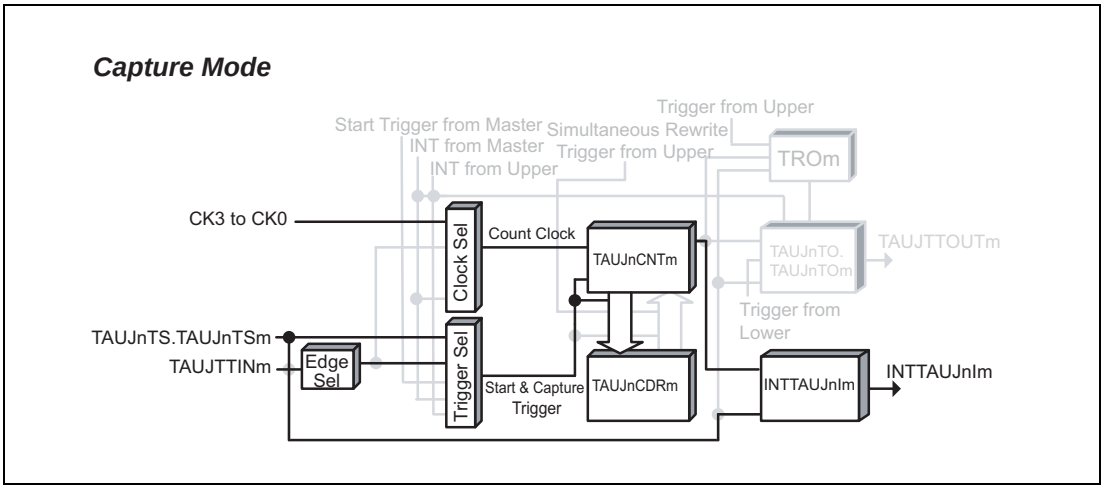


Figure 26.24 Block Diagram for TAUJTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is not generated when operation starts (TAUJnCMORm.TAUJnMD0 = 0).
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)
- When a valid TAUJTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B).

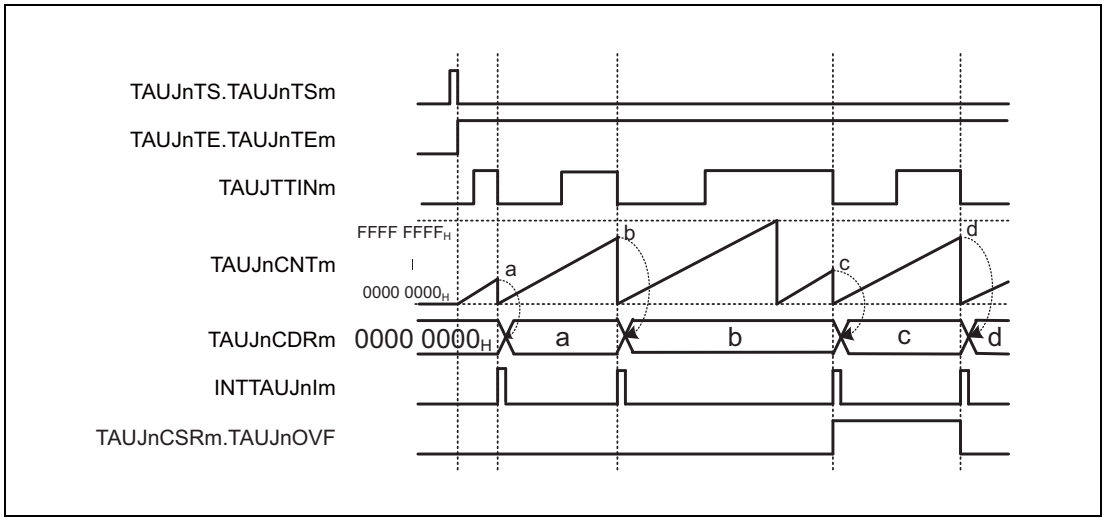


Figure 26.25 General Timing Diagram For TAUJTTINm Input Pulse Interval Measurement Function

26.12.3.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 26.48 Contents of the TAUJnCMORm Register for TAUJTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 001_B.
7, 6	TAUJnCOS[1:0]	See **Table 26.47, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0010_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated when operation starts. 1: Generates INTTAUJnIm when operation starts.

(2) TAUJnCMURm

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

Table 26.49 Contents of the TAUJnCMURm Register for TAUJTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input pulse interval measurement function. Therefore, these registers must be set to 0.

Table 26.50 Simultaneous Rewrite Settings for TAUJTTINm Input Pulse Interval Measurement Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

26.12.3.5 Operating Procedure for TAUJTTINm Input Pulse Interval Measurement Function

Table 26.51 Operating Procedure for TAUJTTINm Input Pulse Interval Measurement Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 26.48, Contents of the TAUJnCMORm Register for TAUJTTINm Input Pulse Interval Measurement Function** and **Table 26.49, Contents of the TAUJnCMURm Register for TAUJTTINm Input Pulse Interval Measurement Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm is cleared to 0000 0000_H. INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
During operation	Detection of TAUJTTINm edges. The values of the TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time. TAUJnCSCm.TAUJnCLOV bit can be written to 1. (TAUJnCSRm.TAUJnOVF bit is cleared to 0.)	TAUJnCNTm starts to count up from 0000 0000_H. When a TAUJTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and returns to 0000 0000_H. • INTTAUJnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

Restart operation

26.12.3.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

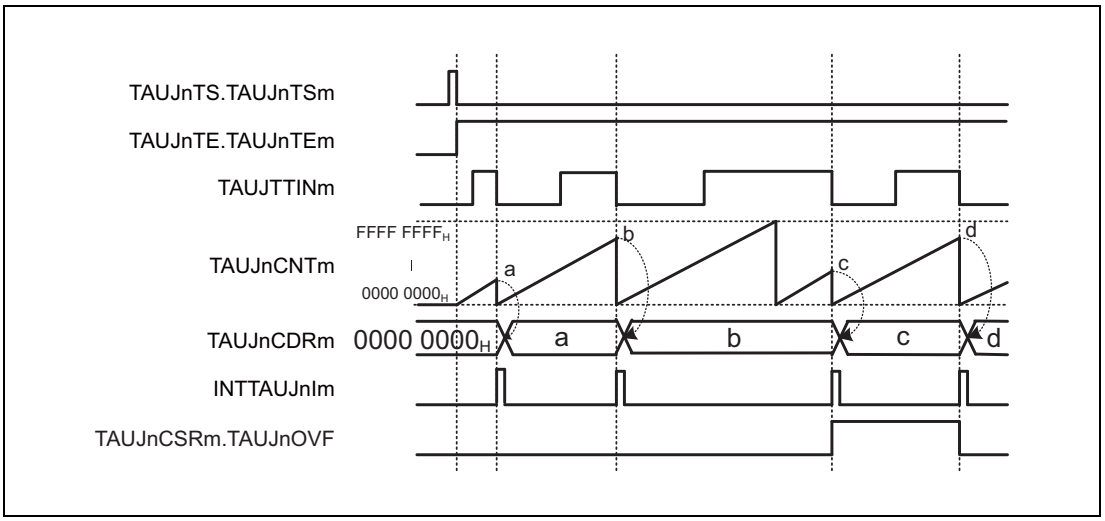


Figure 26.26 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- If the next valid TAUJTTINm input edge is detected when no overflow occurs, TAUJnCSRm.TAUJnOVF is cleared to 0.

(2) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

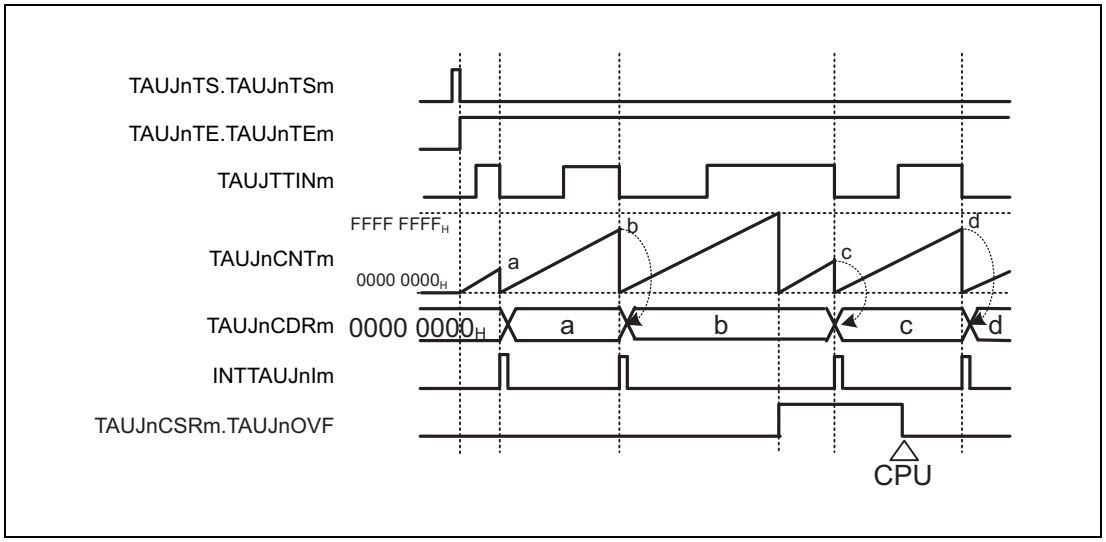


Figure 26.27 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF is set to1.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (by setting the TAUJnCSCm.TAUJnCLOV bit to 1).

(3) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

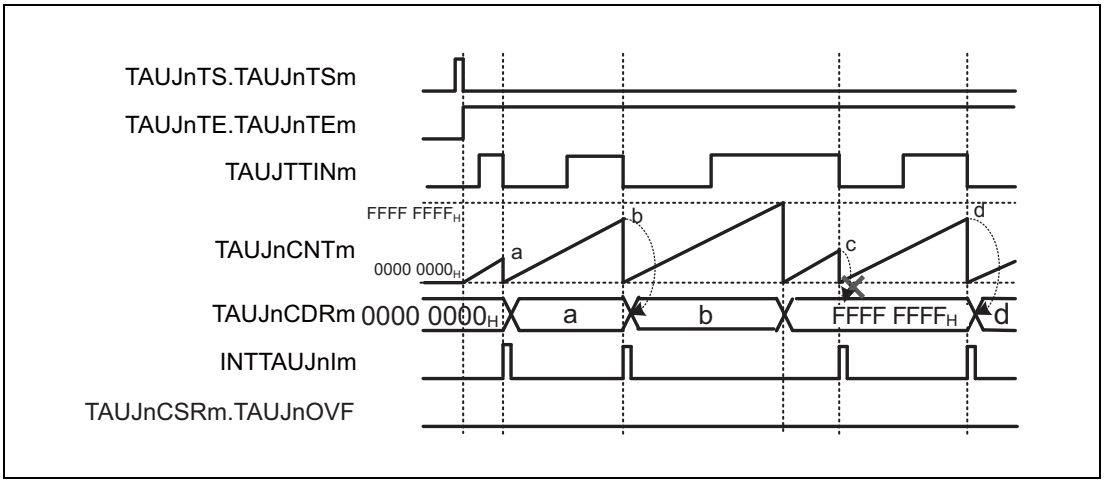


Figure 26.28 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$ and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.

(4) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

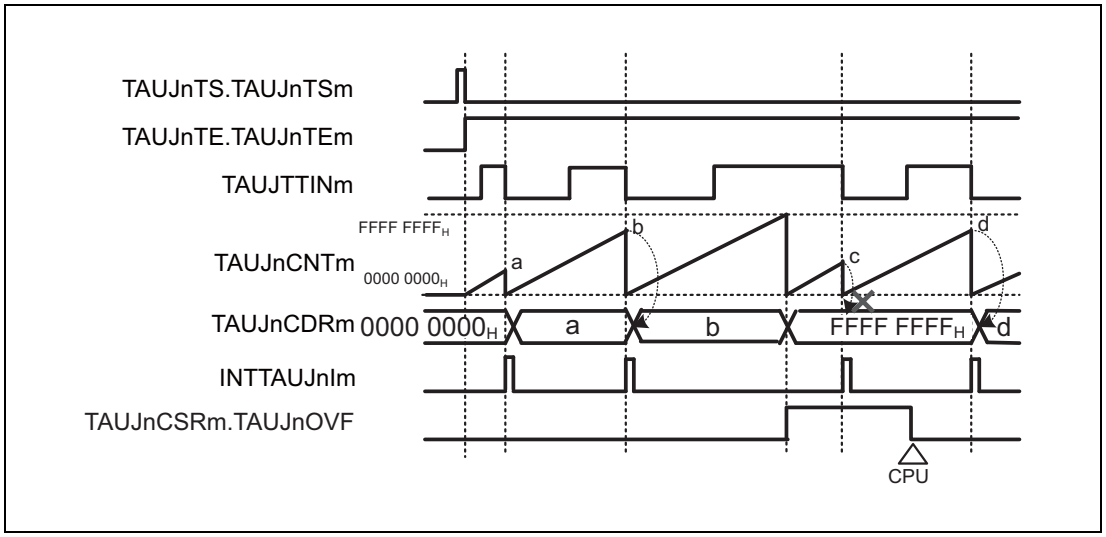


Figure 26.29 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

26.12.4 TAUJTTINm Input Signal Width Measurement Function

26.12.4.1 Overview

Summary

This function measures the width of a TAUJTTINm signal by starting counting on one edge of the TAUJTTINm signal and capturing the counter value on the opposite edge.

Prerequisites

TAUJTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. When a valid TAUJTTINm start edge is detected, the counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJTTINm stop edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter retains its value and awaits the next valid TAUJTTINm input start edge.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJTTINm stop edge is detected, it overflows. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0].

Table 26.52 Effects of an Overflow

TAUJnCMORm.COS[1:0]	When Overflow Occurs		When a Valid TAUJTTINm Input Stop Edge is Detected	
	TAUJnCDRm	TAUJnCSRm.TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm.TAUJnOVF
00	Unchanged	0	TAUJnCNTm is loaded to TAUJnCDRm.	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm stops counting, TAUJnCDRm unchanged	Unchanged
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, the overflow bit (TAUJnCSRm.TAUJnOVF = 1) can only be cleared by setting TAUJnCSCm.TAUJnCLOV to 1.

The combination of the values of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the width of the TAUJTTINm signal. However, if an overflow occurs multiple times before a valid TAUJTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

This function cannot be forcibly restarted.

NOTE

When TAUJnCMORm.COS[1] = 1, the value of TAUJnCNTm is not loaded to TAUJnCDRm when the first valid TAUJTTINm input edge occurs after an overflow. However, an interrupt is generated.

26.12.4.2 Equations

TAUJTTINm input signal width = count clock cycle ×
[(TAUJnCSRm.TAUJnOVF × (FFFF FFFF$_H$ + 1)) + TAUJnCDRm capture value + 1]

26.12.4.3 Block Diagram and General Timing Diagram

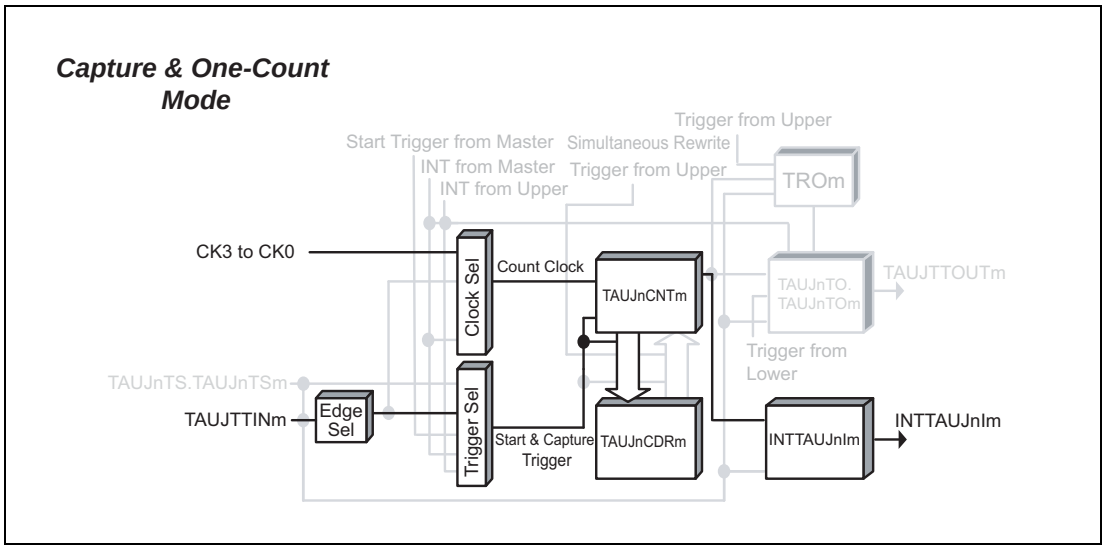


Figure 26.30 Block Diagram for TAUJTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)
- When a valid TAUJTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B).

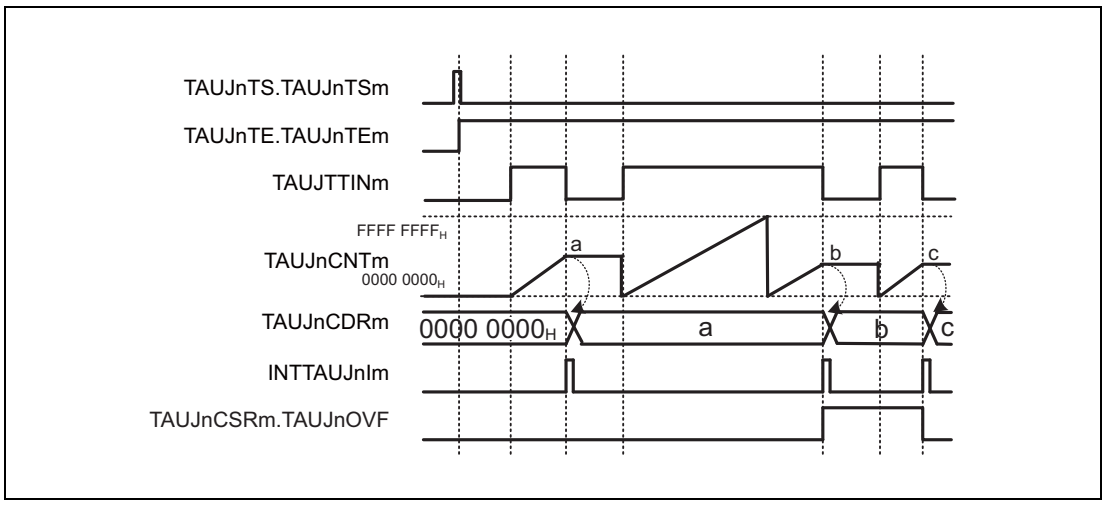


Figure 26.31 General Timing Diagram for TAUJTTINm Input Signal Width Measurement Function

26.12.4.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 26.53 Contents of the TAUJnCMORm Register for TAUJTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	See **Table 26.52, Effects of an Overflow**.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0110_B.
0	TAUJnMD0	Write 0_B.

(2) TAUJnCMURm

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

Table 26.54 Contents of the TAUJnCMURm Register for TAUJTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input signal width measurement function. Therefore, these registers must be set to 0.

Table 26.55 Simultaneous Rewrite Settings for TAUJTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

26.12.4.5 Operating Procedure for TAUJTTINm Input Signal Width Measurement Function

Table 26.56 Operating Procedure for TAUJTTINm Input Signal Width Measurement Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 26.53, Contents of the TAUJnCMORm Register for TAUJTTINm Input Signal Width Measurement Function** and **Table 26.54, Contents of the TAUJnCMURm Register for TAUJTTINm Input Signal Width Measurement Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJTTINm start edge. When a TAUJTTINm start edge is detected, TAUJnCNTm starts to count up.
During operation	The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time. The TAUJnCSCm.TAUJnCLOV bit can be set to 1.	TAUJnCNTm starts to count up from 0000 0000_H. When a TAUJTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and retains its value. • INTTAUJnIm is then generated. • Counting stops at the "value that transferred to TAUJnCDRm + 1" and TAUJnCNTm waits for detection of the TAUJTTINm start edge. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

Restart operation

26.12.4.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

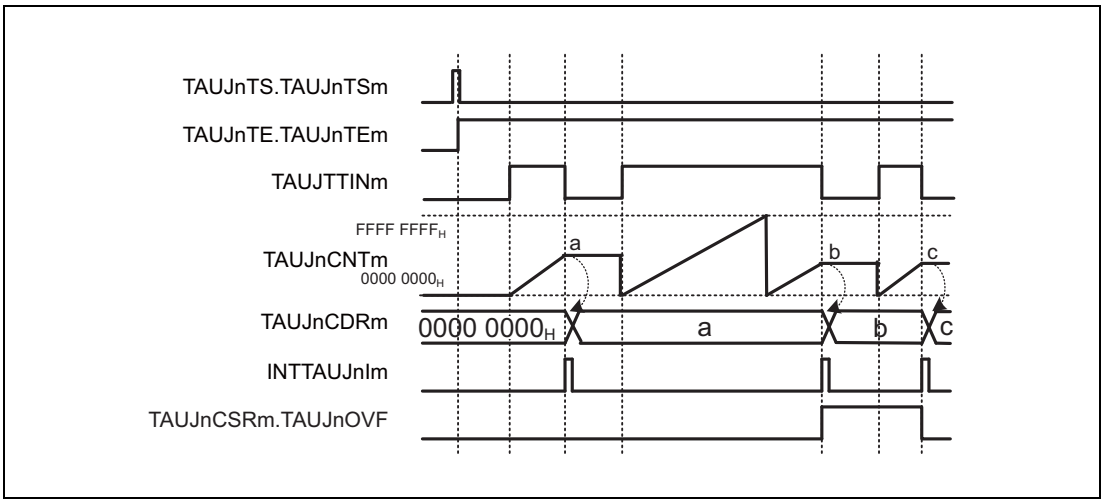


Figure 26.32 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge with no overflow occurring, TAUJnCSRm.TAUJnOVF is cleared to 0.

(2) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

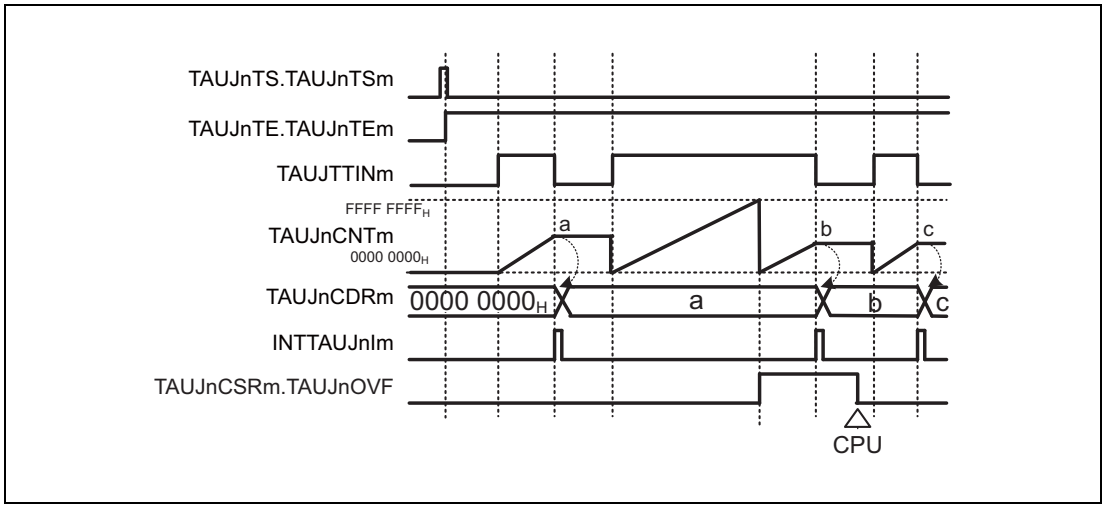


Figure 26.33 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (by setting the TAUJnCSCm.TAUJnCLOV bit to 1).

(3) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

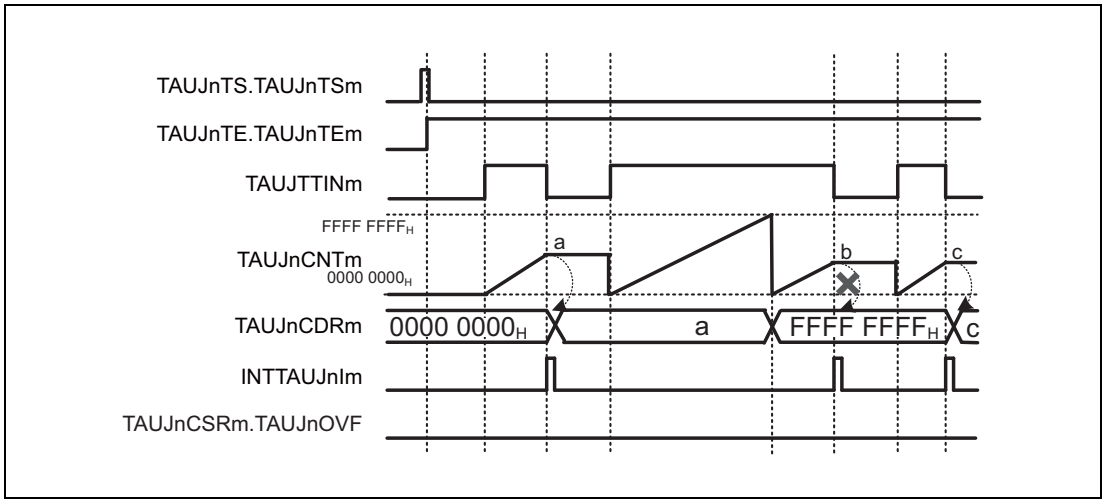


Figure 26.34 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$ and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.

(4) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

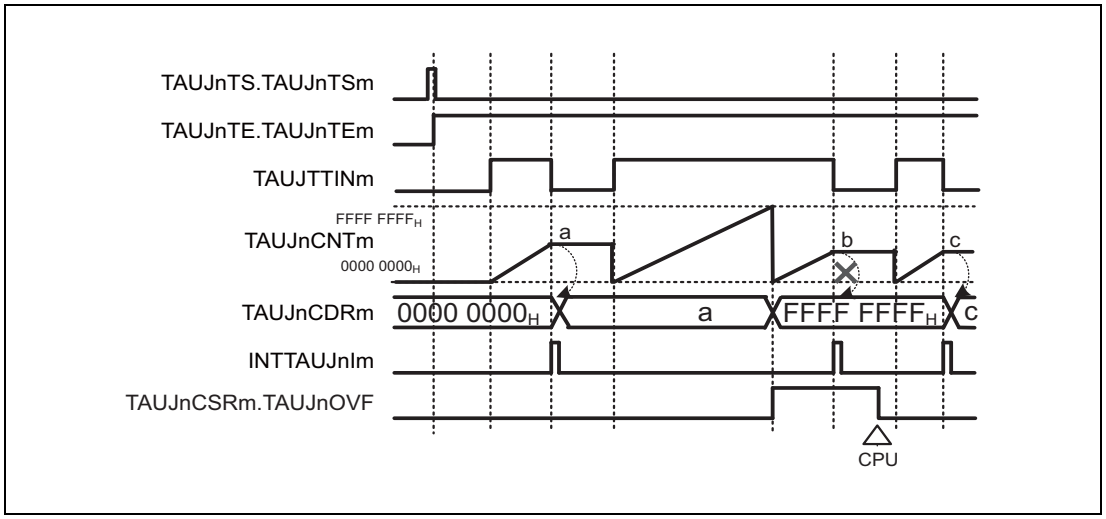


Figure 26.35 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

26.12.5 TAUJTTINm Input Position Detection Function

26.12.5.1 Overview

Summary

This function measures the interval of input signals by capturing the counter value on a valid edge of the TAUJTTINm signal.

Prerequisites

TAUJTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter starts to count from 0000 0000_H. When a valid TAUJTTINm input edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter continues to count.

When the counter reaches FFFF $FFFF_H$, the counter restarts from 0000 0000_H.

NOTE

The input TAUJTTINm is sampled at the frequency of the operation clock, specified by TAUJnCMORm.TAUJnCKS[1:0] bits. As a result, the output cycle of TAUJTTOUTm has an error of ± 1 operation clock cycle.

Conditions

If the TAUJnCMORm. TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated.

26.12.5.2 Equations

Function duration at a TAUJTTINm input pulse =
count clock cycle × (TAUJnCDRm capture value + 1)

26.12.5.3 Block Diagram and General Timing Diagram

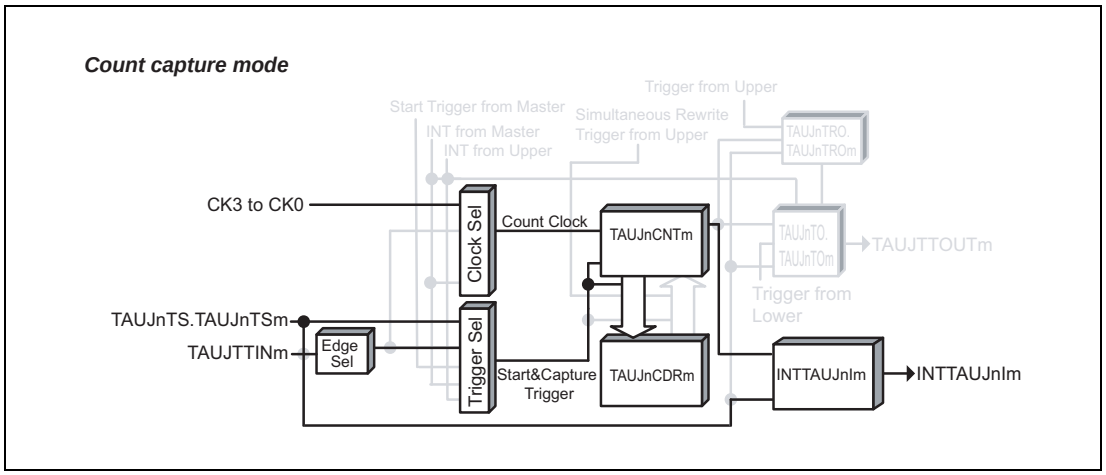


Figure 26.36 Block Diagram of TAUJTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is not generated when operation starts (TAUJnCMORm.TAUJnMD0 = 0).
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

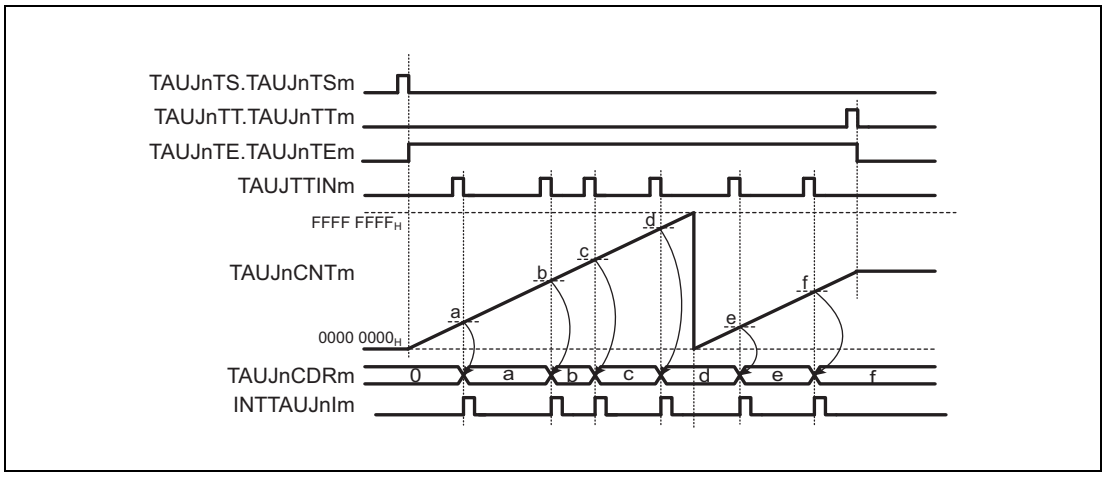


Figure 26.37 General Timing Diagram for TAUJTTINm Input Position Detection Function

26.12.5.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 26.57 Contents of the TAUJnCMORm Register for TAUJTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 001_B.
7, 6	TAUJnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1011_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated when operation starts. 1: Generates INTTAUJnIm when operation starts.

(2) TAUJnCMURm

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

Table 26.58 Contents of the TAUJnCMURm Register for TAUJTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input position detection function. Therefore, these registers must be set to 0.

Table 26.59 Simultaneous Rewrite Settings for TAUJTTINm Input Position Detection Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0

26.12.5.5 Operating Procedure for TAUJTTINm Input Position Detection Function

Table 26.60 Operating Procedure for TAUJTTINm Input Position Detection Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in **Table 26.57, Contents of the TAUJnCMORm Register for TAUJTTINm Input Position Detection Function** and **Table 26.58, Contents of the TAUJnCMURm Register for TAUJTTINm Input Position Detection Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
During operation	The values of the TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time.	TAUJnCNTm starts to count up from 0000 0000_H. When a TAUJTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm. • INTTAUJnIm is output. • The counter value is not cleared to 0000 0000_H and TAUJnCNTm continues count operation. Afterwards, this procedure is repeated. When TAUJnCNTm reaches FFFF $FFFF_H$, the counter restarts from 0000 0000_H.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

26.12.5.6 Specific Timing Diagrams

(1) Operation stop and restart

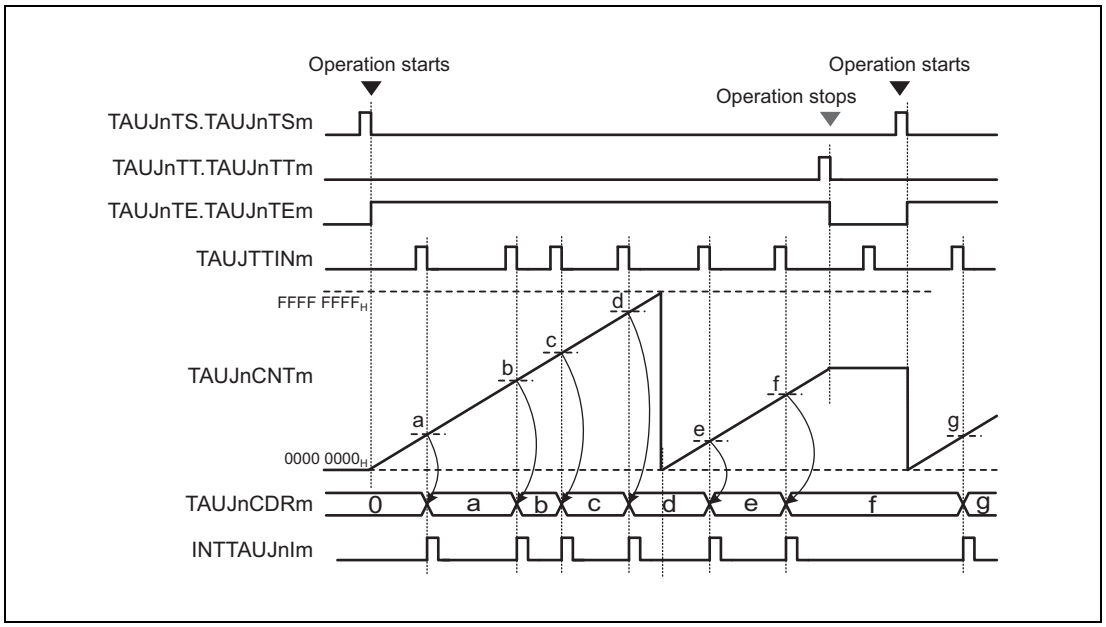


Figure 26.38 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts to count from 0000 0000_H.

26.12.6 TAUJTTINm Input Period Count Detection Function

26.12.6.1 Overview

Summary

This function measures the cumulative width of a TAUJTTINm input signal.

Prerequisites

TAUJTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter awaits a valid TAUJTTINm input edge.

When a valid TAUJTTINm input start edge is detected, the counter starts to count from 0000 0000_H.

When a valid TAUJTTINm input stop edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter stops and retains its value until the next valid TAUJTTINm input start edge is detected.

When the next valid TAUJTTINm input start edge is detected, the counter restarts counting from the stop value.

When the counter reaches FFFF $FFFF_H$, the counter restarts from 0000 0000_H.

This function cannot be forcibly restarted.

NOTE

The input TAUJTTINm signal is sampled at the frequency of the operation clock, specified by the TAUJnCMORm.TAUJnCKS[1:0] bits.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

26.12.6.2 Equations

Cumulative TAUJTTINm input width =
count clock cycle × (TAUJnCDRm capture value + 1)

26.12.6.3 Block Diagram and General Timing Diagram

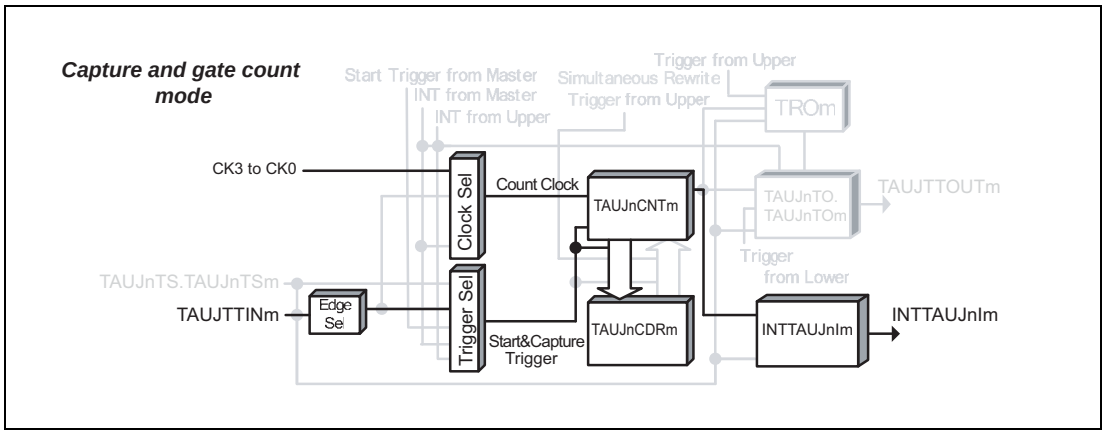


Figure 26.39 Block Diagram for TAUJTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

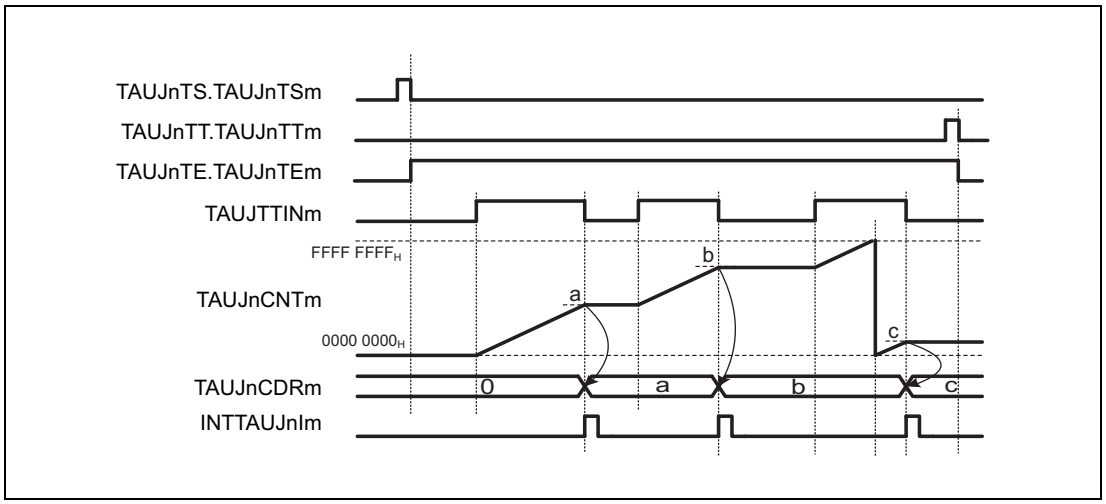


Figure 26.40 General Timing Diagram for TAUJTTINm Input Period Count Detection Function

26.12.6.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 26.61 Contents of the TAUJnCMORm Register for TAUJTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1101_B.
0	TAUJnMD0	Write 0_B.

(2) TAUJnCMURm

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

Table 26.62 Contents of the TAUJnCMURm Register for TAUJTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input period count detection function. Therefore, these registers must be set to 0.

Table 26.63 Simultaneous Rewrite Settings for TAUJTTINm Input Period Count Detection Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0

26.12.6.5 Operating Procedure for TAUJTTINm Input Period Count Detection Function

Table 26.64 Operating Procedure for TAUJTTINm Input Period Count Detection Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 26.61, Contents of the TAUJnCMORm Register for TAUJTTINm Input Period Count Detection Function** and **Table 26.62, Contents of the TAUJnCMURm Register for TAUJTTINm Input Period Count Detection Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJTTINm start edge.
During operation	TAUJTTINm edge detection The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time.	When a TAUJTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUJnCNTm starts to count up from the stop value. When TAUJnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUJnCDRm and INTTAUJnIm is generated. Counting stops at the “value transferred to TAUJnCDRm + 1” and TAUJnCNTm waits for detection of the TAUJTTINm start edge. When TAUJnCNTm reaches FFFF FFFF$_H$, the counter restarts from 0000 0000$_H$. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

26.12.6.6 Specific Timing Diagrams

(1) Operation Stop and Restart

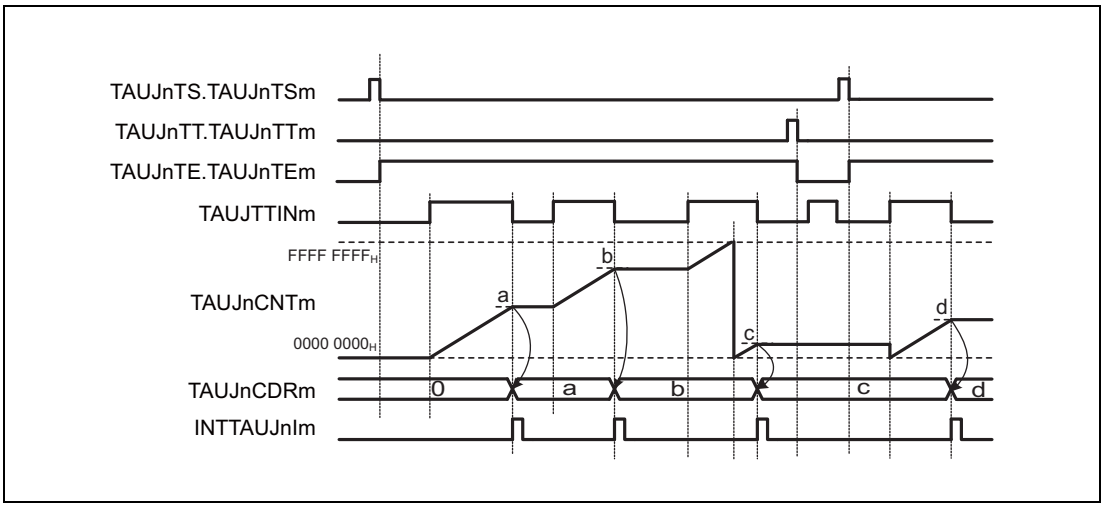


Figure 26.41 Operation Stop and Restart (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts to count from 0000 0000_H.

26.12.7 Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

26.12.7.1 Overview

Summary

This function measures the width of an individual TAUJTTINm input signal. An interrupt is generated if the TAUJTTINm input width is longer than FFFF $FFFF_H$ + 1.

Prerequisites

- TAUJTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to FFFF $FFFF_H$.

Description

he counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation.

The counter starts when a valid TAUJTTINm input start edge is detected. FFFF $FFFF_H$ is loaded to TAUJnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUJTTINm input start edge is detected, TAUJnCNTm loads FFFF $FFFF_H$ and starts to count down.

If the counter reaches $0000\ 0000_H$ before a stop edge is detected, an interrupt is generated.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

26.12.7.2 Block Diagram and General Timing Diagram

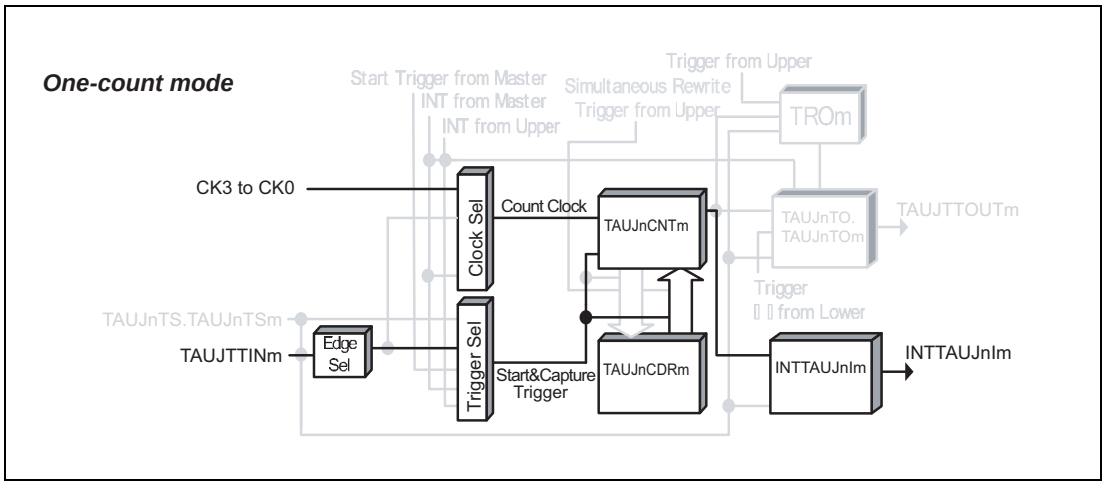


Figure 26.42 Block Diagram for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

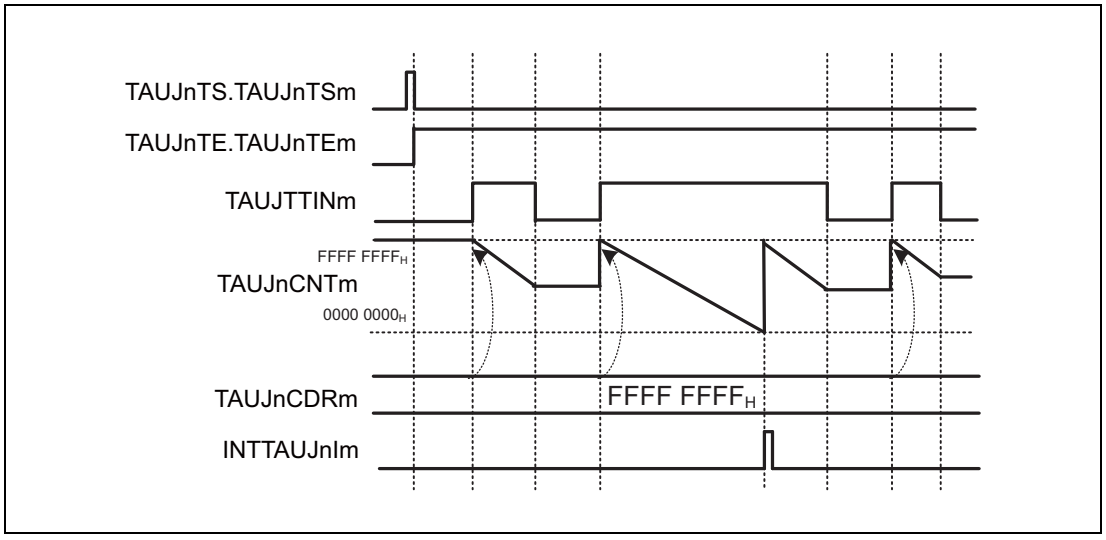


Figure 26.43 General Timing Diagram for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

26.12.7.3 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 26.65 Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0100_B.
0	TAUJnMD0	Write 0_B.

(2) TAUJnCMURm

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

Table 26.66 Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the overflow interrupt output function (during TAUJTTINm width measurement). Therefore, these registers must be set to 0.

Table 26.67 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

26.12.7.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Table 26.68 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in **Table 26.65, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)** and **Table 26.66, Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)**. Set the value of the TAUJnCDRm register to FFFF FFFF$_H$.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUJTTINm start edge.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the start edge. When a start edge is detected, TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF$_H$).
During operation	The TAUJnCNTm register can be read at any time.	TAUJnCNTm counts down. When the counter reaches 0000 0000$_H$: • INTTAUJnIm is generated. When a reverse edge of TAUJTTINm is detected during count operation: • TAUJnCNTm stops counting and waits for a trigger. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

26.12.8 Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

26.12.8.1 Overview

Summary

This function measures the cumulative width of a TAUJTTINm input signal. An interrupt is generated if the cumulative TAUJTTINm input width is longer than FFFF FFFF$_H$, and an overflow interrupt can be output.

Prerequisites

- TAUJTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to FFFF FFFF$_H$.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation.

The counter starts when a valid TAUJTTINm input start edge is detected. FFFF FFFF$_H$ is loaded to TAUJnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUJTTINm input start edge and then continues to count down from the current value.

When the counter reaches 0000 0000$_H$ an interrupt is generated. FFFF FFFF$_H$ is loaded to TAUJnCNTm and the counter continues to count down until a TAUJTTINm input stop edge is detected.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10$_B$, the TAUJTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11$_B$, the TAUJTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

26.12.8.2 Block Diagram and General Timing Diagram

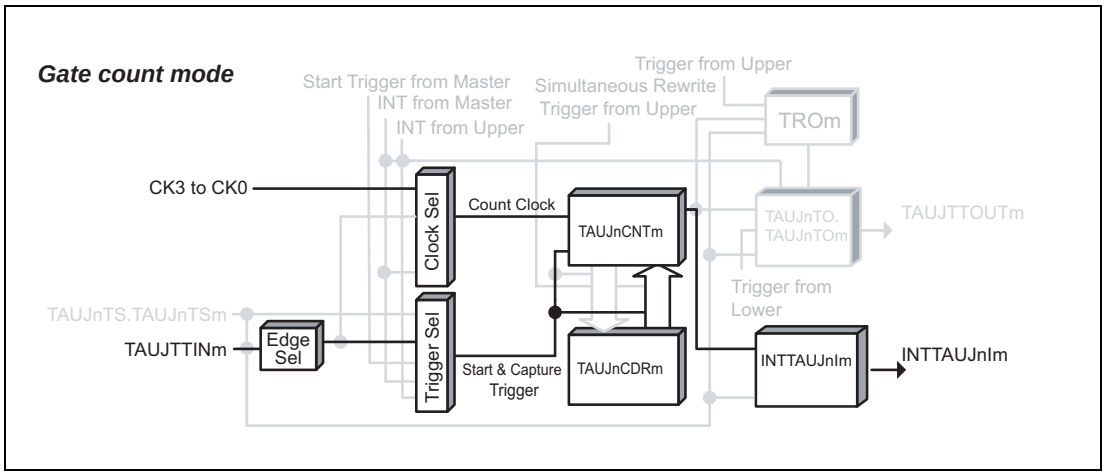


Figure 26.44 Block Diagram for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

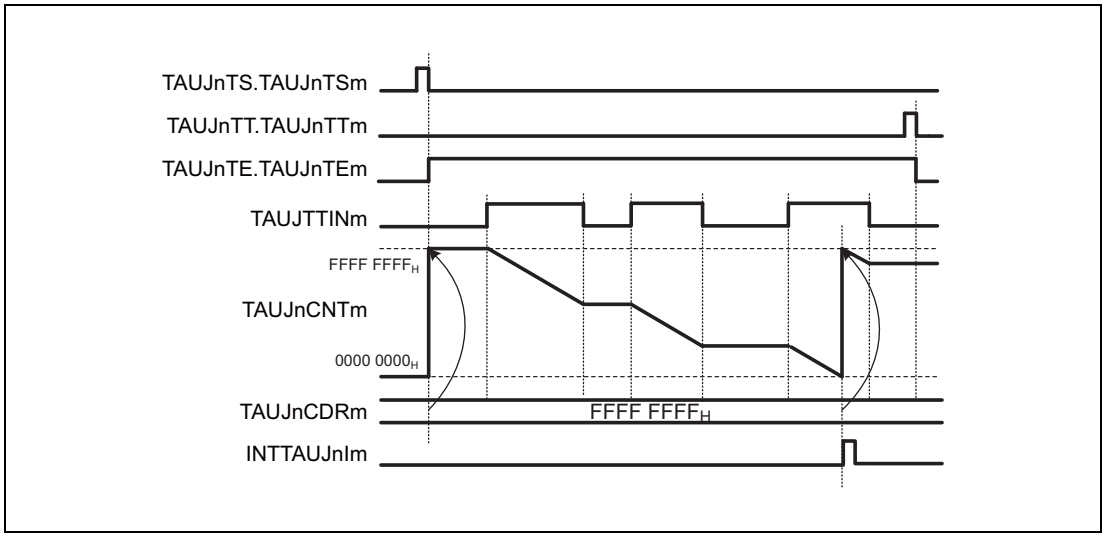


Figure 26.45 General Timing Diagram For Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

26.12.8.3 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 26.69 Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1100_B.
0	TAUJnMD0	Write 0_B.

(2) TAUJnCMURm

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

Table 26.70 Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the Overflow Interrupt Output Function (During TAUJTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 26.71 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

26.12.8.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Table 26.72 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 26.69, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)** and **Table 26.70, Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)**. Set the value of the TAUJnCDRm register to FFFF FFFF$_H$.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUJTTINm start edge.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the start edge. When a start edge is detected, the value of TAUJnCDRm (FFFF FFFF$_H$) is loaded to TAUJnCNTm.
During operation	The TAUJnCNTm register can be read at all times	TAUJnCNTm counts down. When the counter reaches 0000 0000$_H$: • INTTAUJnIm is generated. • TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF$_H$) and continues to count down. When a reverse edge of TAUJTTINm is detected during count operation: • TAUJnCNTm stops and retains the stop value. When a TAUJTTINm valid edge is detected while the counter is stopped: • TAUJnCNTm counts down from the stop value. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

26.13 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the TAUJ. For a general overview of synchronous channel operation, see **Section 26.2, Overview**.

26.13.1 PWM Output Function

26.13.1.1 Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty of the TAUJTTOUTm to be set. The pulse cycle is set in the master channel. The duty is set in the slave channel.

Prerequisites

- Two channels
- The operation mode for the master channel should be set to the interval timer mode. (See **Table 26.73, Contents of the TAUJnCMORm Register for the Master Channel of the PWM Output Function**.)
- The operation mode for the slave channel should be set to the one-count mode. (See **Table 26.76, Contents of the TAUJnCMORm Register for the Slave Channel of the PWM Output Function**.)
- TAUJTTOUTm is not used for the master channel of this function.
- The channel output mode for the slave channels should be set to synchronous channel output mode 1. (See **26.7, Channel Output Modes**.)

Description

The counters are enabled by setting the channel trigger bits (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counters start to count down from these values. INTTAUJnIm is generated on the master channel and TAUJTTOUTm (slave) is set or reset to realize the PWM output.

- Master channel:

  When the counter of the master channel reaches 0000 0000_H and pulse cycle time has elapsed, INTTAUJnIm is generated. The TAUJnCDRm value is loaded to TAUJnCNTm, and the counter counts down.

- Slave channel(s):

  INTTAUJnIm generated on the master channel triggers the counter of the slave channel(s). The current value of TAUJnCDRm (slave) is loaded to TAUJnCNTm (slave) and the counter starts to count down from this value. The TAUJTTOUTm signal is set to the active level.
  When the counter reaches 0000 0000_H, i.e. duty time has elapsed, INTTAUJnIm is generated and the TAUJTTOUTm signal is set to the inactive level. The counter returns to FFFF $FFFF_H$ and awaits the next INTTAUJnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1 for the master and slave channel(s), which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJTTOUTm of master and slave

channel(s) stop but retain their values. The counters can be restarted by setting TAUJnTS.TAUJnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. Please refer to **Section 26.6, Simultaneous Rewrite**.

26.13.1.2 Equations

Pulse cycle = (TAUJnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUJnCDRm (slave)/(TAUJnCDRm (master) +1)) × 100

– Duty cycle = 0%
  TAUJnCDRm (slave) = 0000 0000_H
– Duty cycle = 100%
  TAUJnCDRm (slave) ≥ TAUJnCDRm (master) + 1

26.13.1.3 Block Diagram and General Timing Diagram

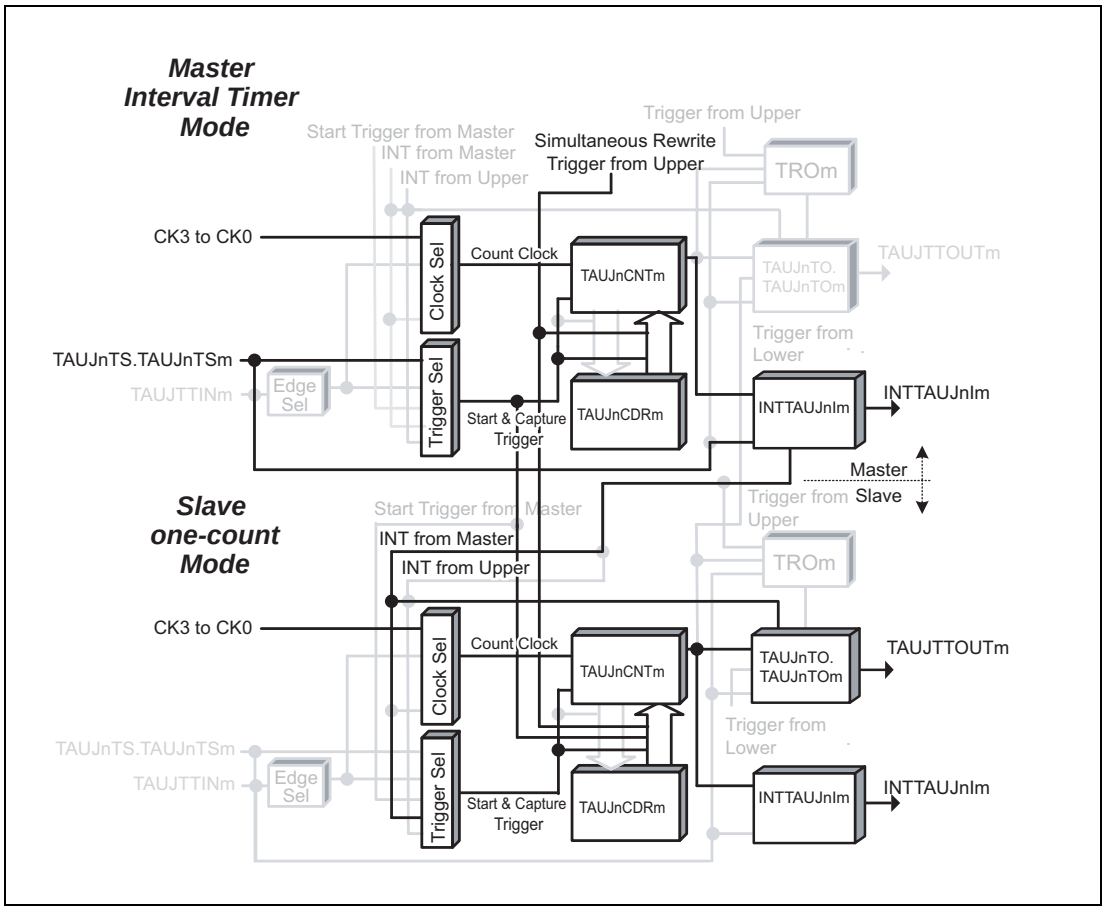


Figure 26.46 Block Diagram for PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUJnTOL.TAUJnTOLm = 0)

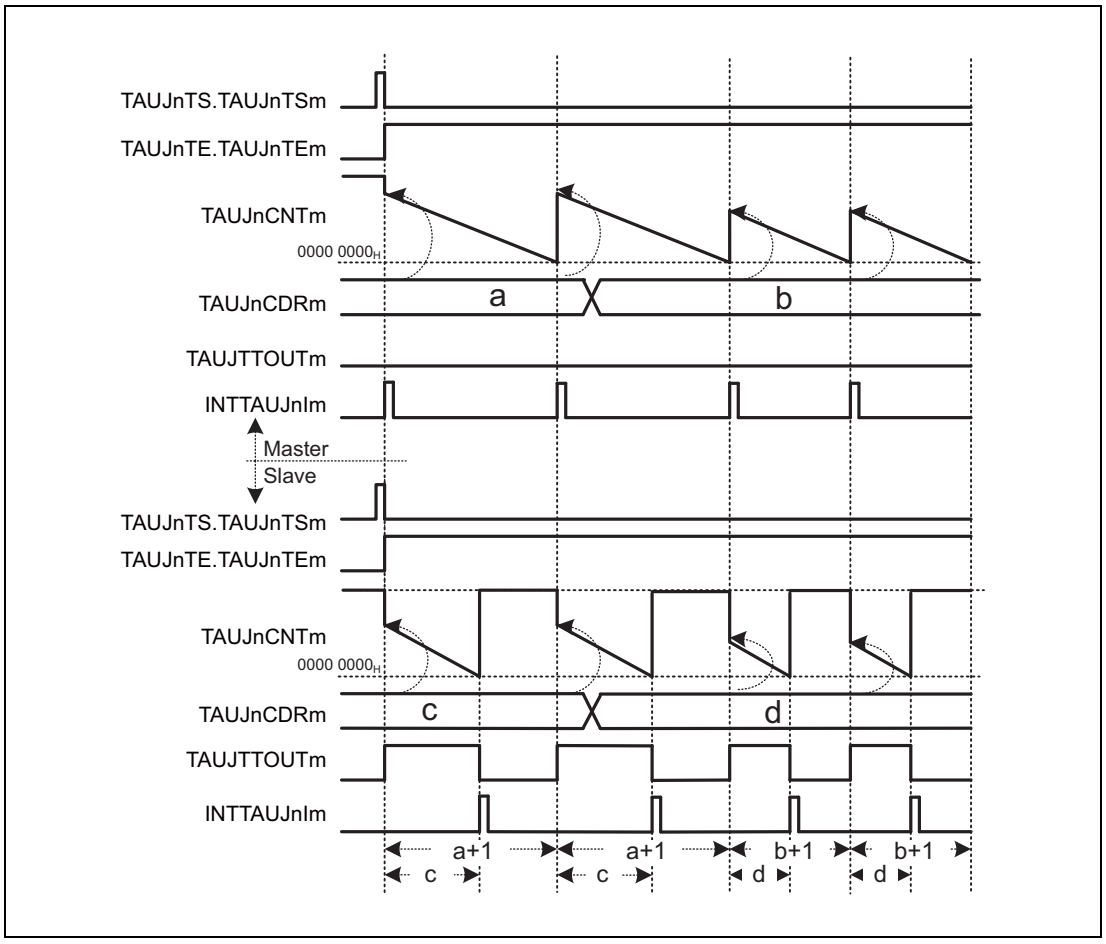


Figure 26.47 General Timing Diagram for PWM Output Function

NOTE

- The interval between the starting to count and an interrupt being generated is the value of corresponding TAUJnCDRm + 1.
- TAUJTTOUTm of the slave channel will rise with a delay of one count clock after the rising of INTTAUJnIm of the master channel.

26.13.1.4 Register Settings for the Master Channel

(1) TAUJnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 26.73 Contents of the TAUJnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bits of the master and slave channel(s) must be identical.
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 1_B.
10 to 8	TAUJnSTS[2:0]	Write 000_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	Write 1_B.

(2) TAUJnCMURm for the master channel

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

Table 26.74 Contents of the TAUJnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

The channel output mode is not used by this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 26.75 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting.

26.13.1.5 Register Settings for the Slave Channel(s)

(1) TAUJnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 26.76 Contents of the TAUJnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bits of the master and slave channel(s) must be identical.
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 100_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0100_B.
0	TAUJnMD0	Write 1_B.

(2) TAUJnCMURm for the slave channel(s)

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

Table 26.77 Contents of the TAUJnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the slave channel(s)

Table 26.78 Control Bit Settings for Synchronous Channel Output Mode 1

Bit Name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 1_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	0: Positive logic 1: Negative logic

(4) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 26.79 Simultaneous Rewrite Settings for the Slave Channel of the PWM Output Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting.

26.13.1.6 Operating Procedure for PWM Output Function

Table 26.80 Operating Procedure for PWM Output Function

	Operation	Status of TAUJn
Initial channel setting	Master channel: Set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in **Section 26.13.1.4, Register Settings for the Master Channel**. Slave channel: Set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in **Section 26.13.1.5, Register Settings for the Slave Channel(s)**. Set the values of the TAUJnCDRm registers of all channels.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm of the master and slave channels to 1 simultaneously. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUJnIm is generated on the master channel and TAUJTTOUTm (slave) is set.
During operation	TAUJnCDRm can be changed at any time. TAUJnCNTm and TAUJnRSF.TAUJnRSFm can be read at any time. TAUJnRDT.TAUJnRDTm can be changed during operation.	TAUJnCNTm of the master channel loads TAUJnCDRm and counts down. When the counter reaches 0000 0000_H: • INTTAUJnIm (master) is generated. • TAUJnCNTm (master) loads the TAUJnCDRm value and continues count operation. • TAUJnCNTm (slave) loads the TAUJnCDRm value and counts down. • TAUJTTOUTm (slave) is set to the active level. When TAUJnCNTm (slave) reaches 0000 0000_H: • INTTAUJnIm (slave) is generated. • TAUJTTOUTm (slave) is set to the inactive level.
Stop operation	Set TAUJnTT.TAUJnTTm of the master and slave channels to 1 simultaneously. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.

Restart operation

26.13.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

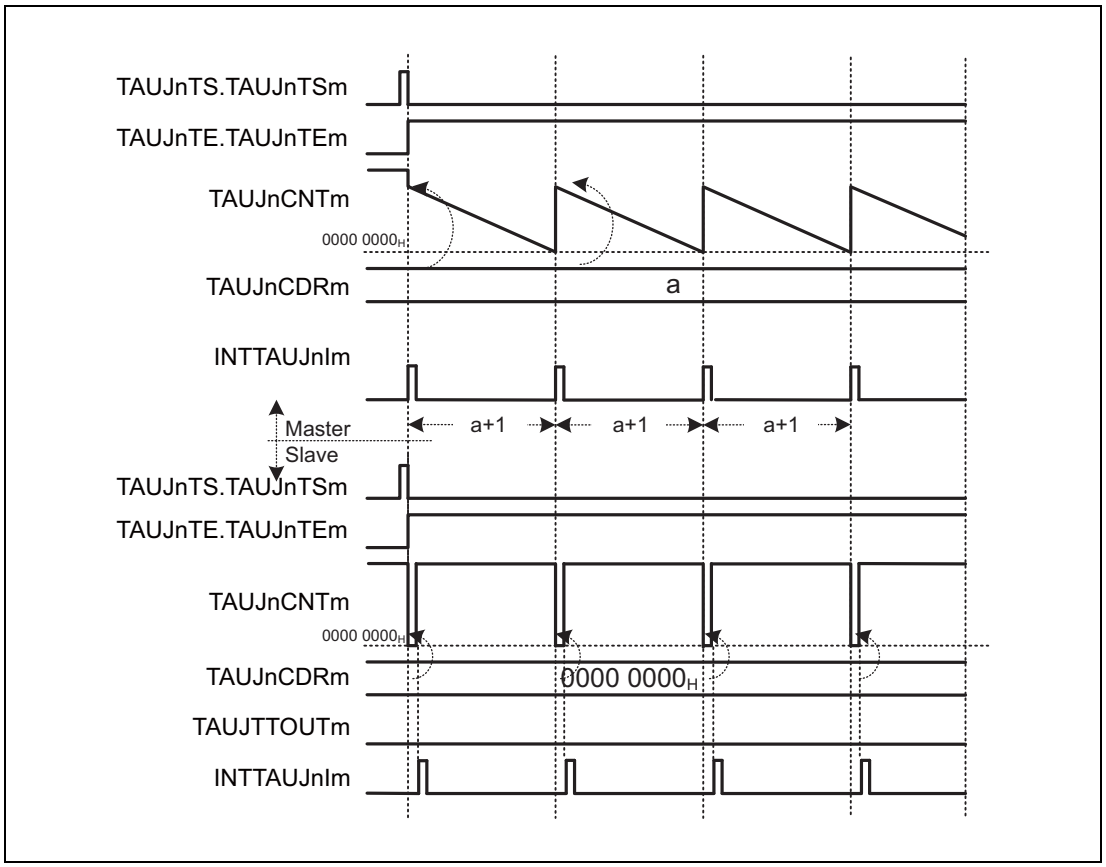


Figure 26.48 TAUJnCDRm (slave) = 0000 0000$_H$, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

- Every time the master channel generates an interrupt (INTTAUJnIm), 0000 0000$_H$ is loaded to TAUJnCNTm (slave). As a result, a slave channel interrupt (INTTAUJnIm) is generated at the same time and TAUJTTOUTm remains inactive.
- The value of TAUJnCDRm is loaded into TAUJnCNTm (slave) to generate an interrupt.

(2) Duty cycle = 100%

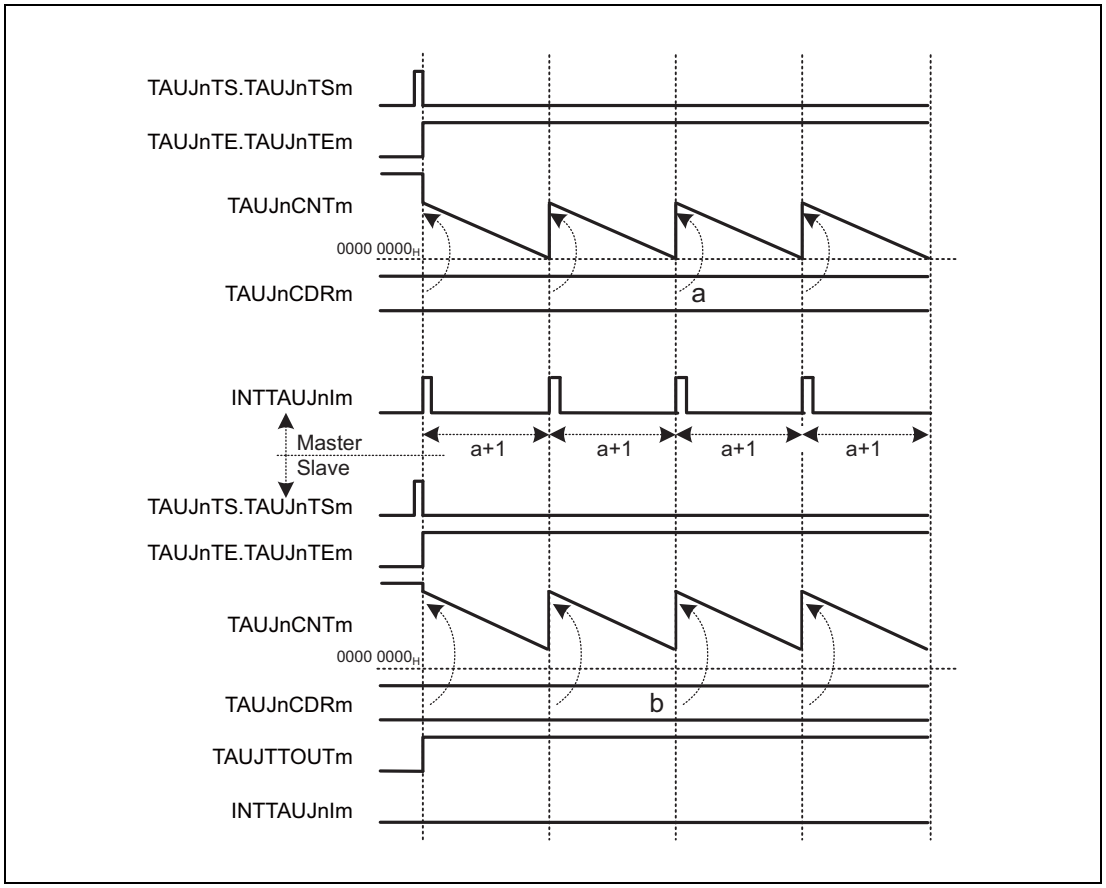


Figure 26.49 TAUJnCDRm (slave) ≥ TAUJnCDRm (master) + 1, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

If the TAUJnCDRm (slave) value is greater than the TAUJnCDRm (master) value, no interrupt occurs because the counter of the slave channel does not reach 0000 0000_H. TAUJTTOUTm remains active.

(3) Operation stop and restart

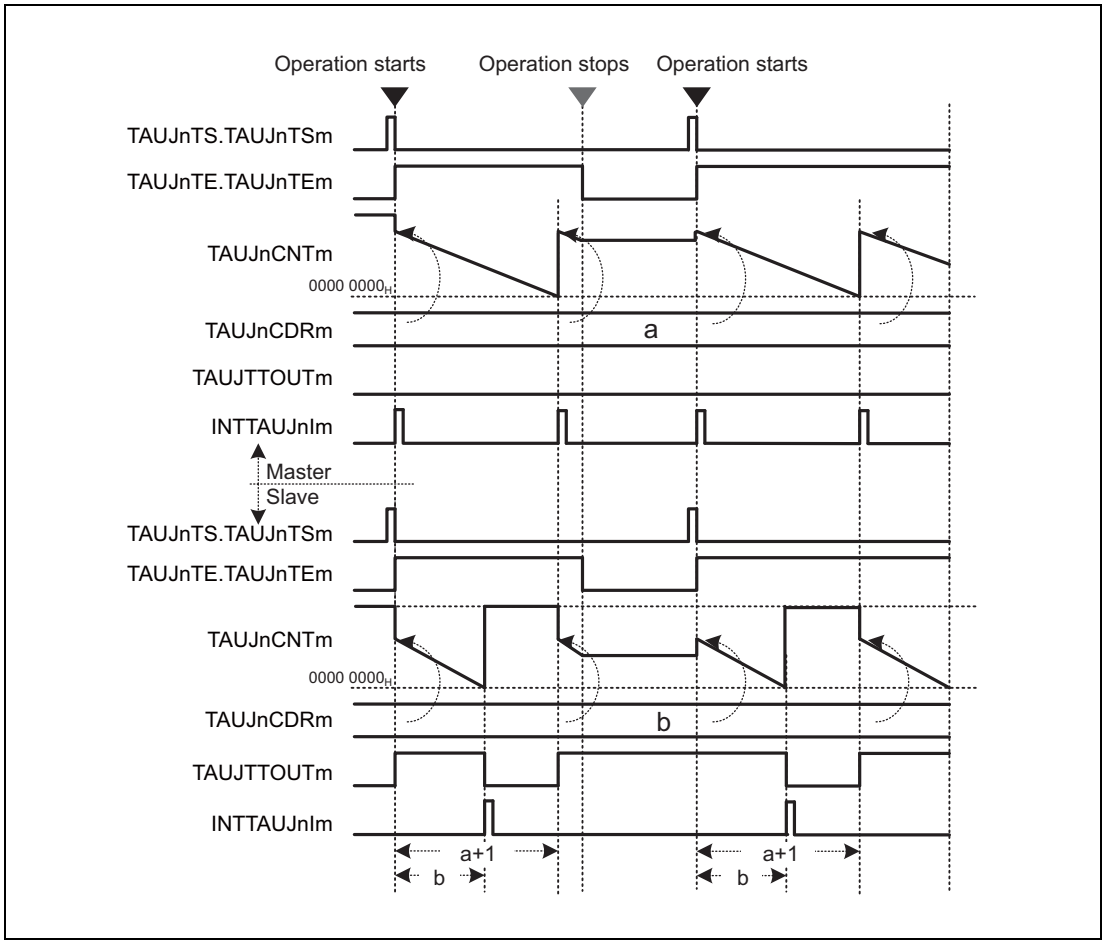


Figure 26.50 Operation Stop and Restart, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm of master and slave channels to 1. This sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJTTOUTm of every channel stop and retain their current values. No interrupt occurs.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm of master and slave channels to 1. The TAUJnCDRm value of master and slave channels is loaded into TAUJnCNTm. The counter starts to count down from this value.

Section 27 Real-Time Clock (RTCA)

This section contains a generic description of the Real-Time Clock (RTCA).

The first part of this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the RTCA.

27.1 Features of RH850/F1M RTCA

27.1.1 Number of Units and Channels

This microcontroller has the following number of RTCA units.

Each RTCA unit has one channel RTCA. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 27.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	1		
Name	RTCAn (n = 0)		

Table 27.2 Index

Index	Description
n	Throughout this section, the individual RTCA units are identified by the index “n” (n = 0); for example, RTCAnCTL0 is the RTCAn control register 0.

27.1.2 Register Base Address

RTCAn base address is listed in the following table.

RTCAn register addresses are given as offsets from the base address.

Table 27.3 Register Base Address

Base Address Name	Base Address
<RTCA0_base>	FFE7 8000_H

27.1.3 Clock Supply

The RTCAn clock supply is shown in the following table.

Table 27.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RTCA0	RTCATCKI	CKSCLK_ARTCA
	PCLK	CPUCLK2
	Register access clock	CPUCLK2

27.1.4 Interrupt Requests

RTCAn interrupt requests are listed in the following table.

Table 27.5 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
RTCA0			
RTCATINT1S	1-second interval interrupt	209	—
RTCATINTAL	Alarm interrupt	210	—
RTCATINTR	Fixed interval interrupt	211	—

27.1.5 Reset Sources

RTCAn reset sources are listed in the following table. RTCAn is initialized by these reset sources.

Table 27.6 Reset Sources

Unit Name	Reset Source
RTCA0	Power-up reset (PURES)

27.1.6 External Input/Output Signals

External input/output signals of RTCAn are listed below.

Table 27.7 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal Name
RTCA0		
RTCAT1HZ	1-Hz pulse output	RTCA0OUT*1

Note 1. RTCA0OUT is connected to TAUJ0. For details, see **Section 26, Timer Array Unit J (TAUJ)**.

27.2 Overview

27.2.1 Functional Overview

The Real-Time Clock (RTCA) has the following features:

- Count clock selection from 32 kHz to 4.194304 MHz
- Counters for years, months, day of the month, day of the week, hours, minutes, seconds, and a sub-counter. The calendar covers 99 years. Leap years are handled by hardware automatically.
- One Hz pulse output function
- Fixed interval interrupt function
- Alarm interrupt function
- Clock error correction function if a 32.768-kHz count clock is used

27.2.2 Block Diagram

The block diagram shows the main components of the RTCA.

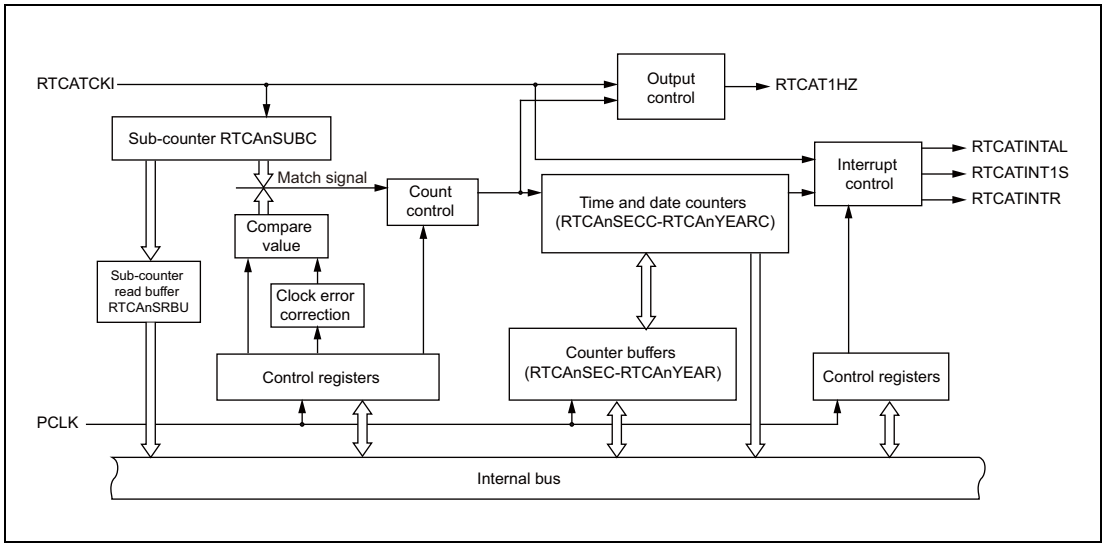


Figure 27.1 Block Diagram of the RTCA

27.2.3 Description of Blocks

The Real-Time Clock RTCA provides information about the present time and date and can generate wake-up signals (interrupts, alarms). This information is derived from the count clock RTCATCKI.

Sub-counter

RTCATCKI is the input to the sub-counter RTCAnSUBC. The sub-counter counts up from 0 until it reaches the compare value. The compare value is always defined as the frequency of RTCATCKI – 1 (in Hz). Thus, the sub-counter overflows after one second. It is then reset to 0 and triggers the seconds counter RTCAnSECC (and, if specified, the interrupt RTCATINT1S).

The sub-counter can generate a fixed interval interrupt every 0.25 seconds, 0.5 seconds, or 1 second, and a 1-Hz output pulse.

Time and date counters

The counters for minutes, hours, day of the week, day of the month, months, and years also count up. They have their own overflow limits. If all the lower counters overflow, the upper counter counts up.

The overflow limit of the counter for the day of the month (RTCAnDAYC) depends on the present month (28, 30, or 31 days) and (in February) on the year counter RTCAnYEARC (years 0, 4, 8, 12, etc. are considered leap years).

The hours counter RTCAnHOURC can be switched between 12- and 24-hour formats.

The counters for seconds, minutes, hours, day of the month, and months can generate a fixed interval interrupt upon overflow (RTCATINTR).

The counters for minutes, hours, and day of the week can also generate an alarm interrupt (RTCATINTAL), e.g. every Tuesday and Thursday at 10:32.

Counter buffers

All counters can be read directly at any time. The clock signal used to access the read/write registers and the count clock are usually asynchronous. An overflow of the sub-counter during the read operation can make all read values obsolete. Therefore, reading the counters must be performed using a special procedure. For details, see **Section 27.5.3, Reading Clock Counters**.

For reasons of synchronization, the counters cannot be written directly.

For reading and writing, all counters are accompanied by buffer registers. The buffer registers provide a synchronized way for reading the counters and for setting time and date. When they are used, the operation of the sub-counter must first be suspended and then re-activated (see also **Section 27.5.3, Reading Clock Counters** and **Section 27.5.2, Updating Clock Counters**.

The RTCAnTIMEC and RTCAnCALC registers and their corresponding buffer registers can be used to check and set the time (hours, minutes and seconds) or the date (day of the week, day of the month, month, and year) with one read/write operation.

27.3 Registers

27.3.1 List of Registers

RTCA registers are listed in the following table.

<RTCAn_base> is defined in **Section 27.1.2, Register Base Address**.

Table 27.8 List of Registers

Module Name	Register Name	Symbol	Address
Control registers			
RTCAn	Control register 0	RTCAnCTL0	<RTCAn_base> + 00_H
RTCAn	Control register 1	RTCAnCTL1	<RTCAn_base> + 04_H
RTCAn	Control register 2	RTCAnCTL2	<RTCAn_base> + 08_H
Sub-counter registers			
RTCAn	Sub-count register	RTCAnSUBC	<RTCAn_base> + $0C_H$
RTCAn	Sub-count register read buffer	RTCAnSRBU	<RTCAn_base> + 10_H
RTCAn	Clock error correction register	RTCAnSUBU	<RTCAn_base> + 38_H
RTCAn	Sub-counter compare register	RTCAnSCMP	<RTCAn_base> + $3C_H$
Clock counter and buffer registers			
RTCAn	Seconds count register	RTCAnSECC	<RTCAn_base> + $4C_H$
RTCAn	Seconds count buffer register	RTCAnSEC	<RTCAn_base> + 14_H
RTCAn	Minute count register	RTCAnMINC	<RTCAn_base> + 50_H
RTCAn	Minute count buffer register	RTCAnMIN	<RTCAn_base> + 18_H
RTCAn	Hour count register	RTCAnHOURC	<RTCAn_base> + 54_H
RTCAn	Hour count buffer register	RTCAnHOUR	<RTCAn_base> + $1C_H$
RTCAn	Day of the week count register	RTCAnWEEKC	<RTCAn_base> + 58_H
RTCAn	Day of the week count buffer register	RTCAnWEEK	<RTCAn_base> + 20_H
RTCAn	Day count register	RTCAnDAYC	<RTCAn_base> + $5C_H$
RTCAn	Day count buffer register	RTCAnDAY	<RTCAn_base> + 24_H
RTCAn	Month count register	RTCAnMONC	<RTCAn_base> + 60_H
RTCAn	Month count buffer register	RTCAnMONTH	<RTCAn_base> + 28_H
RTCAn	Year count register	RTCAnYEARC	<RTCAn_base> + 64_H
RTCAn	Year count buffer register	RTCAnYEAR	<RTCAn_base> + $2C_H$
Special counter and buffer registers			
RTCAn	Time count register	RTCAnTIMEC	<RTCAn_base> + 68_H
RTCAn	Time count buffer register	RTCAnTIME	<RTCAn_base> + 30_H
RTCAn	Calendar count register	RTCAnCALC	<RTCAn_base> + $6C_H$
RTCAn	Calendar count buffer register	RTCAnCAL	<RTCAn_base> + 34_H
Alarm time setting registers			
RTCAn	Alarm minute setting register	RTCAnALM	<RTCAn_base> + 40_H
RTCAn	Alarm hour setting register	RTCAnALH	<RTCAn_base> + 44_H
RTCAn	Alarm day of the week setting register	RTCAnALW	<RTCAn_base> + 48_H
Emulation register			
RTCAn	Emulation register	RTCAnEMU	<RTCAn_base> + 74_H

27.3.2 Details of RTCA Control Registers

27.3.2.1 RTCAnCTL0 — RTCA Control Register 0

This register controls the count operation of the sub-counter RTCAnSUBC, the format (12-hour/24-hour) of the hours counter RTCAnHOURC and the alarm hour setting register RTCAnALH, and the operation mode.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 00_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnCE	RTCAnCEST	RTCAnAMPM	RTCAnSLSB	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R	R	R	R

Table 27.9 RTCAnCTL0 Register Contents

Bit Position	Bit Name	Function
7	RTCAnCE	Starts/stops the sub-counter RTCAnSUBC operation. 0: Stops the sub-counter operation. All output pins and all status flags in control register RTCAnCTL2 are cleared. 1: Starts the sub-counter operation. The sub-counter counts up.
6	RTCAnCEST	Indicates the operation enabled/stopped status of the sub-counter: 0: Operation stopped status 1: Operation enabled status For details on how to use this status flag, see **Section 27.5.1, Initial Setting of the RTCA**.
5	RTCAnAMPM	Selects the format of the hours counter RTCAnHOURC and the alarm hour setting register RTCAnALH: 0: 12-hour format (1 to 12, am/pm) 1: 24-hour format (0 to 23, military time) For details on the format, see **Table 27.21, 12- and 24-Hour Format**.
4	RTCAnSLSB	Selects the operation mode: 0: 32.768 kHz mode 1: Frequency selection mode For details on the operation modes, see **Section 27.4, Operation**. The operation mode must not be changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1). For details on the initialization of RTCAn, see **Section 27.5.1, Initial Setting of the RTCA**.
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

27.3.2.2 RTCAnCTL1 — RTCA Control Register 1

This register controls the interrupt request generation and the 1-Hz pulse output.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 04_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnEN1HZ	RTCAnENALM	RTCAnEN1S	RTCAnCT2	RTCAnCT1	RTCAnCT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.10 RTCAnCTL1 Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	RTCAnEN1HZ	Enables/stops 1-Hz pulse output (RTCAT1HZ): 0: RTCAT1HZ disabled (RTCAT1HZ is fixed to 0) 1: RTCAT1HZ enabled
4	RTCAnENALM	Enables/disables alarm interrupt request generation (RTCATINTAL): 0: RTCATINTAL disabled 1: RTCATINTAL enabled
3	RTCAnEN1S	Enables/disables 1-second interrupt request generation (RTCATINT1S): 0: RTCATINT1S disabled 1: RTCATINT1S enabled
2 to 0	RTCAnCT[2:0]	Specifies the fixed interval interrupt request (RTCATINTR) setting:

	RTCATINTR Interrupt Request Generation	
RTCAnCT[2:0]	Interval	Timing
000	No interrupt request generation	
001	Every 0.25 seconds	Every 0.25, 0.5, 0.75 and 1 second
010	Every 0.5 seconds	Every 0.5 and 1 second
011	Every second	Every 1 second
100	Every minute	Every 1 minute 00 seconds
101	Every hour	Every 1 hour 00 minutes 00 seconds
110	Every day	Every 1 day 00 hours 00 minutes 00 seconds (i.e., every midnight)
111	Every month	Every 1 month first day 00 hours 00 minutes 00 seconds (i.e., every first midnight of a month)

If the settings of RTCAnCT[2:0] are changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), a glitch may be output to RTCATINTR. Implement appropriate interrupt mask processing procedures.

27.3.2.3 RTCAnCTL2 — RTCA Control Register 2

This register contains status information and controls the data transfer from the sub-counter RTCAnSUBC to the dedicated sub-counter read buffer RTCAnSRBU and the operation setting of the clock counters (RTCAnSECC to RTCAnYEARC).

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnWUST	RTCAnWSST	RTCAnRSST	RTCAnRSUB	RTCAnWST	RTCAnWAIT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R/W

Table 27.11 RTCAnCTL2 Register Contents (1/2)

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	RTCAnWUST	Indicates whether RTCAnSUBU write operation has been completed: 0: RTCAnSUBU write completed 1: RTCAnSUBU write in progress The write operation ends with the next sub-counter overflow. While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1) and if write operation to RTCAnSUBU is completed, this bit is set to 1. See **Section 27.5.5, Writing to RTCAnSUBU**, for details.
4	RTCAnWSST	Indicates whether RTCAnSCMP write operation has been completed: 0: RTCAnSCMP write completed 1: RTCAnSCMP write in progress The write operation ends with the next sub-counter overflow. While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1) and if write operation to RTCAnSCMP is completed, this bit is set to 1. See **Section 27.5.6, Writing to RTCAnSCMP**, for details.
3	RTCAnRSST	Indicates whether the value of the sub-counter (RTCAnSUBC) has been transferred to the sub-count register read buffer (RTCAnSRBU): 0: Transfer in progress, or waiting for a transfer trigger 1: Transfer completed This bit is cleared (transfer is triggered) by RTCAnRSUB=1.This bit is automatically set when the transfer is completed. See **Section 27.5.4, Reading RTCAnSRBU**, for details.
2	RTCAnRSUB	Triggers transfer of the value of the sub-counter (RTCAnSUBC) to the dedicated read buffer (RTCAnSRBU) or clears the transfer state of the sub-counter: 0: Transfer status (RTCAnRSST) is cleared. 1: Transfer is triggered. This bit is used to read the value of RTCAnSRBU when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1). The value of RTCAnSUBC is synchronized with RTCATCKI and loaded to RTCAnSRBU. For details, see **Section 27.5.4, Reading RTCAnSRBU**.
1	RTCAnWST	Indicates the status of all clock counters (RTCAnSECC to RTCAnYEARC): 0: All clock counters are running. 1: All clock counters are stopped The sub-counter is still running. The clock counters must be stopped before reading or writing clock counter values during sub-counter operation (RTCAnCTL0.RTCAnCE = 1). To stop the clock counters, set RTCAnWAIT = 1.

Table 27.11 RTCAnCTL2 Register Contents (2/2)

Bit Position	Bit Name	Function
0	RTCAnWAIT	Restarts/stops all clock counters (RTCAnSECC to RTCAnYEARC): 0: Restarts all clock counters either immediately or immediately after the clock counter write operation finishes. 1: Stops all clock counters temporarily. The sub-counter is still running. The clock counters must be stopped before reading or writing counter buffers during sub-counter operation (RTCAnCTL0.RTCAnCE = 1). **CAUTION** Only one overflow can be held internally. When two overflows occur, the seconds counter is incremented only by one when it is restarted. Thus, the procedure must be completed within one second.

27.3.3 Details of RTCA Sub-Counter Registers

27.3.3.1 RTCAnSUBC — RTCA Sub-Count Register

This counter counts the 1-second reference time. It operates using the count clock RTCATCKI.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RTCAn_base> + $0C_H$

Value after reset: $0000\ 0000_H$

This register is initialized:

– When write operation is performed to the seconds count buffer register (RTCAnSEC) or to the time count buffer register (RTCAnTIME) and the value is reflected to the seconds count register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSUBC[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSUBC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.12 RTCAnSUBC Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 0	RTCAnSUBC [21:0]	Sub-counter value The sub-counter only operates while RTCAnCTL0.RTCAnCEST = 1.

NOTES

1. This sub-counter operates with RTCATCKI while the read operation is clocked by PCLK. Reading this sub-counter during operation (RTCAnCTL0.RTCAnCEST = 1) is asynchronous to RTCATCKI and can lead to wrong results.
   Use the sub-count register read buffer (RTCAnSRBU) to read the sub-counter value during operation.
   For details, see **Section 27.5.4, Reading RTCAnSRBU**.
2. The count-operation of this sub-counter depends on the selected operation mode. See **Section 27.4, Operation**, for details.

27.3.3.2 RTCAnSRBU — RTCA Sub-Count Register Read Buffer

This register is the read buffer for the sub-counter RTCAnSUBC.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RTCAn_base> + 10_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSRBU[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSRBU[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.13 RTCAnSRBU Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 0	RTCAnSRBU [21:0]	Sub-counter value at the time of the last RTCAnSUBC read. When RTCAnCTL2.RTCAnRSUB is set to 1, the value of the RTCAnSUBC is loaded to the read buffer in synchronization with RTCATCKI.

NOTE

Perform RTCAnSRBU read according to the flow described in **Section 27.5.4, Reading RTCAnSRBU**.

27.3.3.3 RTCAnSUBU — RTCA Clock Error Correction Register

This register enables and specifies clock error correction. This register only applies in 32.768-kHz mode (RTCAnCTL0.RTCAnSLSB = 0).

For details on clock error correction, see **Section 27.4.4, Clock Error Correction**.

Access: This register can be read or written in 8-bit units.
Note the following when writing this register during sub-counter operation:
- Previous RTCAnSUBU write must be completed (RTCAnCTL2.RTCAnWUST = 0).
- The write operation ends with the next sub-counter overflow.

Address: <RTCAn_base> + 38_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnDEV	RTCAnF6	RTCAnF[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.14 RTCAnSUBU Register Contents

Bit Position	Bit Name	Function
7	RTCAnDEV	Specifies how often clock error correction is performed per minute: 0: Three times every minute (when RTCAnSECC equals 00, 20, and 40) 1: Once every minute (when RTCAnSECC equals 00)
6	RTCAnF6	Specifies whether the sub-counter value is incremented or decremented: 0: Incremented (+ correction) Incrementation value = (RTCAnF[5:0] value – 1) × 2 1: Decremented (– correction) Decrementation value = (inverted data of RTCAnF[5:0] value + 1) × 2
5 to 0	RTCAnF[5:0]	Error correction value

NOTES

1. When RTCAnF[5:1] = 00000_B, clock error correction is not performed.
2. Perform RTCAnSUBU write as described in
   - **Section 27.5.1, Initial Setting of the RTCA**, and
   - **Section 27.5.5, Writing to RTCAnSUBU**.

27.3.3.4 RTCAnSCMP — RTCA Sub-Counter Compare Register

This register sets the compare value of the sub-counter RTCAnSUBC in frequency selection mode (RTCAnCTL0.RTCAnSLSB = 1).

When the sub-counter values matches the value of this register, an overflow signal is output to the seconds counter RTCAnSECC and the sub-counter is cleared.

Set the value for this register according to the frequency of the input clock RTCATCKI.

Access: This register can be read or written in 32-bit units.
Note the following when writing this register during sub-counter operation:
- Previous RTCAnSCMP write must be completed (RTCAnCTL2.RTCAnWSST = 0).
- The write operation ends with the next sub-counter overflow.

Address: <RTCAn_base> + $3C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSCMP[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.15 RTCAnSCMP Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 0	RTCAnSCMP [21:0]	Sub-counter compare value in frequency selection mode.

Example

The following example illustrates the setting of RTCAnSCMP:

- RTCATCKI = 4 MHz = 4,000,000 Hz
- RTCAnSCMP = 4,000,000 – 1 = 3,999,999 (decimal code) = $3D08FF_H$
- The seconds counter RTCAnSECC is triggered when the sub-counter value changes from $3D08FF_H$ to 0_H.

NOTES

1. The operation of the RTCA cannot be guaranteed if a value of 3198 (decimal code) or lower is set in this register.
2. Perform RTCAnSCMP write as described in **Section 27.5.1, Initial Setting of the RTCA** and **Section 27.5.6, Writing to RTCAnSCMP**.

27.3.4 Details of RTCA Clock Counter and Buffer Registers

27.3.4.1 RTCAnSECC — RTCA Seconds Count Register

This register is the seconds counter. It counts seconds from 00 to 59 in BCD.

This register counts as follows.

- It is triggered by every overflow of the sub-counter RTCAnSUBC.

  If the sub-counter overflows while the seconds counter is stopped (RTCAnCTL2.RTCAnWST = 1), the seconds counter behaves as follows:

  – If one sub-counter overflow occurs while the seconds counter is stopped, the overflow is held internally.
  The seconds counter is incremented by one when it is restarted.

  – If two or more overflows occur while the seconds counter is stopped, the overflow count cannot be held internally.
  The seconds counter is incremented by one when it is restarted.

  – If the seconds counter was updated while the seconds counter is stopped, the sub-counter overflow(s) are ignored.

- It outputs an overflow signal when the value changes from 59 to 00. The overflow signal triggers the minutes counter (RTCAnMINC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + $4C_H$

Value after reset: 00_H

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

Table 27.16 RTCAnSECC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnSECC [6:0]	Seconds in BCD

NOTES

1. Perform RTCAnSECC read according to the flow described in **Section 27.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the seconds count buffer register RTCAnSEC or to the clock time setting register RTCAnTIME. See
   - **Section 27.5.1, Initial Setting of the RTCA**, and
   - **Section 27.5.2, Updating Clock Counters**

27.3.4.2 RTCAnSEC — RTCA Seconds Count Buffer Register

This register is a buffer register to read/write the seconds counter RTCAnSECC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 14_H

Value after reset: 00_H

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

Table 27.17 RTCAnSEC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnSEC [6:0]	Seconds in BCD

NOTES

1. When writing this register, only decimal values between 00 and 59 in BCD are allowed.
2. Perform RTCAnSEC read/write as described in
   - **Section 27.5.1, Initial Setting of the RTCA,**
   - **Section 27.5.2, Updating Clock Counters**, and
   - **Section 27.5.3, Reading Clock Counters.**

27.3.4.3 RTCAnMINC — RTCA Minutes Count Register

This register is the minutes counter. It counts minutes from 00 to 59 in BCD.

This register counts as follows.

- It is triggered by every overflow of the seconds counter RTCAnSECC.
- It outputs an overflow signal when the value changes from 59 to 00. The overflow signal triggers the hours counter (RTCAnHOURC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 50_H

Value after reset: 00_H

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

Table 27.18 RTCAnMINC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnMINC [6:0]	Minutes in BCD

NOTES

1. Perform RTCAnMINC read according to the flow described in **Section 27.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the minutes count buffer register RTCAnMIN or to the time count buffer register RTCAnTIME. See
   - **Section 27.5.1, Initial Setting of the RTCA**, and
   - **Section 27.5.2, Updating Clock Counters.**

27.3.4.4 RTCAnMIN — RTCA Minutes Count Buffer Register

This register is a buffer register to read/write the minutes counter RTCAnMINC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 18_H

Value after reset: 00_H

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

Table 27.19 RTCAnMIN Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnMIN [6:0]	Minutes in BCD

NOTES

1. When writing this register, only decimal values between 00 and 59 in BCD are allowed.
2. Perform RTCAnMIN read/write as described in
   - **Section 27.5.1, Initial Setting of the RTCA,**
   - **Section 27.5.2, Updating Clock Counters**, and
   - **Section 27.5.3, Reading Clock Counters.**

27.3.4.5 RTCAnHOURC — RTCA Hours Count Register

This register is the hours counter. It counts the hours in BCD. The count range depends on the selected hour format. See **Table 27.21, 12- and 24-Hour Format**.

This register counts as follows.

- It is triggered by every overflow of the minutes counter RTCAnMINC.
- It outputs an overflow signal when the value changes from 23 to 00 (in 24-hour format) or from 31 to 12 (in 12-hour format). The overflow signal triggers two counters:
  - Day of the week counter (RTCAnWEEKC)
  - Day of the month counter (RTCAnDAYC)

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 54_H

Value after reset: 12_H

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

Table 27.20 RTCAnHOURC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	RTCAnHOURC [5:0]	Hours in BCD. See **Table 27.21, 12- and 24-Hour Format**, for details.

NOTES

1. Perform RTCAnHOURC read according to the flow described in **Section 27.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the hours count buffer register RTCAnHOUR or to the time count buffer register RTCAnTIME. See
   - **Section 27.5.1, Initial Setting of the RTCA**, and
   - **Section 27.5.2, Updating Clock Counters.**

12- or 24-hour format

The count values of RTCAnHOURC depend on the selected hour format.

If 12-hour format is selected (RTCAnCTL0.RTCAnAMPM = 0), bit 5 in the RTCAnHOURC register is the am/pm indicator:

- RTCAnHOURC[5] = 0: am
- RTCAnHOURC[5] = 1: pm

The following table shows the count range of RTCAnHOURC in both 12- and 24-hour format.

Table 27.21 12- and 24-Hour Format

12-Hour Format (RTCAnAMPM = 0)			24-Hour Format (RTCAnAMPM = 1)	
Time	RTCAnHOURC		Time	RTCAnHOURC
0 am	12_H		0	00_H
1 am	01_H		1	01_H
2 am	02_H		2	02_H
3 am	03_H		3	03_H
4 am	04_H		4	04_H
5 am	05_H		5	05_H
6 am	06_H		6	06_H
7 am	07_H		7	07_H
8 am	08_H		8	08_H
9 am	09_H		9	09_H
10 am	10_H		10	10_H
11 am	11_H		11	11_H
0 pm	32_H	↓ pm indicator in 12-hour format: RTCAnHOUR.RTCAnHOUR[5] = 1	12	12_H
1 pm	21_H		13	13_H
2 pm	22_H		14	14_H
3 pm	23_H		15	15_H
4 pm	24_H		16	16_H
5 pm	25_H		17	17_H
6 pm	26_H		18	18_H
7 pm	27_H		19	19_H
8 pm	28_H		20	20_H
9 pm	29_H		21	21_H
10 pm	30_H		22	22_H
11 pm	31_H		23	23_H

27.3.4.6 RTCAnHOUR — RTCA Hours Count Buffer Register

This register is a buffer register to read/write the hours counter RTCAnHOURC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + $1C_H$

Value after reset: 12_H

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

Table 27.22 RTCAnHOUR Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RTCAnHOUR [5:0]	Hours in BCD See **Table 27.21, 12- and 24-Hour Format**, for details.

NOTES

1. When writing this register, only the following decimal values in BCD are allowed:
   - 12-hour format (RTCAnCTL0.RTCAnAMPM = 0):
     01 to 12 or 21 to 32
   - 24-hour format (RTCAnCTL0.RTCAnAMPM = 1):
     00 to 23
2. Perform RTCAnHOUR read/write as described in
   - **Section 27.5.1, Initial Setting of the RTCA,**
   - **Section 27.5.2, Updating Clock Counters**, and
   - **Section 27.5.3, Reading Clock Counters.**

27.3.4.7 RTCAnWEEKC — RTCA Day of the Week Count Register

This register is the day of the week counter. It counts from 0 to 6.

This register counts as follows.

- It is triggered by every overflow of the hours counter RTCAnHOURC.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 58_H

Value after reset: 00_H

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

Table 27.23 RTCAnWEEKC Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	RTCAnWEEKC [2:0]	Day of the week

NOTES

1. Perform RTCAnWEEKC read according to the flow described in **Section 27.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the day of the week count buffer register RTCAnWEEK or to the calendar count buffer register RTCAnCAL. See
   - **Section 27.5.1, Initial Setting of the RTCA**, and
   - **Section 27.5.2, Updating Clock Counters.**

27.3.4.8 RTCAnWEEK — RTCA Day of the Week Count Buffer Register

This register is a buffer register to read/write the day of the week counter RTCAnWEEKC.

There is no particular correspondence between the value of RTCAnWEEK and the day of the week. Set the correspondence according to the application to be used.

Example: 0 = Sunday, 1 = Monday, …, 6 = Saturday

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 20_H

Value after reset: 00_H

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

Table 27.24 RTCAnWEEK Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	RTCAnWEEK [2:0]	Day of the week

NOTES

1. When writing this register, only decimal values between 0 and 6 in BCD are allowed.
2. Perform RTCAnWEEK read/write as described in
   - **Section 27.5.1, Initial Setting of the RTCA,**
   - **Section 27.5.2, Updating Clock Counters**, and
   - **Section 27.5.3, Reading Clock Counters.**

27.3.4.9 RTCAnDAYC — RTCA Day of the Month Count Register

This register is the day of the month counter. It counts from 01 to a maximum of 31 in BCD, depending on the value of the month counter (RTCAnMONC) and the year counter (RTCAnYEARC):

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, non-leap year)

Years 0, 4, 8, 12, etc., are considered leap years.

This register counts as follows.

- It is triggered by every overflow of the hours counter RTCAnHOURC.
- It outputs an overflow signal when the value changes from 28, 29, 30, or 31 to 01, depending on the current month and year. The overflow signal triggers the month counter (RTCAnMONC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + $5C_H$

Value after reset: 01_H

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

Table 27.25 RTCAnDAYC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	RTCAnDAYC [5:0]	Day of the month in BCD

NOTES

1. Perform RTCAnDAYC read according to the flow described in **Section 27.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the day of the month count buffer register RTCAnDAY or to the calendar count buffer register RTCAnCAL. See
   - **Section 27.5.1, Initial Setting of the RTCA**, and
   - **Section 27.5.2, Updating Clock Counters.**

27.3.4.10 RTCAnDAY — RTCA Day of the Month Count Buffer Register

This register is a buffer register to read/write the day of the month counter RTCAnDAYC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 24_H

Value after reset: 01_H

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

Table 27.26 RTCAnDAY Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RTCAnDAY [5:0]	Day of the month in BCD

NOTES

1. When writing this register, only decimal values between 01 and 31 in BCD are allowed:
   - 01 to 31 (January, March, May, July, August, October, December)
   - 01 to 30 (April, June, September, November)
   - 01 to 29 (February, leap year)
   - 01 to 28 (February, non-leap year)
2. Perform RTCAnDAY read/write as described in
   - **Section 27.5.1, Initial Setting of the RTCA,**
   - **Section 27.5.2, Updating Clock Counters**, and
   - **Section 27.5.3, Reading Clock Counters.**

27.3.4.11 RTCAnMONC — RTCA Month Count Register

This register is the month counter. It counts the month of the year, starting from 01 to 12 in BCD.

This register counts as follows.

- It is triggered by every overflow of the counter for the day of the month RTCAnDAYC.
- It outputs an overflow signal when the value changes from 12 to 01. The overflow signal triggers the year counter (RTCAnYEARC).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 60_H

Value after reset: 01_H

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

Table 27.27 RTCAnMONC Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned.
4 to 0	RTCAnMONC [4:0]	Month of the year in BCD

NOTES

1. Perform RTCAnMONC read according to the flow described in **Section 27.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the month count buffer register RTCAnMONTH or to the calendar count buffer register RTCAnCAL. See
   - **Section 27.5.1, Initial Setting of the RTCA**, and
   - **Section 27.5.2, Updating Clock Counters.**

27.3.4.12 RTCAnMONTH — RTCA Month Count Buffer Register

This register is a buffer register to read/write the month counter RTCAnMONC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 28_H

Value after reset: 01_H

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

Table 27.28 RTCAnMONTH Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 0	RTCAnMONTH [4:0]	Month of the year in BCD

NOTES

1. When writing this register, only decimal values between 01 and 12 in BCD are allowed.
2. Perform RTCAnMONTH read/write as described in
   - **Section 27.5.1, Initial Setting of the RTCA,**
   - **Section 27.5.2, Updating Clock Counters**, and
   - **Section 27.5.3, Reading Clock Counters.**

27.3.4.13 RTCAnYEARC — RTCA Year Count Register

This register is the year counter. It counts years from 00 to a maximum of 99 in BCD.

Years 00, 04, 08, …, 92, and 96 (every four years) are considered leap years.

This register counts as follows.

- It is triggered by every overflow of the month counter RTCAnMONC.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <RTCAn_base> + 64_H

Value after reset: 00_H

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

Table 27.29 RTCAnYEARC Register Contents

Bit Position	Bit Name	Function
7 to 0	RTCAnYEARC [7:0]	Year in BCD

NOTES

1. Perform RTCAnYEARC read according to the flow described in **Section 27.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the year count buffer register RTCAnYEAR or to the calendar count buffer register RTCAnCAL. See
   - **Section 27.5.1, Initial Setting of the RTCA**, and
   - **Section 27.5.2, Updating Clock Counters.**

27.3.4.14 RTCAnYEAR — RTCA Year Count Buffer Register

This register is a buffer register to read/write the year counter RTCAnYEARC.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + $2C_H$

Value after reset: 00_H

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

Table 27.30 RTCAnYEAR Register Contents

Bit Position	Bit Name	Function
7 to 0	RTCAnYEAR [7:0]	Year in BCD

NOTES

1. When writing this register, only decimal values between 00 and 99 in BCD are allowed.
2. Perform RTCAnYEAR read/write as described in
   - **Section 27.5.1, Initial Setting of the RTCA,**
   - **Section 27.5.2, Updating Clock Counters**, and
   - **Section 27.5.3, Reading Clock Counters.**

27.3.5 Details of RTCA Special Counter and Buffer Registers

27.3.5.1 RTCAnTIMEC — RTCA Time Count Register

This register enables the RTCAnHOURC, RTCAnMINC, and RTCAnSECC counters to be read simultaneously.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RTCAn_base> + 68_H

Value after reset: $0012\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnHOURC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RTCAnMINC[6:0]							—	RTCAnSECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.31 RTCAnTIMEC Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 16	RTCAnHOURC [5:0]	Hours in BCD. See **Table 27.21, 12- and 24-Hour Format**, for details.
15	Reserved	When read, the value after reset is returned.
14 to 8	RTCAnMINC [6:0]	Minutes in BCD
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnSECC [6:0]	Seconds in BCD

NOTES

1. Perform RTCAnTIMEC read according to the flow described in **Section 27.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the time count buffer register RTCAnTIME. See
   - **Section 27.5.1, Initial Setting of the RTCA**, and
   - **Section 27.5.2, Updating Clock Counters.**

27.3.5.2 RTCAnTIME — RTCA Time Count Buffer Register

This register enables the RTCAnHOUR, RTCAnMIN, and RTCAnSEC buffer registers to be read/written simultaneously.

Access: This register can be read or written in 32-bit units.

Address: <RTCAn_base> + 30_H

Value after reset: $0012\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnHOUR[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RTCAnMIN[6:0]							—	RTCAnSEC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.32 RTCAnTIME Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21 to 16	RTCAnHOUR [5:0]	Hours in BCD See **Table 27.21, 12- and 24-Hour Format**, for details.
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14 to 8	RTCAnMIN [6:0]	Minutes in BCD
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnSEC [6:0]	Seconds in BCD

NOTE

Perform RTCAnTIME read/write as described in

- **Section 27.5.1, Initial Setting of the RTCA,**
- **Section 27.5.2, Updating Clock Counters**, and
- **Section 27.5.3, Reading Clock Counters.**

27.3.5.3 RTCAnCALC — RTCA Calendar Count Register

This register enables the RTCAnYEARC, RTCAnMONC, RTCAnDAYC, and RTCAnWEEKC counters to be read simultaneously.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <RTCAn_base> + $6C_H$

Value after reset: $0001\ 0100_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RTCAnYEARC[7:0]								—	—	—	RTCAnMONC[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RTCAnDAYC[5:0]						—	—	—	—	—	RTCAnWEEKC[2:0]		
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.33 RTCAnCALC Register Contents

Bit Position	Bit Name	Function
31 to 24	RTCAnYEARC [7:0]	Year in BCD
23 to 21	Reserved	When read, the value after reset is returned.
20 to 16	RTCAnMONC [4:0]	Month of the year in BCD
15, 14	Reserved	When read, the value after reset is returned.
13 to 8	RTCAnDAYC [5:0]	Day of the month in BCD
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	RTCAnWEEKC [2:0]	Day of the week in BCD

NOTES

1. Perform RTCAnCALC read according to the flow described in **Section 27.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the clock time setting register RTCAnCAL. See
   - **Section 27.5.1, Initial Setting of the RTCA**, and
   - **Section 27.5.2, Updating Clock Counters.**

27.3.5.4 RTCAnCAL — RTCA Calendar Count Buffer Register

This register enables the RTCAnYEAR, RTCAnMONTH, RTCAnDAY, and RTCAnWEEK buffer registers to be read/written simultaneously.

Access: This register can be read or written in 32-bit units.

Address: <RTCAn_base> + 34_H

Value after reset: $0001\ 0100_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RTCAnYEAR[7:0]								—	—	—	RTCAnMONTH[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RTCAnDAY[5:0]						—	—	—	—	—	RTCAnWEEK[2:0]		
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 27.34 RTCAnCAL Register Contents

Bit Position	Bit Name	Function
31 to 24	RTCAnYEAR [7:0]	Year in BCD
23 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 16	RTCAnMONTH [4:0]	Month of the year in BCD
15, 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13 to 8	RTCAnDAY [5:0]	Day of the month in BCD
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	RTCAnWEEK [2:0]	Day of the week in BCD

NOTE

Perform RTCAnCAL read/write as described in

- **Section 27.5.1, Initial Setting of the RTCA.**
- **Section 27.5.2, Updating Clock Counters**, and
- **Section 27.5.3, Reading Clock Counters.**

27.3.6 Details of RTCA Alarm Setting Registers

27.3.6.1 RTCAnALM — RTCA Alarm Minute Setting Register

This register specifies the minute of the alarm interrupt.

For details and example settings, see **Section 27.4.3, Alarm Interrupt Function**.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 40_H

Value after reset: 00_H

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

Table 27.35 RTCAnALM Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnALM [6:0]	Minute of the alarm interrupt in BCD

NOTES

1. If decimal values outside the range of 00 to 59 in BCD are set, no alarm interrupt request will be generated.
2. When the setting of RTCAnALM is changed during sub-counter operation (RTCAnCTL0.RTCAnCEST = 1), a glitch may be output to RTCATINTAL. Implement appropriate interrupt mask processing procedures.

27.3.6.2 RTCAnALH — RTCA Alarm Hour Setting Register

This register specifies the hour of the alarm interrupt.

For details and example settings, see **Section 27.4.3, Alarm Interrupt Function**.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 44_H

Value after reset: 12_H

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

Table 27.36 RTCAnALH Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	RTCAnALH [5:0]	Hour of the alarm interrupt in BCD

NOTES

1. If decimal values outside the following range are set, no alarm interrupt request will be generated:
   - 12-hour format (RTCAnCTL0.RTCAnAMPM = 0): 01 to 12 or 21 to 32
   - 24-hour format (RTCAnCTL0.RTCAnAMPM = 1): 00 to 23
2. When the setting of RTCAnALH is changed during sub-counter operation (RTCAnCTL0.RTCAnCEST = 1), a glitch may be output to RTCATINTAL. Implement appropriate interrupt mask processing procedures.

27.3.6.3 RTCAnALW — RTCA Alarm Day of the Week Setting Register

This register specifies the day(s) of the week of the alarm interrupt.

Access: This register can be read or written in 8-bit units.

Address: <RTCAn_base> + 48_H

Value after reset: 00_H

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

Table 27.37 RTCAnALW Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	RTCAnALW [6:0]	Specifies day of the week m (m = 0 to 6) as a day, when an alarm interrupt request is generated: 0: No alarm interrupt request is generated on day m. 1: Alarm interrupt request is generated on day m at the time set using RTCAnALM and RTCAnALH. The bits of this register correspond to the count value of the day of the week counter (RTCAnWEEKC).

NOTE

When the setting of RTCAnALW is changed during sub-counter operation (RTCAnCTL0.RTCAnCE = 1), a glitch may be output to RTCATINTAL. Implement appropriate interrupt mask processing procedures.

Example

If Sunday is RTCAnWEEK = 0, Monday is RTCAnWEEK = 1, Tuesday is RTCAnWEEK = 2, …, Saturday is RTCAnWEEK = 6:

- To set the alarm for Sunday, set RTCAnALW = 0000 0001_B.
- To set the alarm for Monday and Wednesday, set RTCAnALW = 0000 1010_B.
- To set the alarm for Tuesday, Thursday, and Saturday, set RTCAnALW = 0101 0100_B.

For more examples, see **Section 27.4.3, Alarm Interrupt Function**.

27.3.7 RTCA Emulation Register

27.3.7.1 RTCAnEMU — RTCA Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read or written in 8- or 1-bit units.
A write should be performed when EPC.SVSTOP = 0.

Address: <RTCAn_base> + 74_H

Value after reset: 00_H

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

Table 27.38 RTCAnEMU Register Contents

Bit Position	Bit Name	Function
7	RTCAnSVSDIS	When the EPC.SVSTOP bit is set to 0: Count clock at a breakpoint, etc. when the debugger gains microcontroller control (as at a breakpoint) regardless of the value of this bit. When the EPC.SVSTOP bit is set to 1: 0: Count clock is stopped when the debugger gains microcontroller control (as at a breakpoint). 1: Count clock continues to be supplied when the debugger gains microcontroller control (at a breakpoint, etc.).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

27.4 Operation

The RTCA provides two operation modes:

- Frequency selection mode
- 32.768-kHz mode

The operation mode that can be used depends on the available input clock RTCATCKI. The operation mode specifies the sub-counter compare value that is used to trigger the seconds counter and thus all subsequent counters. Clock error correction is only possible in 32.768-kHz mode.

The following table provides an overview of the properties of the two operation modes.

Table 27.39 RTCA Operation Mode Overview

		32.768-kHz Mode	
	Frequency Selection Mode	**Clock Correction Disabled**	**Clock Correction Enabled**
Allowed input clock RTCATCKI	Any frequency from 32 kHz to 4.194304 MHz	32.768 kHz	Any frequency from 32.76180000 kHz to 32.77420000 kHz
Sub-counter RTCAnSUBC operation	• Counter overflow at value of RTCAnSCMP • RTCAnSCMP must be set to RTCATCKI-1 (in Hz)	Counter overflow at $7FFF_H$	Counter overflow at $7FFF_H$ or Every 20 or 60 seconds: $7FFF_H$ ±RTCAnSUBU.RTCAnF[5:0]

The operation mode is selected by control bit RTCAnCTL0.RTCAnSLSB. For details on how to set the operation mode during RTCA initialization, see **Section 27.5.1, Initial Setting of the RTCA**.

CAUTIONS

1. **The input clock RTCATCKI must not be outside the allowed frequency range.**
2. **The operation mode must not be changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1).**

27.4.1 Clock Counter Format

The clock counters (RTCAnSECC to RTCAnYEARC) operate on binary coded decimals (BCD): Each digit is represented by its own binary sequence.

Depending on the valid data range, the number of bits for a digit differs. For example, the tens digit of the month of the year counter has only one bit (for 0 and 1) whereas the tens digit of the minutes counter has 3 bits (for 0 to 5).

The following table lists the decimals 0 to 59 in binary and BCD.

Table 27.40 Example of BCD Code – Seconds or Minutes Counter (0 to 59)

Decimal	Binary	BCD
0	000000	000 0000
1	000001	000 0001
2	000010	000 0010
3	000011	000 0011
4	000100	000 0100
5	000101	000 0101
6	000110	000 0110
7	000111	000 0111
8	001000	000 1000
9	001001	000 1001
10	001010	001 0000
11	001011	001 0001
12	001100	001 0010
:	:	:
58	111010	101 1000
59	111011	101 1001

27.4.2 Fixed Interval Interrupt Function

Interrupt RTCATINTR can be specified to occur after every 0.25 seconds, 0.5 seconds, 1 (full) second, 1 (full) minute, 1 (full) hour, 1 (full) day, or 1 (full) month.

The fixed interval interrupt function is controlled by bits RTCAnCTL1.RTCAnCT[2:0].

27.4.3 Alarm Interrupt Function

Interrupt RTCATINTAL can be specified to occur at a certain time on one or several days of the week. This interrupt can be used as a wake-up signal.

The alarm interrupt function is enabled and disabled by bit RTCAnCTL1.RTCAnENALM.

The alarm setting is specified by the following control registers:

- RTCAnALW selects the weekday(s).

The allocation of bits to weekdays is defined by the day of the week count buffer register RTCAnWEEK.

- RTCAnALH and RTCAnALM specify the hour and minute in BCD.

Examples

The following tables show some exemplary settings of the alarm control registers for both 12-hour and 24-hour format.

In this example, Sunday is RTCAnWEEK = 0, Monday is RTCAnWEEK = 1, Tuesday is RTCAnWEEK = 2, …, Saturday is RTCAnWEEK = 6:

Table 27.41 Alarm Setting in 12-Hour Format (RTCAnCTL0.RTCAnAMPM = 0)

Alarm Setting Time	RTCAnALW	RTCAnALH	RTCAnALM
Sunday 7:00 am	01_H	07_H	00_H
Sunday, Monday 12:15 pm	03_H	32_H	15_H
Monday, Wednesday, Friday 5:30 pm	$2A_H$	25_H	30_H
Daily, 10:45 pm	$7F_H$	30_H	45_H

Table 27.42 Alarm Setting in 24-Hour Format (RTCAnCTL0.RTCAnAMPM = 1)

Alarm Setting Time	RTCAnALW	RTCAnALH	RTCAnALM
Sunday 7:00	01_H	07_H	00_H
Sunday, Monday 12:15	03_H	12_H	15_H
Monday, Wednesday, Friday 17:30	$2A_H$	17_H	30_H
Daily, 22:45	$7F_H$	22_H	45_H

27.4.4 Clock Error Correction

Clock error correction compensates for deviations of the oscillator from the nominal clock rate. With clock error correction input clock rates from 32.76180 kHz to 32.77420 kHz are possible.

The clock error correction function is only available in 32.768-kHz operation mode. In this operation mode, a nominal clock rate of 32.768 kHz is expected and the sub-counters overflow value is fixed to $7FFF_H$.

The following figures illustrate the clock error when the input clock rate deviates from the nominal clock.

RTCATCKI = 32.768 kHz

Figure 27.2, RTCATCKI = 32.768 kHz, No Clock Error Correction Required shows the timing diagram if RTCATCKI matches the nominal clock rate of 32.768 kHz. No clock error correction is required.

Counting from 0 to 32767 (0 to $7FFF_H$) with a 32.768-kHz clock is exactly equal to one second.

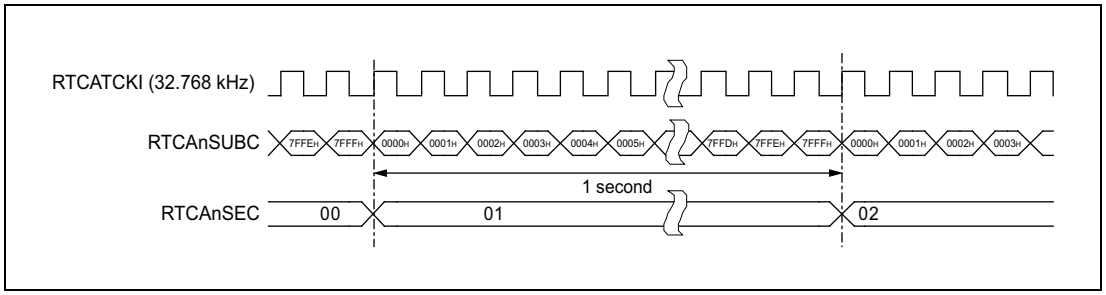


Figure 27.2 RTCATCKI = 32.768 kHz, No Clock Error Correction Required

RTCATCKI = 32.769 kHz

Figure 27.3, RTCATCKI = 32.769 kHz, No Clock Error Correction Enabled shows the timing diagram if RTCATCKI deviates from the nominal clock rate of 32.768 kHz. In this example, RTCATCKI is connected to a 32.769-kHz oscillator. Clock error correction is not enabled.

Counting from 0 to 32767 (0 to $7FFF_H$) with a 32.769-kHz clock is equal to approximately 0.99997 seconds (32768/32769). A “+ error” (faster than 32.768-kHz) occurs. In one month, RTCA deviates approximately –79 seconds from the real time.

Error = (32768/32769 - 1) × 60 (s) × 60 (min) × 24 (h) × 30 (d)

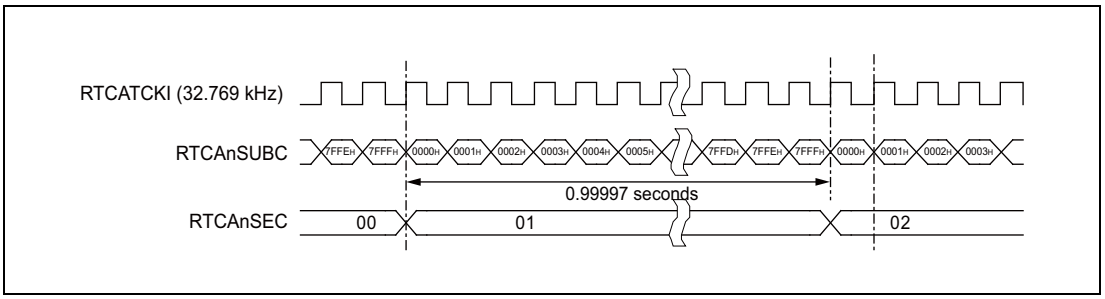


Figure 27.3 RTCATCKI = 32.769 kHz, No Clock Error Correction Enabled

Clock error correction is performed by stretching/reducing the 1-second period of the sub-counter at regular intervals. The sub-counter’s upper limit of $7FFF_H$ is increased or decreased by setting the following parameters in register RTCAnSUBU:

- A correction value greater than one
- An operator (add/subtract)
- An interval (20 or 60 seconds)

The corrected overflow value becomes effective every 20 or 60 seconds, so that on the average RTCAnSECC is triggered exactly every second.

27.4.4.1 Setting the Correction Value and the Operator

The correction value and operator are specified by the RTCAnF6, RTCAnF[5:0] bits of the RTCAnSUBU register:

- RTCAnF6 specifies whether the overflow value is incremented or decremented.
- RTCAnF[5:0] specifies the correction value.

The correction values are calculated as follows:

Table 27.43 Correction Value Settings

RTCAnF6	Increment/Decrement	Correction Value
0	Increment	(Value of RTCAnF[5:0] – 1) × 2
1	Decrement	(Inverted value of RTCAnF[5:0] + 1) × 2

Some examples are given in the following table:

Table 27.44 Correction Value Examples

RTCAnF6	RTCAnF[5:0]	Correction Value	Count Limit of RTCAnSUBC
0	15_H	$(15_H - 1) \times 2 = 40$	32768 + 40 = 32808
1	15_H	$(\overline{15}_H + 1) \times 2$ $= (2A_H + 1) \times 2$ $= 86$	32768 – 86 = 32682

27.4.4.2 Impact of the Repetition Interval

The correction value set by RTCAnF6, RTCAnF[5:0] does not change the count limit of RTCAnSUBC every second. The repetition interval at which the correction value becomes effective is specified by bit RTCAnDEV.

This bit also influences the size of the correctable frequency range and the correction accuracy.

The following table summarizes the RTCAnDEV settings.

Table 27.45 Setting of Bit RTCAnSUBU.RTCAnDEV

RTCAnDEV	Count Limit of RTCAnSUBC is Changed	Frequency Range that can be Corrected	Correction Accuracy
0	Every 20 seconds when RTCAnSECC = 00, 20, or 40	32.76180000 to 32.77420000 kHz	
1	Every 60 seconds when RTCAnSECC = 00	32.76593333 to 32.77006667 kHz	Three times higher than for RTCAnDEV = 0

27.4.4.3 Sample Settings

The frequencies that can be corrected, as well as the setting values of bits RTCAnDEV, RTCAnF6, and RTCAnF[5:0], are listed in the following table.

Table 27.46 Correctable Frequency Range when RTCAnDEV = 0

Input Clock Frequency	RTCAnF6	RTCAnF[5:0]	Correction Value of RTCAnSUBC
—	0	000000	No correction
—	0	000001	No correction
32.76810000 kHz	0	000010	Once every 20 s, RTCAnSUBC count value + 2
32.76820000 kHz	0	000011	Once every 20 s, RTCAnSUBC count value + 4
32.76830000 kHz	0	000100	Once every 20 s, RTCAnSUBC count value + 6
:	:	:	:
32.77400000 kHz	0	111101	Once every 20 s, RTCAnSUBC count value + 120
32.77410000 kHz	0	111110	Once every 20 s, RTCAnSUBC count value + 122
32.77420000 kHz (upper limit)	0	111111	Once every 20 s, RTCAnSUBC count value + 124
—	1	000000	No correction
—	1	000001	No correction
32.76180000 kHz (lower limit)	1	000010	Once every 20 s, RTCAnSUBC count value – 124
32.76190000 kHz	1	000011	Once every 20 s, RTCAnSUBC count value – 122
32.76200000 kHz	1	000100	Once every 20 s, RTCAnSUBC count value – 120
:	:	:	:
32.76770000 kHz	1	111101	Once every 20 s, RTCAnSUBC count value – 6
32.76780000 kHz	1	111110	Once every 20 s, RTCAnSUBC count value – 4
32.76790000 kHz	1	111111	Once every 20 s, RTCAnSUBC count value – 2

Table 27.47 Correctable Frequency Range when RTCAnDEV = 1

Input Clock Frequency	RTCAnF6	RTCAnF[5:0]	Correction Value of RTCAnSUBC
—	0	000000	No correction
—	0	000001	No correction
32.76803333 kHz	0	000010	Once every 60 s, RTCAnSUBC count value + 2
32.76806667 kHz	0	000011	Once every 60 s, RTCAnSUBC count value + 4
32.76810000 kHz	0	000100	Once every 60 s, RTCAnSUBC count value + 6
:	:	:	:
32.77000000 kHz	0	111101	Once every 60 s, RTCAnSUBC count value + 120
32.77003333 kHz	0	111110	Once every 60 s, RTCAnSUBC count value + 122
32.77006667 kHz (upper limit)	0	111111	Once every 60 s, RTCAnSUBC count value + 124
—	1	000000	No correction
—	1	000001	No correction
32.76593333 kHz (lower limit)	1	000010	Once every 60 s, RTCAnSUBC count value – 124
32.76596667 kHz	1	000011	Once every 60 s, RTCAnSUBC count value – 122
32.76600000 kHz	1	000100	Once every 60 s, RTCAnSUBC count value – 120
:	:	:	:
32.76790000 kHz	1	111101	Once every 60 s, RTCAnSUBC count value – 6
32.76793333 kHz	1	111110	Once every 60 s, RTCAnSUBC count value – 4
32.76796667 kHz	1	111111	Once every 60 s, RTCAnSUBC count value – 2

27.5 Procedures for Setup, Writing and Reading

The following subsections provide flow charts that illustrate the procedures for RTCA setup and for reading and writing the RTCA clock counters.

27.5.1 Initial Setting of the RTCA

The RTCA must be stopped before setting the initial setting value of each counter.

27.5.1.1 RTCA Stop Procedure

Stop the RTCA according to the following flow.

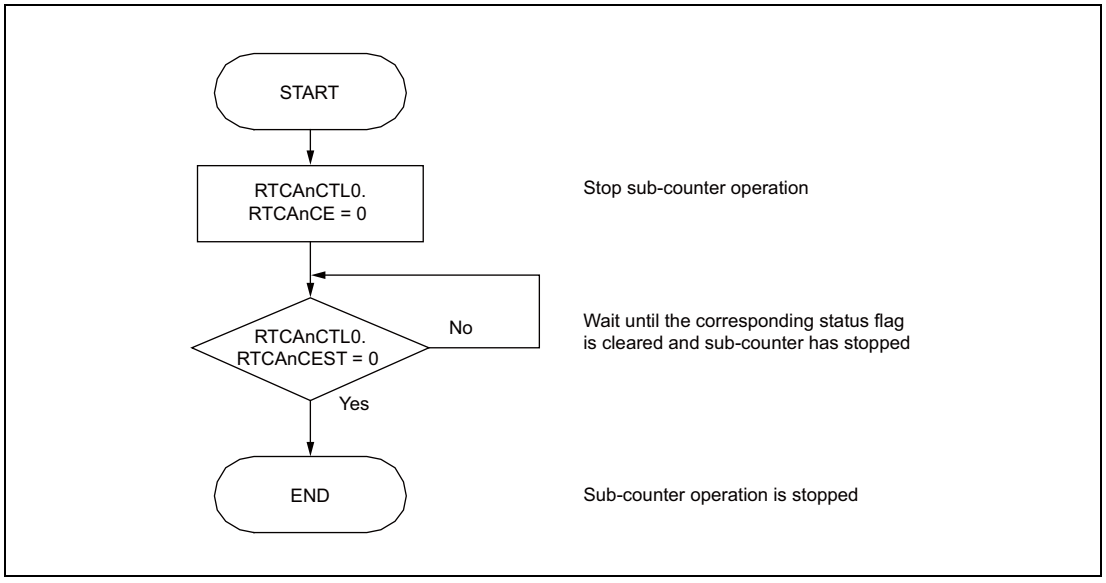


Figure 27.4 RTCA Stop Procedure

27.5.1.2 RTCA Initialization Procedure

Perform the initial setting of the RTCA according to the following flow:

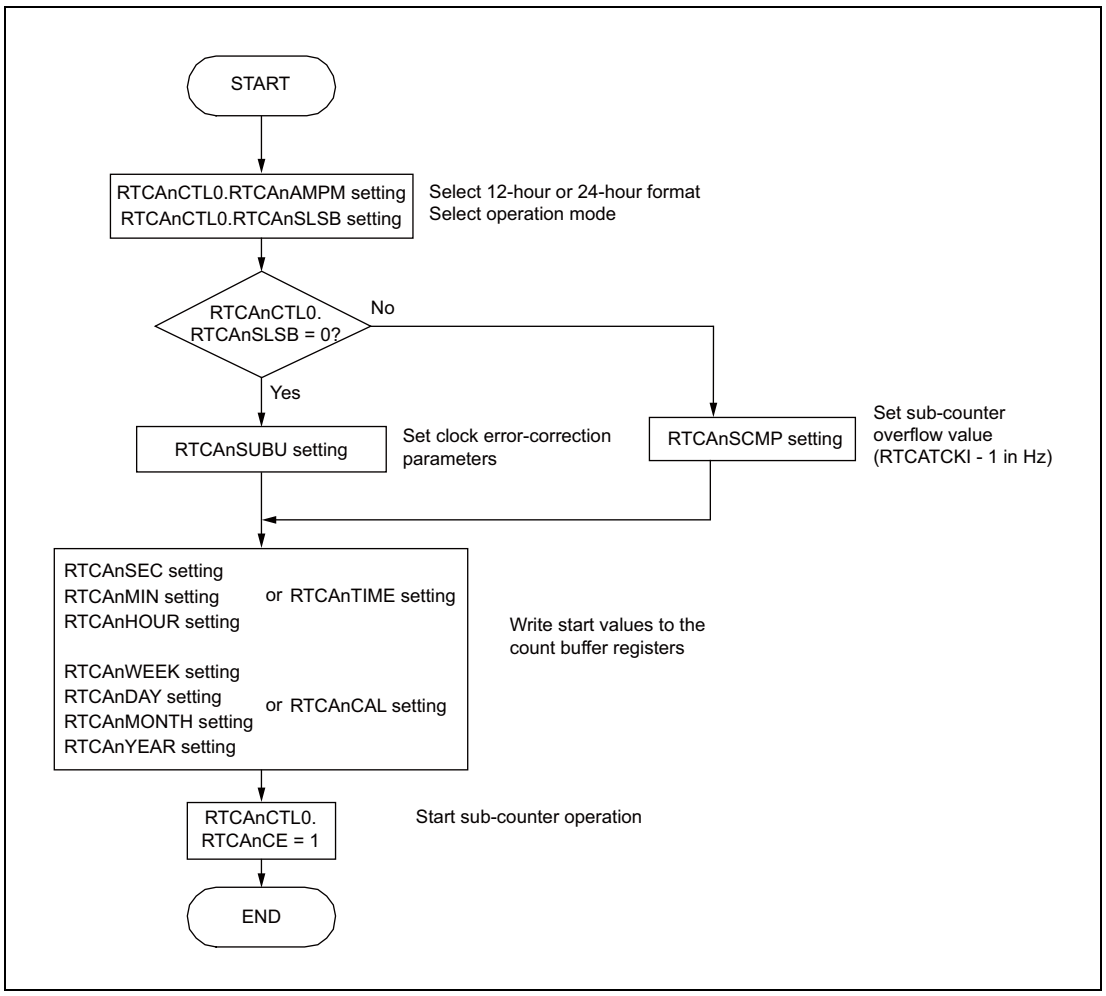


Figure 27.5 RTCA Initial Setup Procedure

CAUTION

The internal clock counter is synchronized with RTCATCKI.

In addition, two RTCATCKI periods are required before the clock counter starting behind END of the above flow.

Therefore, PCLK must be continuously supplied until the completion of the initial setting.

Check that RTCAnCTL0.RTCAnCEST = 1, when the supply of PCLK is stopped after setting the initial setting value of RTCA.

27.5.2 Updating Clock Counters

The clock counters RTCAnSECC to RTCAnYEARC can be stopped and updated while the sub-counter is running.

To update the clock counter when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), follow the flowchart shown below.

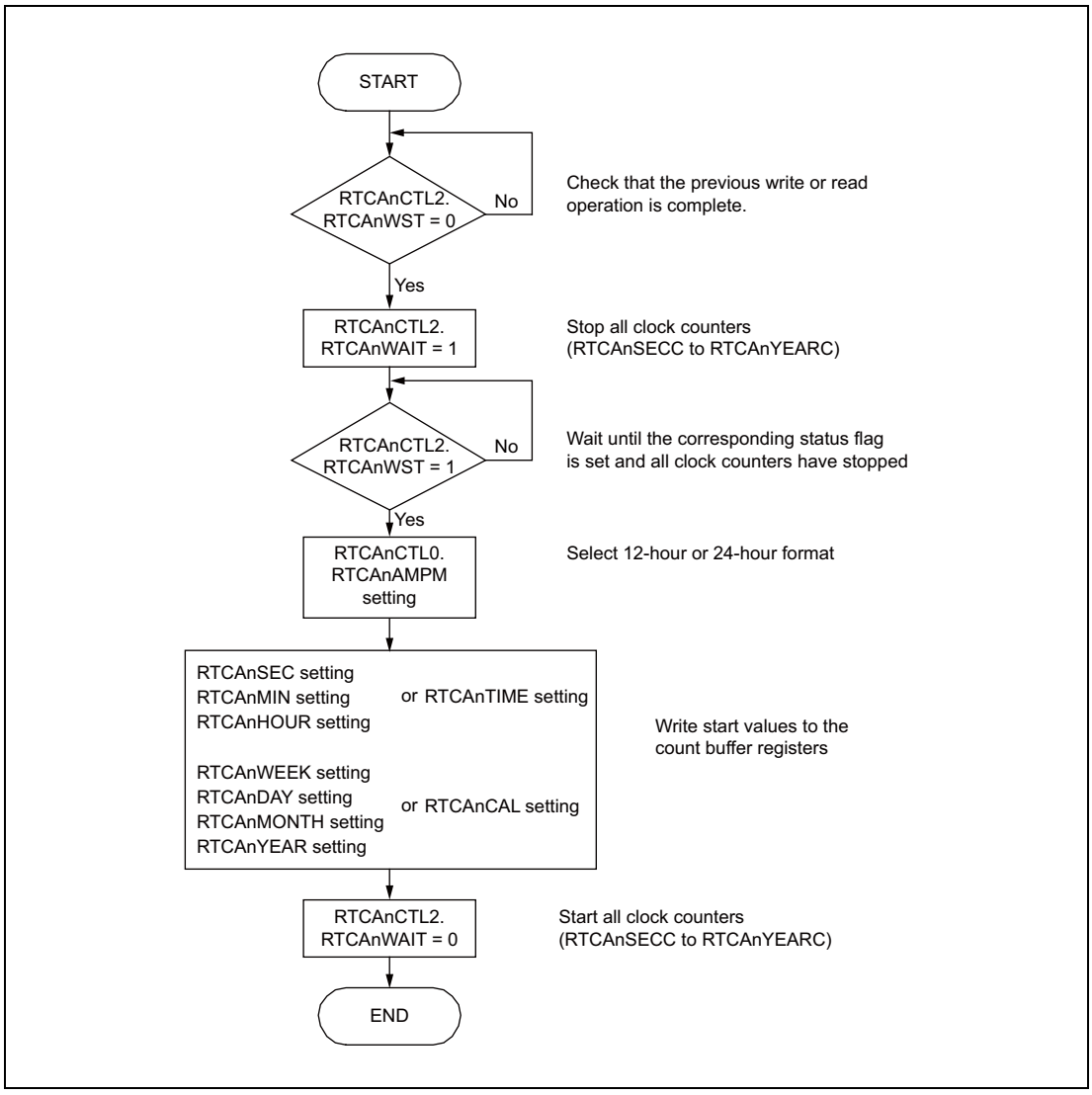


Figure 27.6 Updating Clock Counter Values

CAUTIONS

1. **The internal clock counter is synchronized with RTCATCKI.**
   In addition, two RTCATCKI periods are required before the clock counter updating behind END of the above flow.
   Therefore, PCLK must be continuously supplied until the completion of the clock counter updating.
   Check that RTCAnCTL2.RTCAnWST = 0 before stopping the supply of PCLK after the completion of clock counter updating.
2. **The update procedure must be completed within one second. Otherwise the Real-Time Clock will not count correctly any more:**

3. **Only one sub-counter overflow can be held internally and increment the seconds counter after restarting the clock counters if the value is held.**
4. **If the sub-counter overflows more than once during clock counter stop, the overflow count cannot be held internally. Thus the seconds counter is incremented by one instead of by two when it is restarted.**

27.5.3 Reading Clock Counters

There are two methods to read the clock counters while sub-counter operation is enabled:

- Reading count buffer registers
- Reading counter registers

The advantages and disadvantages of the two methods are summarized in the following table.

Table 27.48 Comparison of the Two Read Methods

	Advantage	Disadvantage
Reading count buffer registers	It is unnecessary to read clock counters several times because the clock counters are read synchronously.	A program wait state occurs between setting RTCAnCTL2.RTCAnWAIT = 1 and completion of data transfer.
Reading count registers	Program wait state does not occur.	If the sub-counter increments, the clock counters must be read several times because they are read asynchronously to RTCATCKI.

27.5.3.1 Procedure for Reading Count Buffer Registers

The following operations are necessary:

1. Stop all clock counters (RTCAnCTL2.RTCAnWAIT = 1). The value of the clock counters is transferred to the corresponding count buffer registers.
2. Read the count buffer registers.

A program wait state occurs between setting RTCAnCTL2.RTCAnWAIT = 1 and completion of data transfer.

The maximum delay is three PCLK periods plus two RTCATCKI periods. For example, if the RTCA operates with PCLK = 60 MHz and RTCATCKI = 32.768 kHz, the delay is about 61 µs.

To read the count buffer register when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1), follow the flowchart shown below.

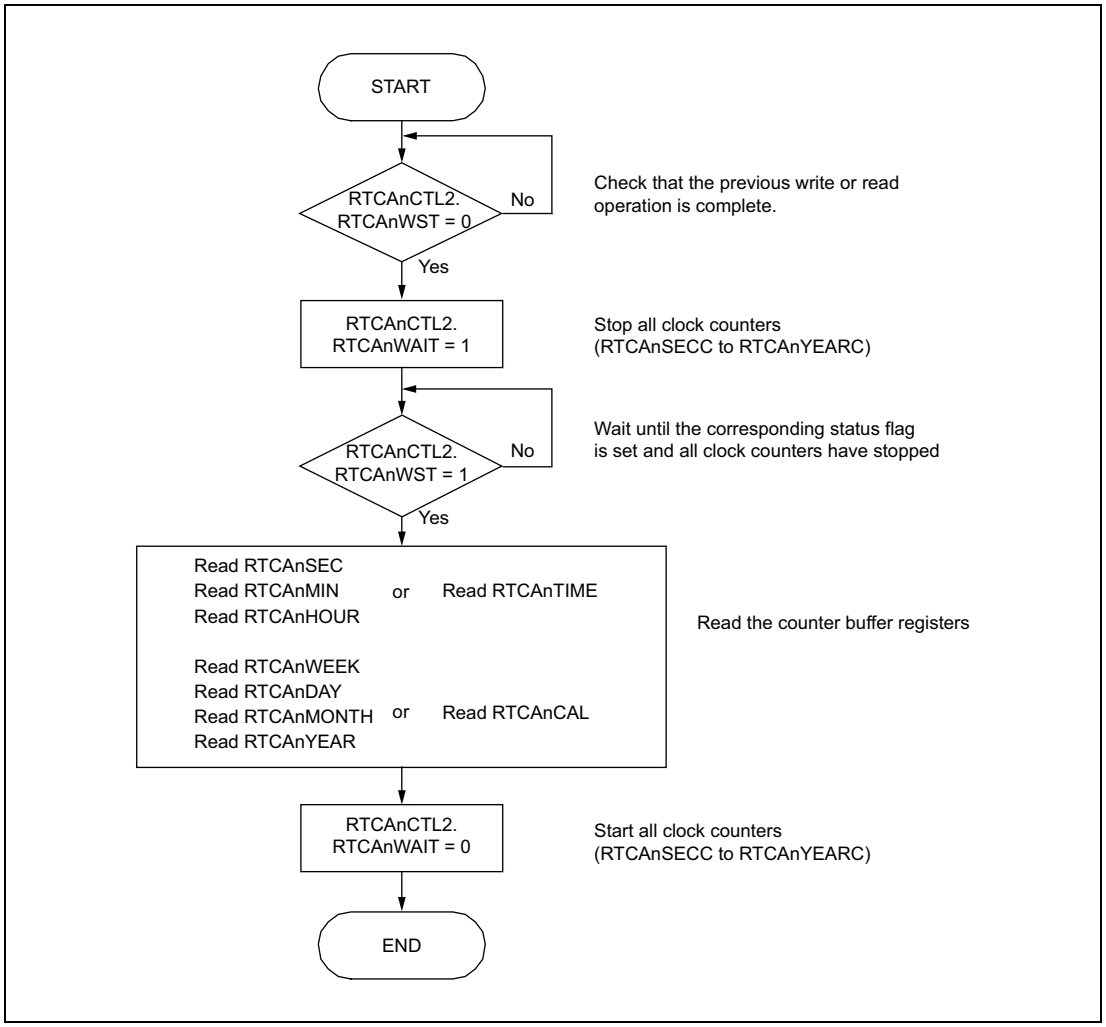


Figure 27.7 Reading Clock Count Buffer Registers

CAUTIONS

1. **The internal clock counter is synchronized with RTCATCKI.**
   In addition, two RTCATCKI periods are required before resuming counter behind END of the above flow.
   Therefore, PCLK must be continuously supplied until the counter resuming.
   Check that RTCAnCTL0.RTCAnCEST = 1 first to stop the supply of PCLK after count buffer register reading.
2. **The reading procedure must be completed within one second. Otherwise the Real-Time Clock will not count correctly any more.**
3. **Only one sub-counter overflow can be held internally. If there is a value held internally when the clock counter restarts, the seconds counter will be incremented by 1.**
4. **If the sub-counter overflows more than once during clock counter stop, the overflow count cannot be held internally. Thus the seconds counter is incremented by one instead of by two when it is restarted.**

27.5.3.2 Procedure for Reading Counter Registers Directly

To ensure that the sub-counter did not overflow while reading the counters, the seconds counter RTCAnSECC must be read twice in the beginning and at the end of the procedure. The first read value is compared with the second read value.

- First read value = second read value:
  No overflow of sub-counter occurred during counter read operation.
- First read value ≠ second read value:
  Overflow of the sub-counter occurred during counter read operation. The counters must be read again to get the current counter values.

To read the counter register directly when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), follow the flowchart shown below.

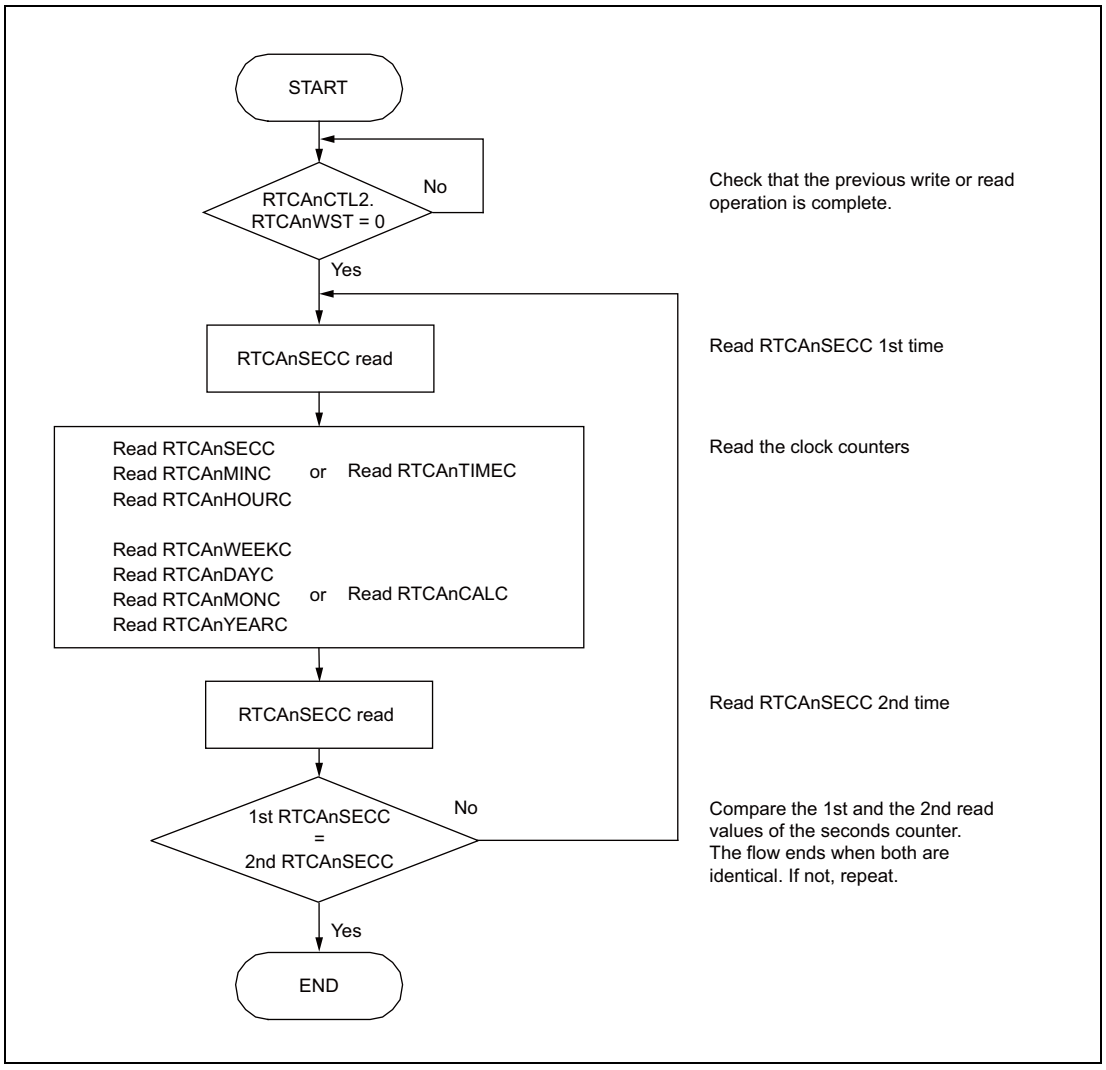


Figure 27.8 Reading Clock Counter Registers

NOTE

The procedure must be completed within one second.

27.5.4 Reading RTCAnSRBU

RTCAnSRBU is the read buffer register for the sub-counter.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), read RTCAnSRBU according to the following flow.

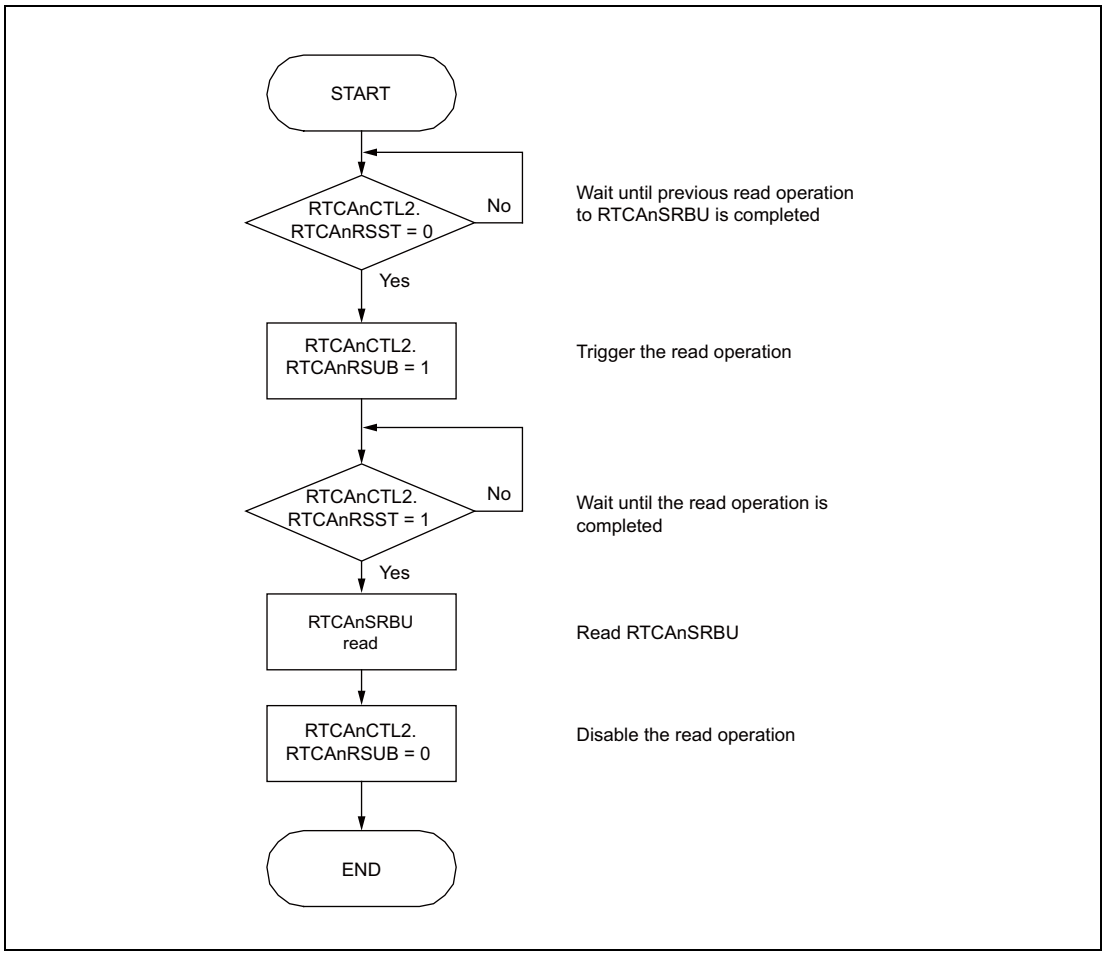


Figure 27.9 Reading the RTCAnSRBU Register

27.5.5 Writing to RTCAnSUBU

RTCAnSUBU is the clock error correction register for the sub-counter.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), write to RTCAnSUBU according to the flow described below.

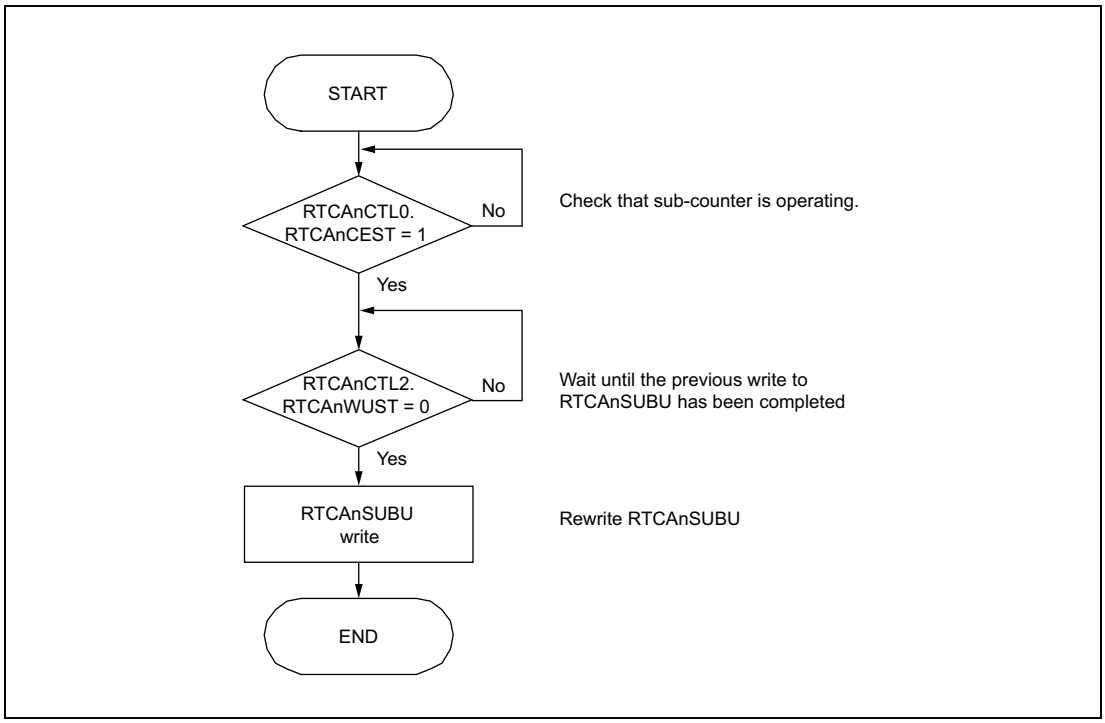


Figure 27.10 Writing to the RTCAnSUBU Register

NOTE

While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), the status flag RTCAnCTL2.RTCAnWUST is set when RTCAnSUBU is written to. It is cleared when the write operation to RTCAnSUBU is completed. This is synchronous with the next RTCAnSUBC overflow.
RTCAnCTL2.RTCAnWUST can be set for up to one second. Be careful when performing polling (checking if RTCAnCTL2.RTCAnWUST = 1 at the beginning of this flow).

27.5.6 Writing to RTCAnSCMP

RTCAnSCMP is the sub-counter compare register.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), write to RTCAnSCMP according to the flow described below.

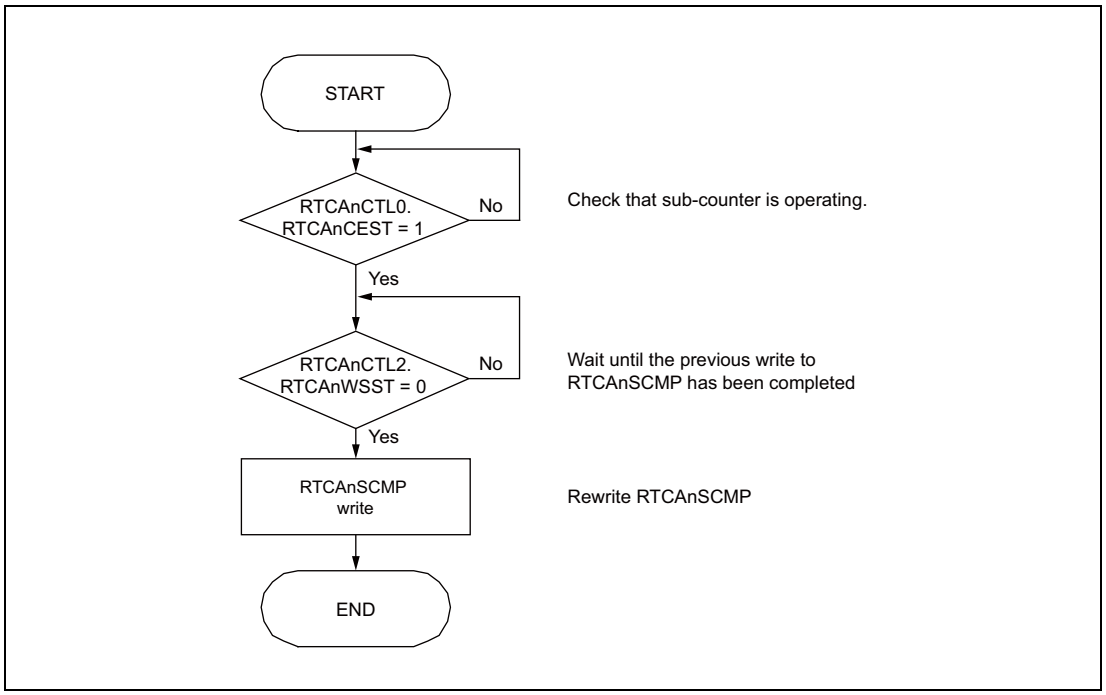


Figure 27.11 Writing the RTCAnSCMP Register

NOTE

While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), the status flag RTCAnCTL2.RTCAnWSST is set when RTCAnSCMP is written to. It is cleared when the write operation to RTCAnSCMP is completed. This is synchronous with the next RTCAnSUBC overflow.

RTCAnCTL2.RTCAnWSST can be set for up to one second. Be careful when performing polling (checking if RTCAnCTL2.RTCAnWSST = 1 at the beginning of this flow).

27.6 Timing Diagrams

27.6.1 Timing of Counter Start

The following diagram illustrates the counter start after setting the time in the buffer registers.

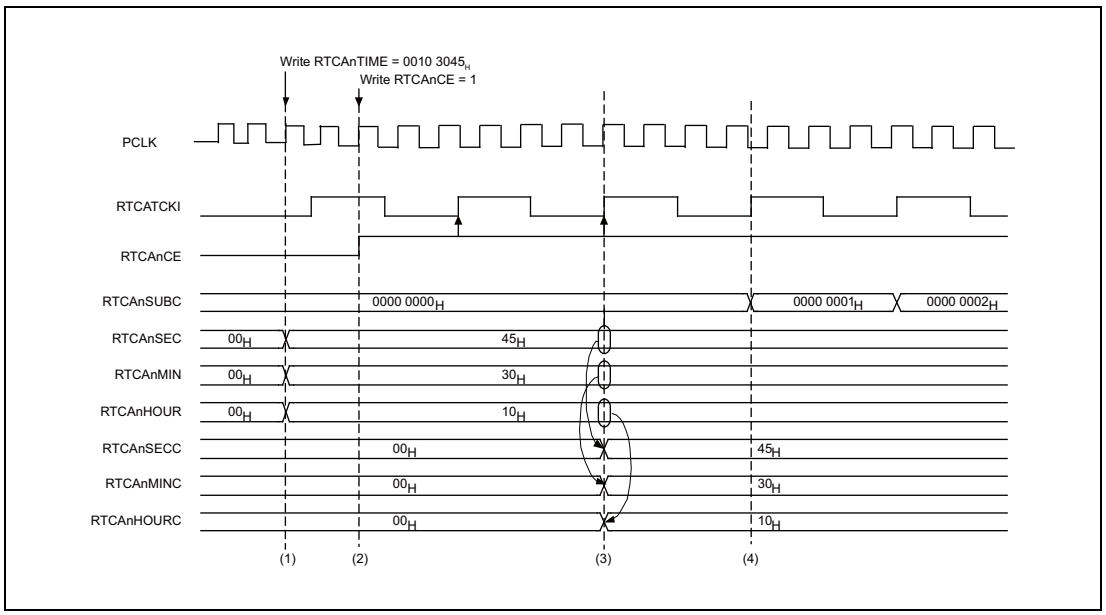


Figure 27.12 Counter Start Timing

The timing diagram above shows the following:

(1) The initial setting value of the time count buffer is set to 10:30:45 by setting RTCAnTIME = 0010 3045_H.
Count buffer registers RTCAnSEC, RTCAnMIN, and RTCAnHOUR are also automatically written.

(2) Sub-counter operation is started by setting RTCAnCTL0.RTCAnCE = 1.

(3) When the second rising edge of RTCATCKI occurs, the buffer register values are loaded to the corresponding count registers.

(4) When the next rising edge of RTCATCKI occurs, count up of the sub-counter starts.

27.6.2 Timing of Clock Counter Update while Counter Is Enabled

The following diagram illustrates the counter restart after setting the time in the buffer registers.

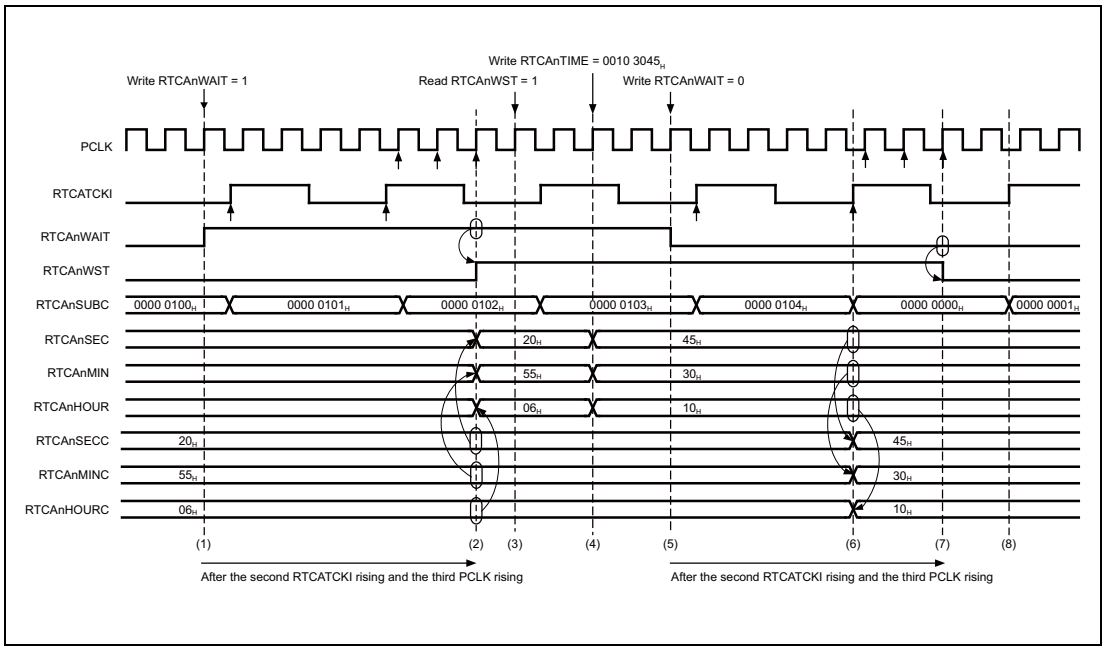


Figure 27.13 Clock Counter Update Timing

The timing diagram above shows the following:

(1) Trigger the clock counters stop (RTCAnCTL2.RTCAnWAIT = 1).

(2) RTCAnCTL2.RTCAnWST is set to 1 after the second rising edge of RTCATCKI and the third rising edge of PCLK, and the counter clock stops. The sub-counter continues counting.

(3) RTCAnCTL2.RTCAnWST = 1 can be readable.

(4) The initial setting value of the time count buffer is set to 10:30:45 by setting RTCAnTIME to 0010 3045_H.
Count buffer registers RTCAnSEC, RTCAnMIN, and RTCAnHOUR are also automatically written.

(5) Trigger the clock counters restart (RTCAnCTL2.RTCAnWAIT = 0).

(6) When the second rising edge of RTCATCKI occurs, the values of the buffer registers are loaded to the corresponding count registers. Write operation to RTCAnSECC is performed and RTCAnSUBC is cleared.

(7) When the third rising edge of PCLK occurs, RTCAnCTL2.RTCAnWST is set to 0.

(8) Clock counter operation is resumed.

27.6.3 Timing of Sub-Counter Read Buffer Reading while Counter is Enabled

The following diagram illustrates the timing when reading the sub-counter read buffer RTCAnSRBU.

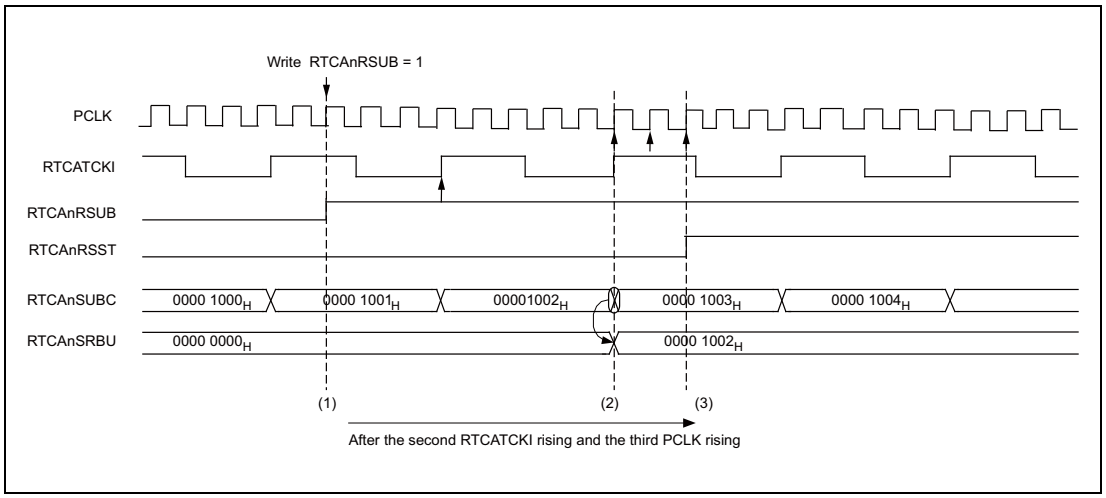


Figure 27.14 Timing when Reading the Sub-Counter Read Buffer Register Value

The timing diagram above shows the following:

(1) Setting RTCAnRSUB = 1 triggers loading of the sub-counter value to RTCAnSRBU.

(2) When the second rising edge of RTCATCKI occurs, the value of RTCAnSUBC is loaded to RTCAnSRBU.

(3) When the third rising edge of PCLK occurs, RTCAnCTL2.RTCAnRSST is set to 1 and RTCAnSRBU can be read.

Section 28 Encoder Timer (ENCA)

This section contains a generic description of the Encoder Timer (ENCA).

The first part in this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the ENCA.

28.1 Features of RH850/F1M ENCA

28.1.1 Number of Units and Channels

This microcontroller has the following number of ENCA units.

Each ENCA unit has one channel ENCA. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 28.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	1		
Name	ENCAn (n = 0)		

Table 28.2 Index

Index	Description
n	Throughout this section, the individual ENCA units are identified by the index “n” (n = 0); for example, ENCAnCTL is the ENCAn control register.

28.1.2 Register Base Address

ENCAn base address is listed in the following table.

ENCAn register addresses are given as offsets from the base address.

Table 28.3 Register Base Address

Base Address Name	Base Address
<ENCA0_base>	FFE8 0000_H

28.1.3 Clock Supply

The ENCAn clock supply is shown in the following table.

Table 28.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
ENCA0	PCLK	CKSCLK_IPERI1
	Register access clock	CKSCLK_IPERI1

28.1.4 Interrupt Requests

ENCAn interrupt requests are listed in the following table.

Table 28.5 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
ENCA0			
ENCATIOV	Overflow interrupt	85	—
ENCATIUD	Underflow interrupt	86	—
ENCATINT0	Capture/compare match interrupt 0	87	—
ENCATINT1	Capture/compare match interrupt 1	88	—
ENCATIEC	Encoder clear interrupt	89	—

28.1.5 Reset Sources

ENCAn reset sources are listed in the following table. ENCAn is initialized by these reset sources.

Table 28.6 Reset Sources

Unit Name	Reset Source
ENCA0	All reset sources (ISORES)

28.1.6 External Input/Output Signals

External input/output signals of ENCAn are listed below.

Table 28.7 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal Name
ENCA0		
ENCATTIN0	ENCAn capture trigger input 0*1	ENCA0TIN0
ENCATTIN1	ENCAn capture trigger input 1*1	ENCA0TIN1
ENCAnE0	ENCAn encoder input 0*1	ENCA0E0
ENCAnE1	ENCAn encoder input 1*1	ENCA0E1
ENCAnEC	ENCAn encoder clear input*1	ENCA0EC

Note 1. When channel input pins are to be used, noise filters must be set for the corresponding port pin functions. For details, see **Section 2.12, Noise Filter & Edge/Level Detector**.

28.1.7 Internal Input/Output Signals

Input/output signals to be connected between ENCA and PIC are listed below.

Table 28.8 Internal Input/Output Signals

Unit Signal Name	Description	Connected to
ENCATSST	Simultaneous start trigger	PIC
ENCATTIN1	ENCAn capture trigger input 1	PIC

28.2 Overview

28.2.1 Functional Overview

- Generation of the counter control signal from the encoder input signal, and count operation in synchronization with PCLK
- Capture function for capturing the counter value with an external trigger signal
- Compare function for compare match judgment with the counter value
- Two capture/compare registers that can be set separately for capture operation and for compare operation
- Interrupt mask function for masking the interrupt request signal output as a result of compare match judgment during compare operation
- Function for loading the value of the capture/compare register to the counter upon underflow occurrence
- Encoder input signal can be used as the timer counter clear condition
- Edge or level can be selected for determining the presence of the encoder input signal that is used as the timer counter clear condition
- Detection of counter overflow and underflow and output of error flags and error occurrence interrupts
- Five interrupts: two capture/compare interrupts, one counter clear interrupt, one overflow interrupt, and one underflow interrupt

28.2.2 Block Diagram

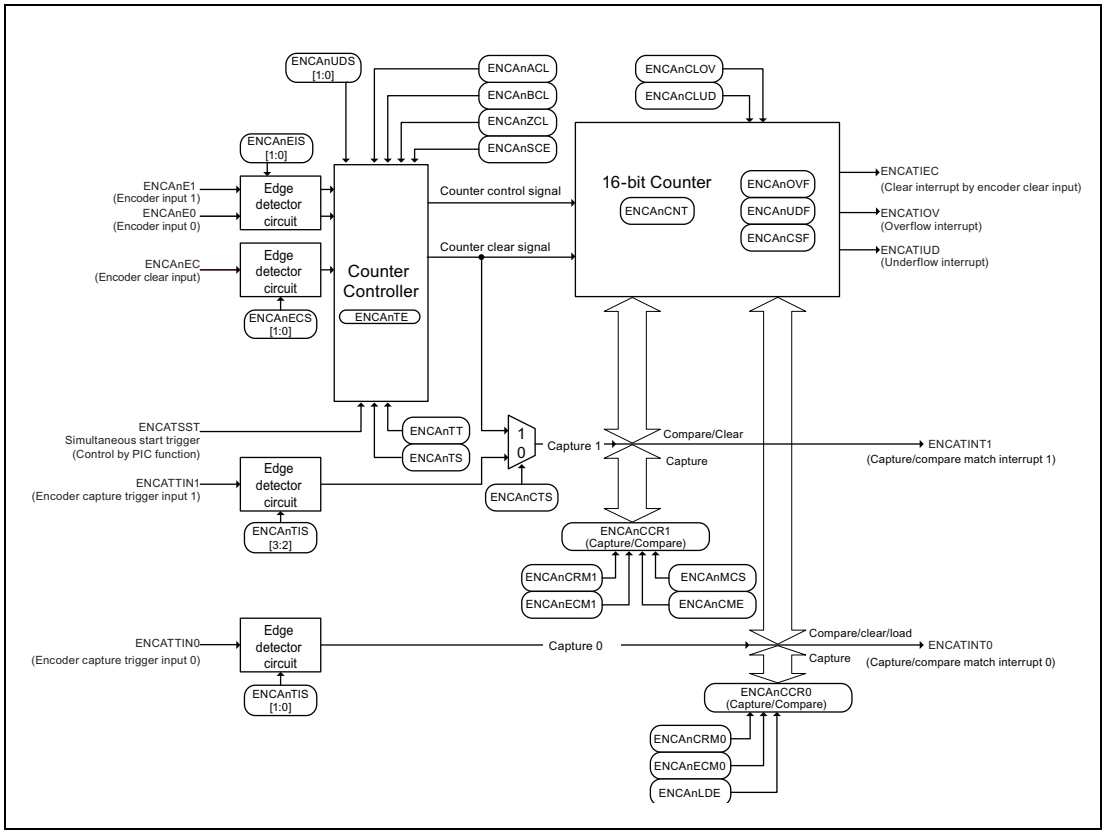


Figure 28.1 ENCA Block Diagram

28.3 Registers

28.3.1 List of Registers

ENCA registers are listed in the following table.

<ENCAn_base> is defined in **Section 28.1.2, Register Base Address**.

Table 28.9 Registers

Module	Register	Symbol	Address
ENCAn	ENCAn capture/compare register 0	ENCAnCCR0	<ENCAn_base>
ENCAn	ENCAn capture/compare register 1	ENCAnCCR1	<ENCAn_base> + 04_H
ENCAn	ENCAn counter register	ENCAnCNT	<ENCAn_base> + 08_H
ENCAn	ENCAn status flag register	ENCAnFLG	<ENCAn_base> + $0C_H$
ENCAn	ENCAn status flag clear register	ENCAnFGC	<ENCAn_base> + 10_H
ENCAn	ENCAn timer enable status register	ENCAnTE	<ENCAn_base> + 14_H
ENCAn	ENCAn timer start trigger register	ENCAnTS	<ENCAn_base> + 18_H
ENCAn	ENCAn timer stop trigger register	ENCAnTT	<ENCAn_base> + $1C_H$
ENCAn	ENCAn I/O control register 0	ENCAnIOC0	<ENCAn_base> + 20_H
ENCAn	ENCAn control register	ENCAnCTL	<ENCAn_base> + 40_H
ENCAn	ENCAn I/O control register 1	ENCAnIOC1	<ENCAn_base> + 44_H
ENCAn	ENCAn emulation register	ENCAnEMU	<ENCAn_base> + 48_H

28.3.2 ENCAnCTL — ENCAn Control Register

This register is used to configure various operation settings of the Encoder Timer.

Access: This register can be read or written in 16-bit units.
Writing to this register during operation is prohibited.

Address: <ENCAn_base> + 40_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAn CME	ENCAn MCS	—	—	—	—	ENCAn CRM1	ENCAn CRM0	ENCAn CTS	—	—	ENCAn LDE	ENCAn ECM1	ENCAn ECM0	ENCAnUDS [1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 28.10 ENCAnCTL Register Contents (1/2)

Bit Position	Bit Name	Function
15	ENCAnCME	Encoder Clear Mask Enable This bit is used to enable/disable masking of compare-match interrupt detection when the compare function is used. 0: Disables the compare-match interrupt (ENCATINT1) mask function for the ENCAnCCR1 register 1: Enables the compare-match interrupt (ENCATINT1) mask function for the ENCAnCCR1 register. This bit is valid only when ENCAnCRM1 = 0. When this bit is set to “1”, setting ENCAnECM1 to “1” is prohibited.
14	ENCAnMCS	Encoder Mask Clear Select This bit is used to select the trigger for cancelling masking of compare-match interrupt detection ENCATINT1 when the compare function is used. This bit is valid only when ENCAnCRM1 = 0. 0: Masking of compare-match interrupt detection is canceled when the ENCAnCCR1 register is written. 1: Masking of compare match interrupt detection is canceled when one of the following three operations is performed. – Timer counter clear operation accompanying encoder clear input – Timer counter clear operation upon compare-match between ENCAnCNT and ENCAnCCR0 when ENCAnECM0 = 1 – Loading from ENCAnCCR0 to the timer counter upon underflow detection when ENCAnLDE = 1
13 to 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	ENCAnCRM1	ENCAnCCR1 Register Mode 0: ENCAnCCR1 used as compare register. 1: ENCAnCCR1 used as capture register.
8	ENCAnCRM0	ENCAnCCR0 Register Mode 0: ENCAnCCR0 used as compare register. 1: ENCAnCCR0 used as capture register.
7	ENCAnCTS	ENCAnCCR1 Capture Trigger Select This is a trigger selection bit for the capture operation to the ENCAnCCR1 register. This bit is valid only when ENCAnCRM1 = 1. 0: Uses ENCATTIN1 of capture trigger 1 signal as the capture trigger for the ENCAnCCR1 register. 1: The counter clear signal selected with ENCAnSCE is used as the capture trigger for the ENCAnCCR1 register.
6, 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 28.10 ENCAnCTL Register Contents (2/2)

Bit Position	Bit Name	Function
4	ENCAnLDE	ENCAn Counter Load Enable This bit is used to enable/disable setting value loading to the counter upon underflow occurrence. This bit is valid only when ENCAnCRM0 = 0. When ENCAnCRM0 = 1, loading of the ENCAnCCR0 register setting value to the counter upon occurrence of an underflow is not performed, regardless of the value of this bit. 0: Disables loading of ENCAnCCR0 register setting value to counter upon occurrence of a counter underflow. 1: Enables loading of ENCAnCCR0 register setting value to counter upon occurrence of a counter underflow.
3	ENCAnECM1	Encoder Clear Mode 1 This bit is used to set the counter clear operation upon match between the counter value and ENCAnCCR1 setting value. This bit is valid only when ENCAnCRM1 = 0. 0: Does not clear the counter to 0000_H upon match of timer counter value and ENCAnCCR1 setting value. 1: Clears the counter to 0000_H upon match of timer counter value and ENCAnCCR1 setting value if the next count is a down-count.
2	ENCAnECM0	Encoder Clear Mode 0 This bit is used to set the counter clear operation upon match between the counter value and ENCAnCCR0 setting value. This bit is valid only when ENCAnCRM0 = 0. 0: Does not clear the counter to 0000_H upon match of timer counter value and ENCAnCCR0 setting value. 1: Clears the counter to 0000_H upon match of timer counter value and ENCAnCCR0 setting value if the next count is a up-count.
1, 0	ENCAnUDS[1:0]	Up/down Count Selection 1 and 0 These are the counter up/down control bits using ENCAnE0 and ENCAnE1. 00: Upon detection of valid edge of ENCAnE0, - down-count when ENCAnE1 = H, - up-count when ENCAnE1 = L 01: Upon detection of valid edge of ENCAnE0, up-count, Upon detection of valid edge of ENCAnE1, down-count 10: At rising edge of ENCAnE0, down-count At falling edge of ENCAnE0, up-count However, count operation is performed only when ENCAnE1 = L. 11: Detection of both edges of ENCAnE0, ENCAnE1. The count operation is determined based on the combination of the detected edge and level.

28.3.3 ENCAnIOC0 — ENCAn I/O Control Register 0

This register is used to select the input edge of capture triggers 0 and 1 (ENCATTIN0, ENCATTIN1).

Access: This register can be read or written in 8-bit units.

Address: <ENCAn_base> + 20_H

Value after reset: 00_H

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

Table 28.11 ENCAnIOC0 Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	ENCAnTIS[3:2]	Input Edge Selection for Capture Trigger 1 These bits are valid only when ENCAnCTL.ENCAnCRM1 = 1 and ENCAnCTL.ENCAnCTS = 0. All other settings of ENCAnCRM1 and ENCAnCTS are invalid. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection
1, 0	ENCAnTIS[1:0]	Input Edge Selection for Capture Trigger 0 These bits are valid only when ENCAnCTL.ENCAnCRM0 = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

28.3.4 ENCAnIOC1 — ENCAn I/O Control Register 1

This register is used to perform the clear condition setting and edge selection for input from the encoder.

Access: This register can be read or written in 8-bit units.
Writing to this register during operation is prohibited.

Address: <ENCAn_base> + 44_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ENCAnSCE	ENCAnZCL	ENCAnBCL	ENCAnACL	ENCAnECS[1:0]		ENCAnEIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.12 ENCAnIOC1 Register Contents (1/2)

Bit Position	Bit Name	Function
7	ENCAnSCE	Encoder Special-Clear Enable This is an encoder special clear enable bit. When setting this bit to 1, set ENCAnUDS1 and ENCAnUDS0 to 10_B or 11_B. The operation is not guaranteed if this bit is set to 1 with ENCAnUDS1 and ENCAnUDS0 set to 00_B or 01_B. 0: Clears the counter upon detection of ENCAnEC valid edge (set with ENCAnECS1 and ENCAnECS0). 1: Clears the counter upon detection of input level condition of ENCAnE0, ENCAnE1 and ENCAnEC (set with ENCAnZCL bit, ENCAnBCL bit, and ENCAnACL bit).
6	ENCAnZCL	Input-Z Clear Condition Selection This bit is used to set the condition for clearing the encoder clear input (ENCAnEC) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
5	ENCAnBCL	Input-B Clear Condition Selection This bit is used to set the condition for clearing the encoder input 1 (ENCAnE1) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
4	ENCAnACL	Input-A Clear Condition Selection This bit is used to set the condition for clearing the encoder input 0 (ENCAnE0) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
3, 2	ENCAnECS[1:0]	Encoder Clear Input Edge Selection 1 and 0 These are the encoder clear input edge selection bits. These bits are valid only when ENCAnSCE = 0; they are invalid when ENCAnSCE = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

Table 28.12 ENCAnIOC1 Register Contents (2/2)

Bit Position	Bit Name	Function
1, 0	ENCAnEIS[1:0]	Encoder Edge Input Selection 1 and 0 These are the encoder input edge selection bits. These bits are valid when ENCAnUDS1 and ENCAnUDS0 = 00_B or 01_B, and are invalid when ENCAnUDS1 and ENCAnUDS0 = 10_B or 11_B. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

28.3.5 ENCAnFLG — ENCAn Status Flag Register

This register holds the status flags of the timer counter of ENCAn.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <ENCAn_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	ENCAnCSF	ENCAnUDF	ENCAnOVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 28.13 ENCAnFLG Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2	ENCAnCSF	Counter Status Flag This bit reflects the current timer counter operation. 0: Timer counter in up-count status 1: Timer counter in down-count status
1	ENCAnUDF	Underflow Flag This bit reflects the occurrence of an underflow during the timer counter operation. This bit is cleared at the start of count operation. 0: This flag is cleared to 0 upon any of the following events: – “1” is written to ENCAnFGC.ENCAnCLUD – The flag is cleared to 0 by setting ENCAnTS bit to “1” when ENCAnTE = 0 or by setting the simultaneous start trigger input (ENCATSST signal) to “High”. 1: This flag is set to “1” upon occurrence of an underflow during the encoder timer count operation.
0	ENCAnOVF	Overflow Flag This bit reflects the occurrence of an overflow during the timer counter operation. This bit is cleared at the start of count operation. 0: This flag is cleared to 0 upon any of the following events: – “1” is written to ENCAnFGC.ENCAnCLOV – The flag is cleared to 0 by setting ENCAnTS bit to “1” when ENCAnTE = 0 or by setting the simultaneous start trigger input (ENCATSST signal) to “High”. 1: This flag is set to “1” upon occurrence of an overflow during the encoder timer count operation.

28.3.6 ENCAnFGC — ENCAn Status Flag Clear Register

This register is used to clear the timer counter status flags of ENCAnFLG.

Access: This register is a write-only register that can be written in 8-bit units.
This register always returns 0 when read.

Address: <ENCAn_base> + 10_H

Value after reset: 00_H

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

Table 28.14 ENCAnFGC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	ENCAnCLUD	Underflow Flag Clear This bit clears the underflow flag. 0: Writing is ignored. 1: Clears ENCAnUDF of the ENCAnFLG register (clears underflow detection).
0	ENCAnCLOV	Overflow Flag Clear This bit clears the overflow flag. 0: Writing is ignored. 1: Clears ENCAnOVF of the ENCAnFLG register (clears overflow detection).

28.3.7 ENCAnCCR0 — ENCAn Capture/Compare Register 0

This register is a 16-bit capture/compare register 0.

Access: This register can be read or written in 16-bit units.
When this register functions as a capture register, only reading is possible. Write operation is ignored.
When this register functions as a compare register, reading and writing is possible.

Address: <ENCAn_base> + 00_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAnCCR0[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.15 ENCAnCCR0 Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAnCCR0 [15:0]	Capture/Compare Register 0 Upon occurrence of an underflow, the setting value of this register may be loaded to the counter according to the ENCAnCTL.ENCAnLDE setting. See the description of the ENCAnLDE bit in ENCA control register ENCAnCTL for details. • If ENCAnCTL.ENCAnCRM0 = 0: ENCAnCCR0 is a compare register. Set the value to be compared with the timer counter value. • If ENCAnCTL.ENCAnCRM0 = 1: ENCAnCCR0 is a capture register. The captured timer counter value is stored.

28.3.8 ENCAnCCR1 — ENCAn Capture/Compare Register 1

This register is a 16-bit capture/compare register 1.

Access: This register can be read or written in 16-bit units.
When this register functions as a capture register, only reading is possible. Write operation is ignored.
When this register functions as a compare register, reading and writing is possible.

Address: <ENCAn_base> + 04_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAnCCR1[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.16 ENCAnCCR1 Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAnCCR1 [15:0]	Capture/Compare Register 1 During capture operation, the capture trigger to this register differs according to the ENCAnCTL.ENCAnCTS setting. See the description of the ENCAnCTS bit in ENCA control register ENCAnCTL for details. • If ENCAnCTL.ENCAnCRM1 = 0: ENCAnCCR1 is a compare register. Set the value to be compared with the timer counter value. • If ENCAnCTL.ENCAnCRM1 = 1: ENCAnCCR1 is a capture register. The captured timer counter value is stored.

28.3.9 ENCAnCNT — ENCAn Counter Register

This register is the 16-bit timer counter register.

Access: This register can be read or written in 16-bit units.
This register can be written only when the operation is stopped.

Address: <ENCAn_base> + 08_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ENCAnCNT[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.17 ENCAnCNT Register Contents

Bit Position	Bit Name	Function
15 to 0	ENCAnCNT [15:0]	Counter Register • ENCAnTE.ENCAnTE status: 0 (initial setting): Count stop An arbitrary value can be set to timer counter. • ENCAnTE.ENCAnTE status: 0 → 1 (operation start): Count operation start Up/down count operation is started with the set arbitrary value. • ENCAnTE.ENCAnTE status: 1 (operating): Counting Up/down count operation is performed. • ENCAnTE.ENCAnTE status: 1 → 0 (stopped): Count stop The counter value immediately before the operation was stopped is held, and the count operation is stopped.

28.3.10 ENCAnTE — ENCAn Timer Enable Status Register

This register indicates the operating status of ENCAn.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <ENCAn_base> + 14_H

Value after reset: 00_H

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

Table 28.18 ENCAnTE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	ENCAnTE	Timer Status Enable This is a status bit that indicates the operation enabled/stopped status of ENCAn. This bit is cleared to 0 when “1” is written to ENCAnTT.ENCAnTT. This bit is set to “1” when “1” is written to ENCAnTS.ENCAnTS, or when the input signal of ENCATSST is set to High level. 0: Operation stopped status 1: Operation enabled status

28.3.11 ENCAnTS — ENCAn Timer Start Trigger Register

This register provides the trigger bit for setting the ENCAn to the operation enabled state.

Access: This register is a write-only register that can be written in 8-bit units.
It is always read as 00_H. This register can be written only when ENCAnTE.ENCAnTE is 0.

Address: <ENCAn_base> + 18_H

Value after reset: 00_H

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

Table 28.19 ENCAnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ENCAnTS	Timer Start Trigger This is the trigger bit that sets the ENCAn to the operation enabled state. 0: Writing is ignored. 1: The ENCAn is set to the operation enabled state by setting ENCAnTE.ENCAnTE = 1.

28.3.12 ENCAnTT — ENCAn Timer Stop Trigger Register

This register provides the trigger bit for setting the ENCAn to the operation stopped state.

Access: This register is a write-only register that can be written in 8-bit units.
It is always read as 00_H.

Address: <ENCAn_base> + $1C_H$

Value after reset: 00_H

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

Table 28.20 ENCAnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ENCAnTT	Timer Stop Trigger This is the trigger bit that sets the ENCAn to the operation stopped state. 0: Writing is ignored. 1: Clears ENCAnTE.ENCAnTE to “0”, to set the ENCAn to the count operation stopped state.

28.3.13 ENCAnEMU — ENCAn Emulation Register

This register controls operations by SVSTOP.

Access: This register can be read or written in 8-bit units.
Writing to this register should be performed in the counter operation stopped status (ENCAnTE.ENCAnTE = 0 and EPC.SVSTOP = 0).

Address: <ENCAn_base> + 48_H

Value after reset: 00_H

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

Table 28.21 ENCAnEMU Register Contents

Bit Position	Bit Name	Function
7	ENCAnSVSDIS	• When EPC.SVSTOP bit = 0: The count clock continues to be provided when the debugger assumes control of the microcontroller (at a break point, etc.), regardless of the value of this bit (1 or 0). • When EPC.SVSTOP bit = 1: 0: The count clock is stopped when the debugger assumes control of the microcontroller (at a break point, etc.). 1: The count clock continues to be provided when the debugger assumes control of the microcontroller (at a break point, etc.).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

28.4 Operation

The ENCAn operates the timer counter with counter up/down control and clear control by encoder inputs. The ENCAnCCR0 and ENCAnCCR1 registers can be used as dedicated compare registers or as dedicated capture registers.

28.4.1 Timer Counter Operation

The timer counter operations of the ENCAn are described below.

The figure below shows the operation phases. See the corresponding section with the section number for detailed descriptions on each operation.

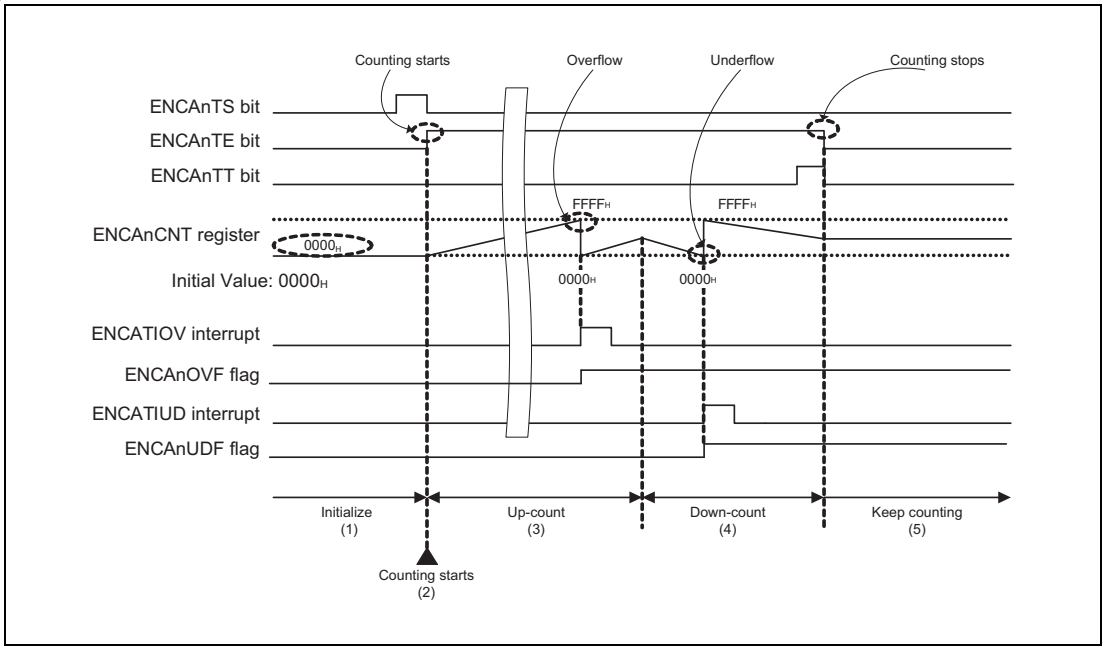


Figure 28.2 Timer Counter Initial Value Setting/Start/Stop

(1) Timer Counter Initial Value Setting

The initial value of the ENCAn counter register (ENCAnCNT) can be set in the counter operation stopped status (ENCAnTE = 0).

(2) Timer Counter Startup

By writing “1” to the timer start trigger bit (ENCAnTS), the timer status enable bit (ENCAnTE) is set to “1”, the count operation is enabled, and counting operation is performed upon detection of the valid edge of the encoder input.

(3) Overflow Operation

An overflow occurs when up-counting is performed when the counter value is $FFFF_H$. If the counter value changes from $FFFF_H$ to 0000_H, an overflow interrupt (ENCATIOV) is generated, and the overflow flag (ENCAnOVF) is set to “1”. The overflow flag (ENCAnOVF) is cleared to “0” when “1” is set to the overflow flag clear bit (ENCAnCLOV). For details about the operation, see **Section 28.6.1, Overflow Occurrence and Overflow Flag Clear Operation**.

(4) Underflow Operation

An underflow occurs when down-counting is performed when the counter value is 0000_H. If the counter value changes from 0000_H to $FFFF_H$, an underflow interrupt (ENCATIUD) is generated, and the underflow flag (ENCAnUDF) is set to “1”. The underflow flag (ENCAnUDF) is cleared to “0” when “1” is set to the underflow flag clear bit (ENCAnCLUD). For details about the operation, see **Section 28.6.2, Underflow Occurrence and Underflow Flag Clear Operation**.

(5) Timer Counter Stop

By writing “1” to the timer stop trigger bit (ENCAnTT), the timer status enable bit (ENCAnTE) is cleared to “0”, and the count operation is stopped. At this time, the timer counter is not reset to 0000_H and holds the value before count operation stop.

28.4.2 Up/Down Control of Timer Counter

Up/down control is performed by judging the phase of the encoder inputs (ENCAnE0, ENCAnE1) according to the settings of ENCAnUDS1 and ENCAnUDS0.

28.4.2.1 When the ENCAnUDS1/ENCAnUDS0 Bits in the ENCAnCTL Register = 00_B

Table 28.22 When ENCAnUDS1/ENCAnUDS0 Bits = 00_B

ENCAnUDS1	ENCAnUDS0	Operation Description		
		ENCAnE0 Pin	ENCAnE1 Pin	Count Operation
0	0	Rising edge	High level	Down
		Falling edge		
		Rising and falling edges		
		Rising edge	Low level	Up
		Falling edge		
		Rising and falling edges		

The valid edge for ENCAnE0 is specified by setting ENCAnEIS1 and ENCAnEIS0.

Up/down count operation is performed when the valid edges and levels of ENCAnE0 and ENCAnE1 match.

The following timing chart shows the count operation when ENCAnUDS1 and ENCAnUDS0 bits = 00_B.

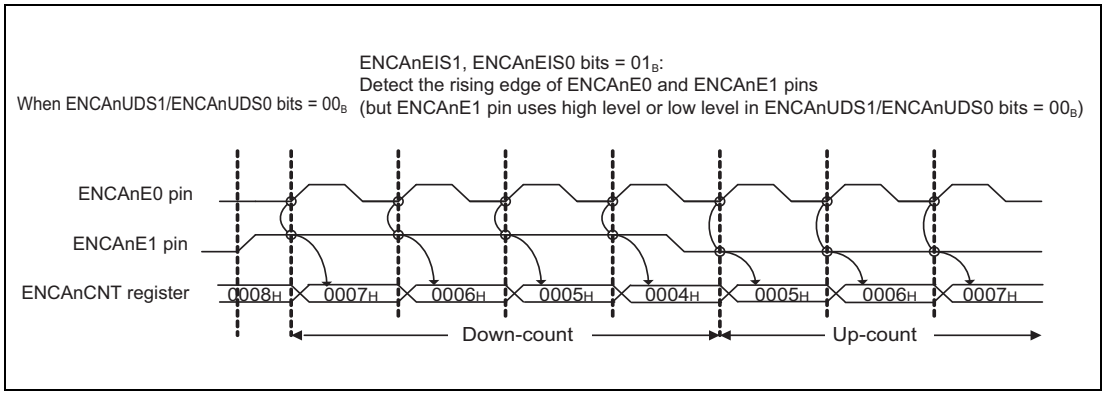


Figure 28.3 Count Operation when the ENCAnUDS1/ENCAnUDS0 Bits in the ENCAnCTL Register = 00_B

28.4.2.2 When the ENCAnUDS1/ENCAnUDS0 Bits in the ENCAnCTL Register = 01_B

Table 28.23 When the ENCAnUDS1/ENCAnUDS0 Bits = 01_B

ENCAnUDS1	ENCAnUDS0	Operation Description		
		ENCAnE0 Pin	ENCAnE1 Pin	Count Operation
0	1	Low level	Rising edge	Down
			Falling edge	
			Rising and falling edges	
		High level	Rising edge	
			Falling edge	
			Rising and falling edges	
		Rising edge	Low level	Up
		Falling edge		
		Rising and falling edges		
		Rising edge	High level	
		Falling edge		
		Rising and falling edges		
		Simultaneous input		Hold

The valid edges for ENCAnE0 and ENCAnE1 are specified by setting ENCAnEIS1 and ENCAnEIS0.

Up/down count operation is performed when the valid edges and levels of the ENCAnE0/ENCAnE1 pins match, and the count is held when the valid edges overlap.

The following timing chart shows the count operation when ENCAnUDS1 and ENCAnUDS0 bits = 01_B.

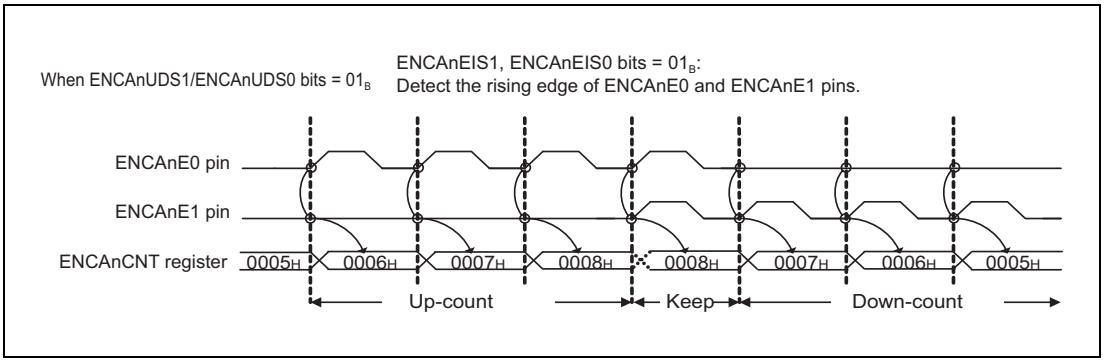


Figure 28.4 Count Operation when the ENCAnUDS1/ENCAnUDS0 Bits in the ENCAnCTL Register = 01_B

28.4.2.3 When the ENCAnUDS1 and ENCAnUDS0 Bits in the ENCAnCTL Register = 10_B

Table 28.24 When the ENCAnUDS1, ENCAnUDS0 Bits = 10_B

ENCAnUDS1	ENCAnUDS0	Operation Description		
		ENCAnE0 Pin	ENCAnE1 Pin	Count Operation
1	0	Rising edge	Low level	Down
		Rising edge	Falling edge	
		Falling edge	Low level	Up
		Falling edge	Falling edge	
		Low level	Rising edge	Hold
		Rising edge	Rising edge	
		High level	Rising edge	
		Falling edge	Rising edge	
		Low level	Falling edge	
		Rising edge	High level	
		High level	Falling edge	
		Falling edge	High level	

The valid edge specification for ENCAnE0 and ENCAnE1 (settings of ENCAnEIS1 and ENCAnEIS0) is invalid.

The following timing chart shows the count operation when the ENCAnUDS1/ENCAnUDS0 bits = 10_B.

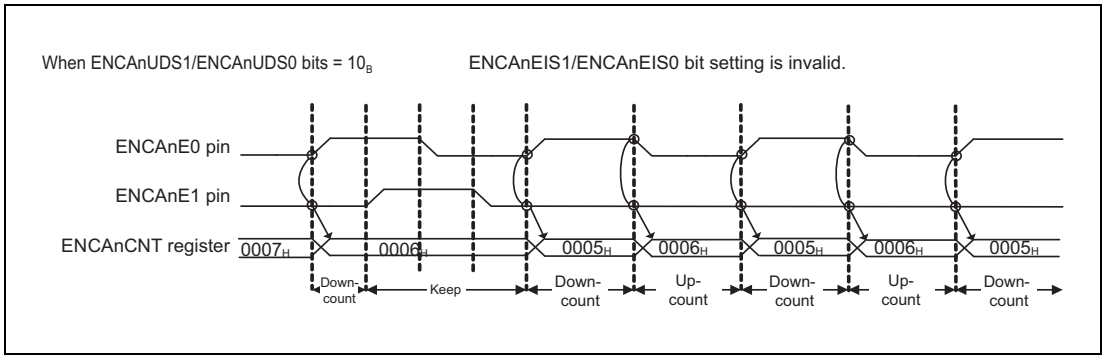


Figure 28.5 Count Operation when ENCAnUDS1/ENCAnUDS0 Bits in ENCAnCTL Register = 10_B

28.4.2.4 When ENCAnUDS1/ENCAnUDS0 Bits in the ENCAnCTL Register = 11_B

Table 28.25 When ENCAnUDS1/ENCAnUDS0 Bits = 11_B

ENCAnUDS1	ENCAnUDS0	Operation Description		
		ENCAnE0 Pin	ENCAnE1 Pin	Count Operation
1	1	Low level	Falling edge	Down
		Rising edge	Low level	
		High level	Rising edge	
		Falling edge	High level	
		Rising edge	High level	Up
		High level	Falling edge	
		Falling edge	Low level	
		Low level	Rising edge	
		Simultaneous input		Hold

Valid edge specification for ENCAnE0 and ENCAnE1 (settings of ENCAnEIS1 and ENCAnEIS0) is invalid.

The counter value is held when the valid edges of ENCAnE0 and ENCAnE1 overlap.

The following timing chart shows the count operation when ENCAnUDS1/ENCAnUDS0 Bits = 11_B.

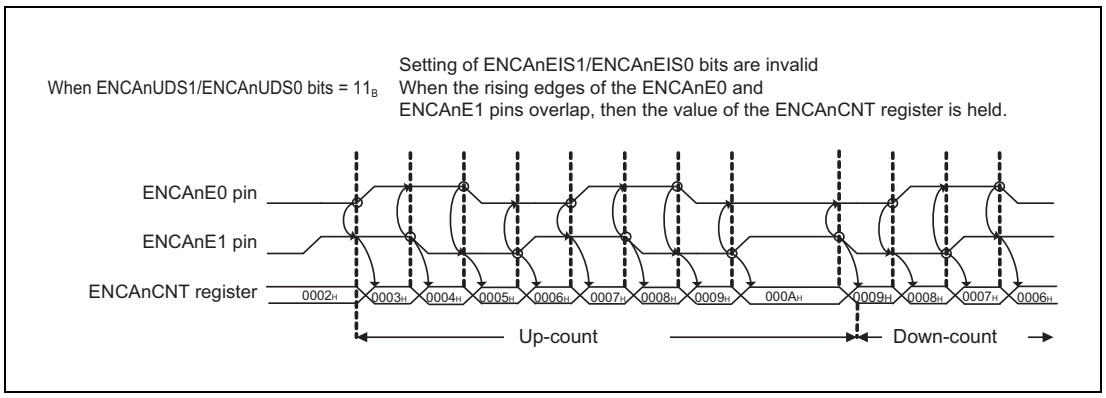


Figure 28.6 Count Operation when ENCAnUDS1 and ENCAnUDS0 Bits in the ENCAnCTL Register = 11_B

28.4.3 Timer Counter Clear Control by Encoder Input

The timer counter is cleared to 0000_H by encoder clear input (ENCAnEC).

Two types of clearing methods can be selected by controlling the ENCAnSCE, ENCAnZCL, ENCAnBCL, ENCAnACL, ENCAnECS1, and ENCAnECS0 bits of the ENCAnIOC1 register.

Table 28.26 Timer Counter Clear Control by Encoder Input

Clearing method	ENCAnSCE	ENCAnZCL	ENCAnBCL	ENCAnACL	ENCAnECS1, ENCAnECS0
(1)	0	Invalid	Invalid	Invalid	Valid
(2)	1	Valid	Valid	Valid	Invalid

28.4.3.1 Clearing Method when ENCAnSCE = 0

- Upon detection of the valid edge of ENCAnEC, the timer counter is cleared to 0000_H in synchronization with the operation clock.
- The valid edge of ENCAnEC is specified by the setting of the ENCAnECS1 and ENCAnECS0 bits.
- The settings of the ENCAnZCL, ENCAnBCL, and ENCAnACL bits are invalid.
- An encoder clear interrupt request signal (ENCATIEC) is output simultaneously with timer counter clearing.

For details about clear operation when ENCAnSCE = 0, see the timing chart in **Section 28.6.19, Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0**.

28.4.3.2 Clearing Method when ENCAnSCE = 1

- When the clear levels of the ENCAnEC, ENCAnE1, ENCAnE0 inputs are detected, the timer counter is cleared to 0000_H in synchronization with the operating clock.
- Specify the clear levels of the ENCAnEC, ENCAnE1, ENCAnE0 inputs by the settings of the ENCAnZCL, ENCAnBCL, and ENCAnACL bits.
- The settings of the ENCAnECS1 and ENCAnECS0 bits are invalid.
- An encoder clear interrupt request signal (ENCATIEC) is output simultaneously with timer counter clearing.

The clearing conditions of the timer counter according to the ENCAnZCL, ENCAnBCL, and ENCAnACL settings are listed in the table below.

Table 28.27 Clearing Conditions of the Timer Counter

Counter Clear Condition Setting			Encoder Pin Input Level		
ENCAnZCL	ENCAnBCL	ENCAnACL	ENCAnEC	ENCAnE1	ENCAnE0
0	0	0	Low	Low	Low
0	0	1	Low	Low	High
0	1	0	Low	High	Low
0	1	1	Low	High	High
1	0	0	High	Low	Low
1	0	1	High	Low	High
1	1	0	High	High	Low
1	1	1	High	High	High

28.4.4 Functions of ENCAnCCR0

28.4.4.1 Compare Function

- When ENCAnCRM0 = 0, the ENCAnCCR0 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCAnCCR0 setting value, a compare 0 match interrupt (ENCATINT0) is output.
- When ENCAnECM0 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is up-count.

Table 28.28 Compare Function of ENCAnCCR0

ENCAnCCR0 Function	Compare Match Clear Control	Next Count Operation	Timer Counter Clearing Upon Compare Match with ENCAnCCR0
ENCAnCRM0	ENCAnECM0		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Clears timer counter to 0000_H.
		Down-count	Does not clear (continues count operation).

When ENCAnLDE = 1

- Upon occurrence of an underflow, the setting value of the ENCAnCCR0 register is loaded to the timer counter.
- An underflow interrupt (ENCATIUD) is output.

NOTE

For the timing chart when ENCAnLDE = 1, see **Section 28.6.8, Using the ENCAnLDE Function Immediately after Startup** to **Section 28.6.12, Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input**.

28.4.4.2 Capture Function

- When ENCAnCRM0 = 1, the ENCAnCCR0 register functions as a dedicated capture register.
- Upon valid edge detection of the capture trigger input 0 (ENCATTIN0), the value of the timer counter is stored into ENCAnCCR0.
- A capture 0 interrupt (ENCATINT0) is output during capture operation.

NOTE

For details about capture operation for ENCAnCCR0, see the timing charts in **Section 28.6.14, Capture Operation between Count Clocks (ENCAnCCR0)** and **Section 28.6.17, Encoder Operation when Compare Match Clear Control is Disabled**.

28.4.5 Functions of ENCAnCCR1

28.4.5.1 Compare Function

- When ENCAnCRM1 = 0, the ENCAnCCR1 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCAnCCR1 setting value, a compare 1 match interrupt (ENCATINT1) is output.
- When ENCAnECM1 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is down-count.

Table 28.29 Compare Function of ENCAnCCR1

ENCAnCCR1 Function	Compare Match Clear Control	Next Count Operation	Timer Counter Clearing Upon Compare Match with ENCAnCCR1
ENCAnCRM1	ENCAnECM1		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Does not clear (continues count operation).
		Down-count	Clears timer counter to 0000_H.

Compare match interrupt mask function

- When ENCAnCME = 1, the compare 1 match interrupt mask function is enabled. In this state, the compare 1 match interrupt is output upon the first match of the value of the timer counter and the ENCAnCCR1 setting value, and interrupts are then masked for the second and subsequent compare matches.
- When ENCAnCME = 1 and ENCAnMCS = 0, a compare 1 match interrupt is output once upon the first compare match by writing to the ENCAnCCR1 register (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- When ENCAnCME = 1 and ENCAnMCS = 1, a compare 1 match interrupt is output once upon the first compare match by a timer counter clear operation accompanying encoder clear input or by a timer counter clear operation upon match between the ENCAnCCR0 register value and the timer counter value (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- When ENCAnCME = 1, ENCAnMCS = 1 and ENCAnLDE = 1, a compare 1 match interrupt is output once upon the first compare match by a loading operation of the ENCAnCCR0 register to the timer counter upon underflow detection (interrupts are masked for the second and subsequent matches until the cancel trigger occurs again).
- Setting ENCAnECM1 to “1” is prohibited when enabling the compare 1 match interrupt mask function.

Table 28.30 Compare Match Interrupt Mask Function of ENCAnCCR1

ENCAnCCR1 Function	Compare 1 Match Interrupt Mask	Interrupt Mask Cancel Trigger	Compare 1 Match Interrupt Output upon Compare Match with ENCAnCCR1
ENCAnCRM1	ENCAnCME	ENCAnMCS	
0 (Compare)	0 (Mask function disabled)	— (Setting invalid)	Outputs compare 1 match interrupt upon each compare match.
	1 (Mask function enabled)	0 (Write operation to ENCAnCCR1)	Outputs compare 1 match interrupt once upon the first compare match. (Interrupts are masked for the second and subsequent matches until the cancel trigger occurs.)
		1 (Timer counter clear operation) (Loading from ENCAnCCR0 to the timer counter upon underflow occurrence when ENCAnLDE = 1)	

28.4.5.2 Capture Function

When ENCAnCRM1 = 1, the ENCAnCCR1 register functions as a dedicated capture register.

NOTE

For details about capture operation to ENCAnCCR1, see the timing chart in **Section 28.6.13, Capture Operation between Count Clocks (ENCAnCCR1)**.

The operations for each of the ENCAnCTS settings are shown in the table below.

Table 28.31 Operations for Each of the ENCAnCTS Settings

ENCAnCCR1 Function	Capture Trigger Selection	Capture Trigger Signal	Timer Counter Clearing	Interrupt Occurrence
ENCAnCRM1	ENCAnCTS			
1 (Capture)	0	Capture trigger 1 input (ENCATTIN1)	Does not clear timer counter.	(1) Capture 1 interrupt (ENCATINT1)
	1	Encoder clear input (set with ENCAnSCE)	Clears timer counter.	(1) Capture 1 interrupt (ENCATINT1) (2) Encoder clear interrupt (ENCATIEC)

NOTE

For details about the timing chart when ENCAnCTS = 0 or ENCAnCTS = 1, see the following:

Section 28.6.3, Count Clearing and Capture Operation by Encoder Clear Input (ENCAnEC Pin), **Section 28.6.4, Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)**, **Section 28.6.5, Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)**, **Section 28.6.11, Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input (ENCAnEC Pin)** and **Section 28.6.12, Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input**.

28.4.5.3 Timer Counter Clearing upon Compare Register Match

Timer counter clearing upon compare match between the value of the timer counter and the ENCAnCCR0/1 setting value, according to the settings of the ENCAnECM1 and ENCAnECM0 bits in the ENCAnCTL register, is detailed in the following table.

Table 28.32 Timer Counter Clearing Operation upon Compare Register Match

ENCAnECM1 and ENCAnECM0	Next Count Operation	Timer Counter Clearing upon Compare Match with ENCAnCCR1	Timer Counter Clearing upon Compare Match with ENCAnCCR0
00	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
01	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000_H.
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
10	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Clears timer counter to 0000_H.	Does not clear (continues count operation).
11	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000_H.
	Down-count	Clears timer counter to 0000_H.	Does not clear (continues count operation).

28.4.6 Startup/Stop of Timer Counter

28.4.6.1 Startup of Timer

The timer operation can be started by setting the ENCAnTS bit to “1”.

PIC setting enables simultaneous start with other timers. For details, see **Section 29.8, Simultaneous Start Trigger Function**.

28.4.6.2 Stop of Timer

When the ENCAnTT bit is set to “1”, the ENCAnTE bit becomes “0” and the timer stops.

28.5 ENCAn Setting Sequences

28.5.1 ENCAn Setting Procedure

The ENCAn setting procedure is described below.

Table 28.33 ENCAn Setting Procedure

Initial Setting	Action	Setting status
Initial setting	Reset deassertion	Power-on status, operation stopped status. (Writing to each register is enabled)
ENCAn initial setting	Perform the following initial settings. • Setting for counter • Setting for counter clear • Setting for ENCAnCCR0 register • Setting for ENCAnCCR1 register	This is the count operation stopped status. The value of the ENCAnTE bit indicating the operating status is 0.
	Perform the counter initial value settings. • Set any 16-bit value to ENCAnCNT register. (When, after setting this register, the ENCAnTS bit is set to “1”, the counter operation starts from the set count value.)	The set value is set as the initial value of the counter register.
Operation start	Perform the counter operation start setting. • Set the ENCAnTS bit to “1”.	This is the counter operation starts status. The value of the ENCAnTE bit indicating the operating status is 1, and the count clock is supplied to the internal circuit.
Operating	Only those registers whose setting can be changed during operation can be rewritten. • ENCAnCCR0 register setting. • ENCAnCCR1 register setting. • ENCAnIOC0 register setting.	The count operation set with the initial setting is performed, and up/down counting is performed according to ENCAnE0 and ENCAnE1 pins.
Operation stop	Perform the counter operation stop setting during operation. • Set the ENCAnTT bit to “1”.	This is the counter operation stopped status. The value of the ENCAnTE bit indicating the operating status is 0.
ENCAn stop	Reset	The setting registers are initialized.

28.5.1.1 Initial Setting Procedure for the Counter

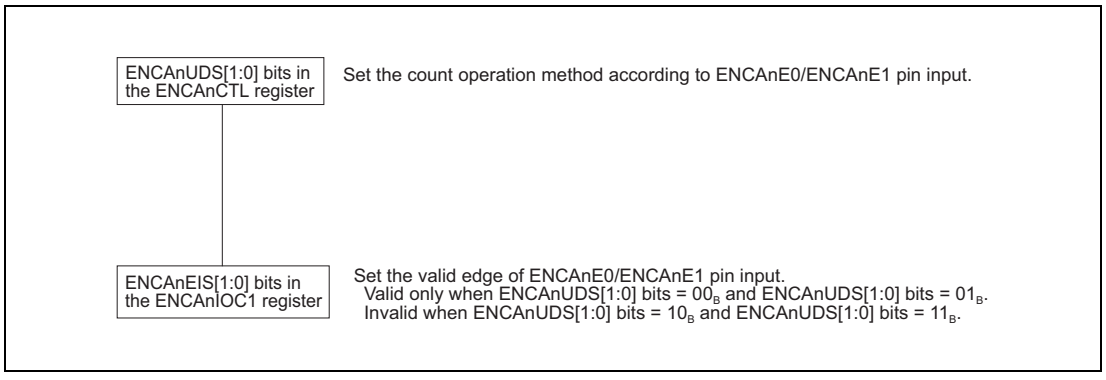


Figure 28.7 Initial Setting Procedure for the Counter

28.5.1.2 Initial Setting Procedure for Counter Clear

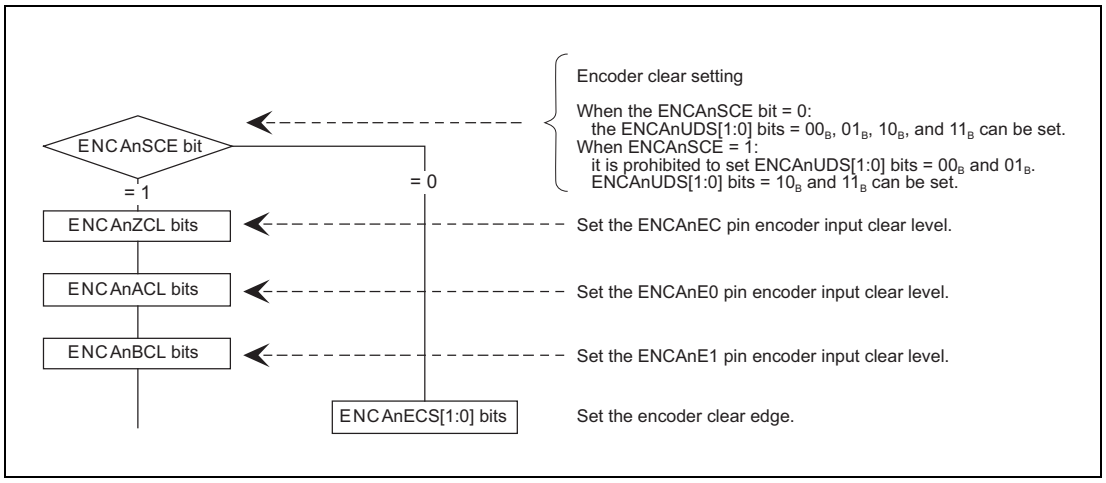


Figure 28.8 Initial Setting Procedure for Counter Clear

28.5.1.3 Setting Procedure for ENCAnCCR0 Register

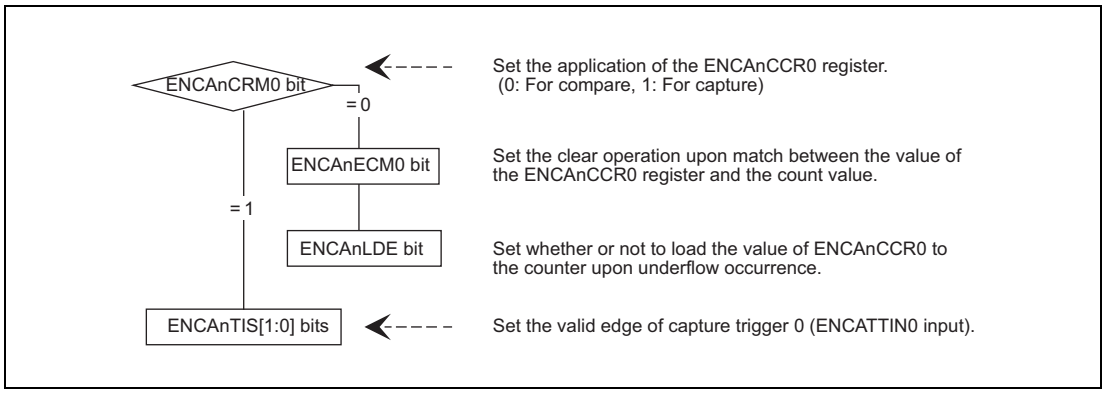


Figure 28.9 Setting Procedure for ENCAnCCR0 Register

28.5.1.4 Setting Procedure for ENCAnCCR1 Register

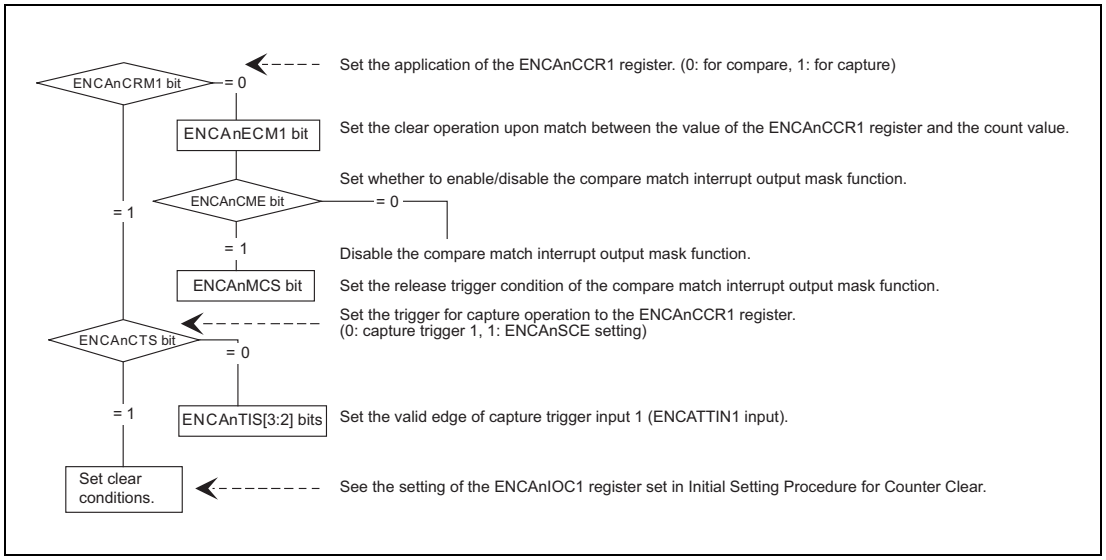


Figure 28.10 Setting Procedure for ENCAnCCR1 Register

28.6 Timing Chart

28.6.1 Overflow Occurrence and Overflow Flag Clear Operation

An overflow occurs when up-counting is performed when the counter value is $FFFF_H$. Once an overflow occurs, an overflow interrupt (ENCATIOV) is output and the overflow flag (ENCAnOVF) is set to 1. When the overflow clear bit (ENCAnCLOV) is set to 1, the overflow flag (ENCAnOVF) is cleared to 0.

The operations of overflow occurrence and overflow flag clearing are described below.

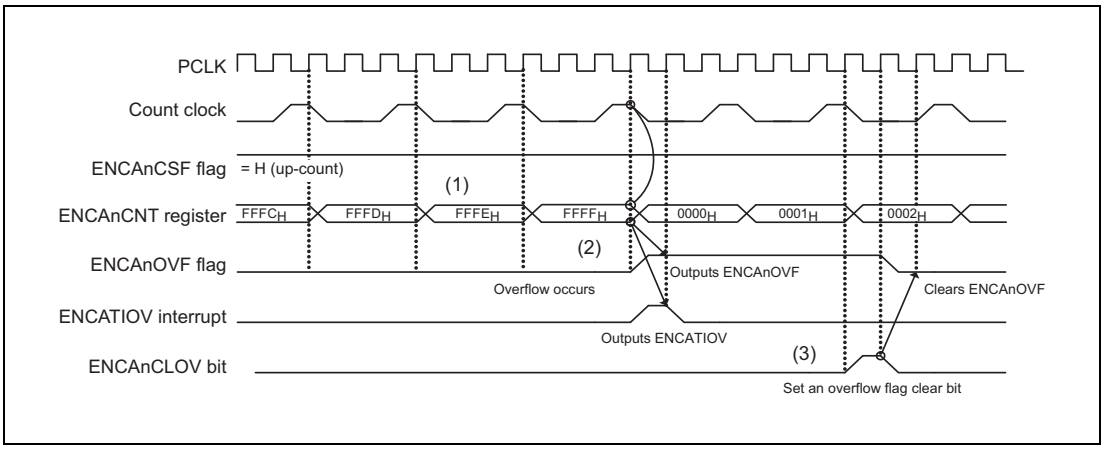


Figure 28.11 Settings of Overflow Occurrence and Overflow Flag Clear

(1) The count value is counted up from $FFFE_H$ to $FFFF_H$.

(2) When the count value changes from $FFFF_H$ to 0000_H, an overflow occurs. At the same time, an overflow interrupt is output and the overflow flag is set to 1.

(3) By setting the ENCAnCLOV bit in the ENCAnFGC register to 1 by the overflow flag clearing method, the overflow flag is cleared to 0. The overflow flag is also cleared by setting the ENCAnTS bit in the ENCAnTS register to 1 when the ENCAnTE bit in the ENCAnTE register is 0, or setting the input signal of ENCATSST (simultaneous start trigger input) to “High”.

28.6.2 Underflow Occurrence and Underflow Flag Clear Operation

An underflow occurs when down-counting is performed when the counter value is 0000_H.
Once an underflow occurs, an underflow interrupt (ENCATIUD) is output and the underflow flag (ENCAnUDF) is set to 1. When the underflow clear bit (ENCAnCLUD) is set to 1, the underflow flag (ENCAnUDF) is cleared to 0.

The operations of underflow occurrence and underflow flag clearing are described below.

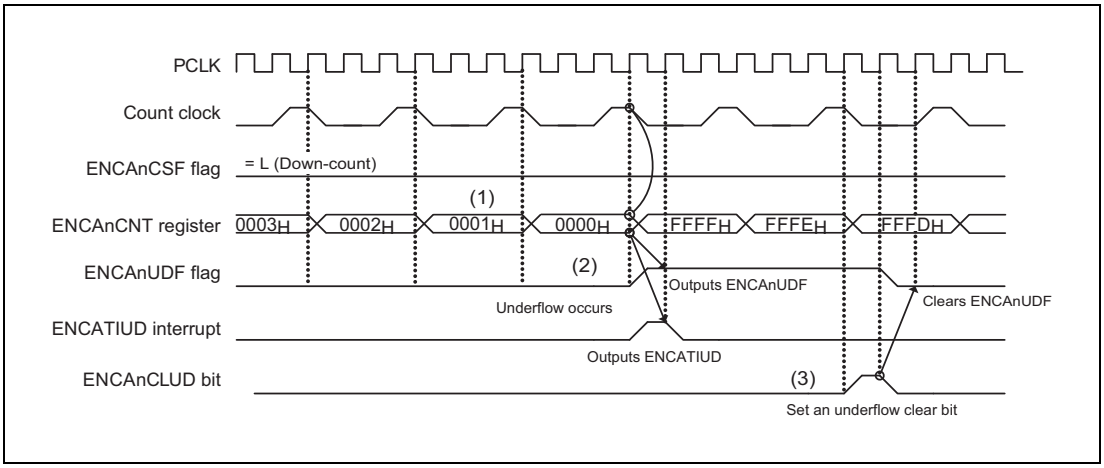


Figure 28.12 Settings of Underflow Occurrence and Underflow Flag Clear

(1) The count value is counted down from 0001_H to 0000_H.

(2) When the count value changes from 0000_H to $FFFF_H$, an underflow occurs. At the same time, an underflow interrupt is output and the underflow flag is set to 1.

(3) By setting the ENCAnCLUD bit in the ENCAnFGC register to 1 by the underflow flag clearing method, the underflow flag is cleared to 0. The underflow flag is also cleared by setting the ENCAnTS bit in the ENCAnTS register to 1 when the ENCAnTE bit in the ENCAnTE register is 0, or by setting the input signal of ENCATSST (simultaneous start trigger) to “High”.

28.6.3 Count Clearing and Capture Operation by Encoder Clear Input (ENCAnEC Pin)

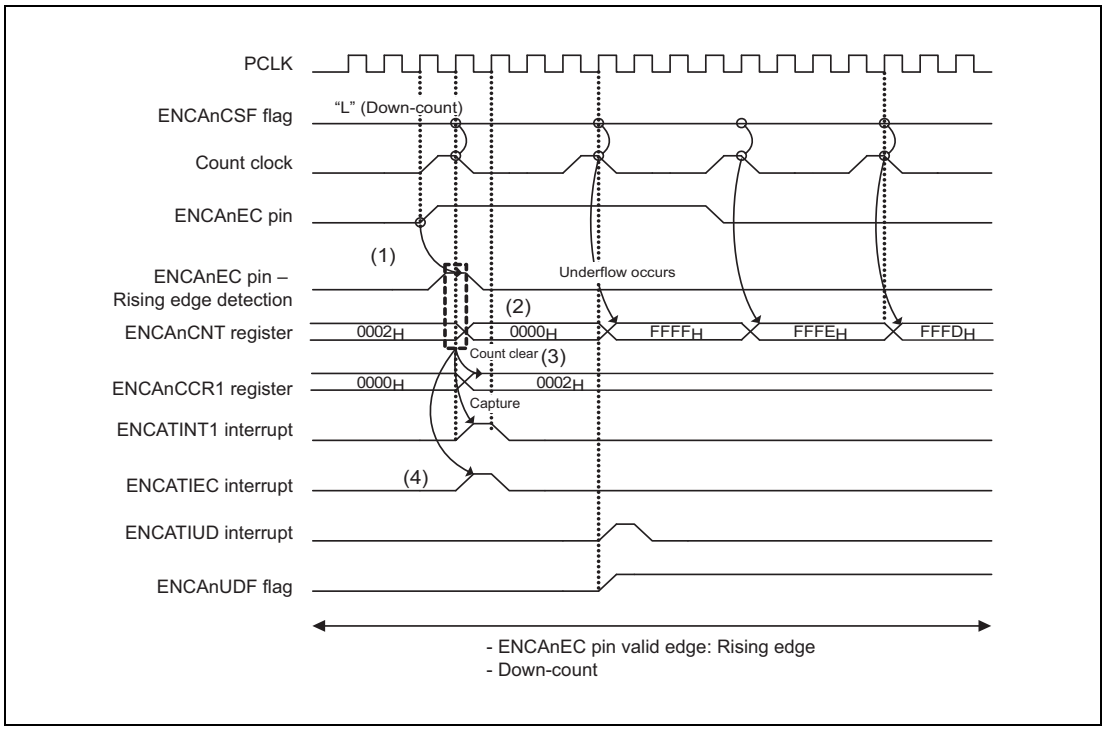


Figure 28.13 Timing Chart of Count Clearing and Capture Operation by Encoder Clear Input (ENCAnEC Pin)

Setting conditions

- ENCAnCRM1 bit in the ENCAnCTL register = 1
  (Select the ENCAnCCR1 register as capture.)
- ENCAnCTS bit in the ENCAnCTL register = 1
  (Select the ENCAnEC pin input as capture trigger input.)
- ENCAnECS1 and ENCAnECS0 bits in the ENCAnIOC1 register = 01_B
  (Select the ENCAnEC pin input as rising edge detection.)

(1) Capture operation is performed by the rising edge of the ENCAnEC pin input trigger.

(2) Clearing is performed by the ENCAnEC pin input and the count value is set to 0000_H.

(3) The counter value (0002_H) is captured in the ENCAnCCR1 register by the rising edge of the ENCAnEC pin input.

(4) At the same time, a clear interrupt (ENCATIEC) and capture interrupt (ENCATINT1) due to the ENCAnEC pin input are output.

28.6.4 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

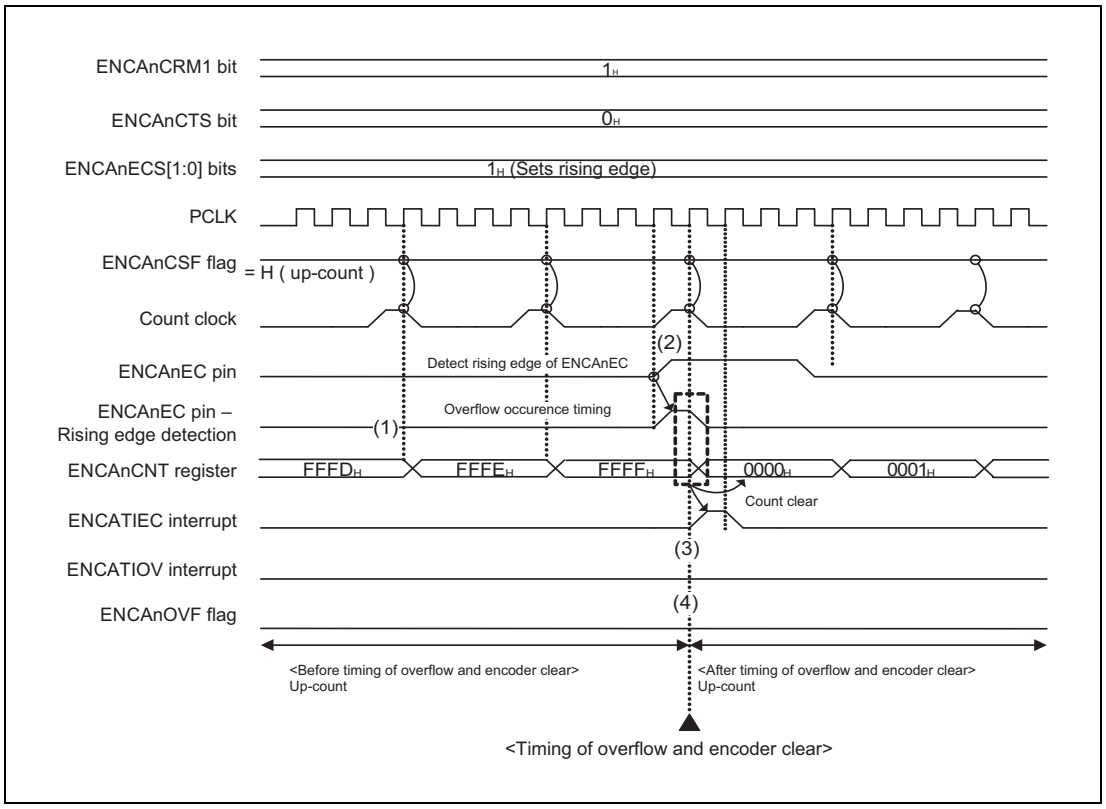


Figure 28.14 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC pin)

(1) An up-count from $FFFD_H$ is continuously performed.

(2) When an overflow occurs if the count value is $FFFF_H$, and the rising edge of ENCAnEC is detected simultaneously, clear operation by the encoder clear input is performed. The counter value is cleared to 0000_H.

(3) When the counter value is cleared by the encoder clear input, a clear interrupt (ENCATIEC) by encoder clear input is output simultaneously. Because a clear operation by the encoder clear input is performed simultaneously with the overflow occurrence, an overflow interrupt is not output (An overflow does not occur. Clear operation is performed by the encoder clear input).

(4) Because an overflow does not occur as is the case with step 3, the overflow flag is not set.

28.6.5 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

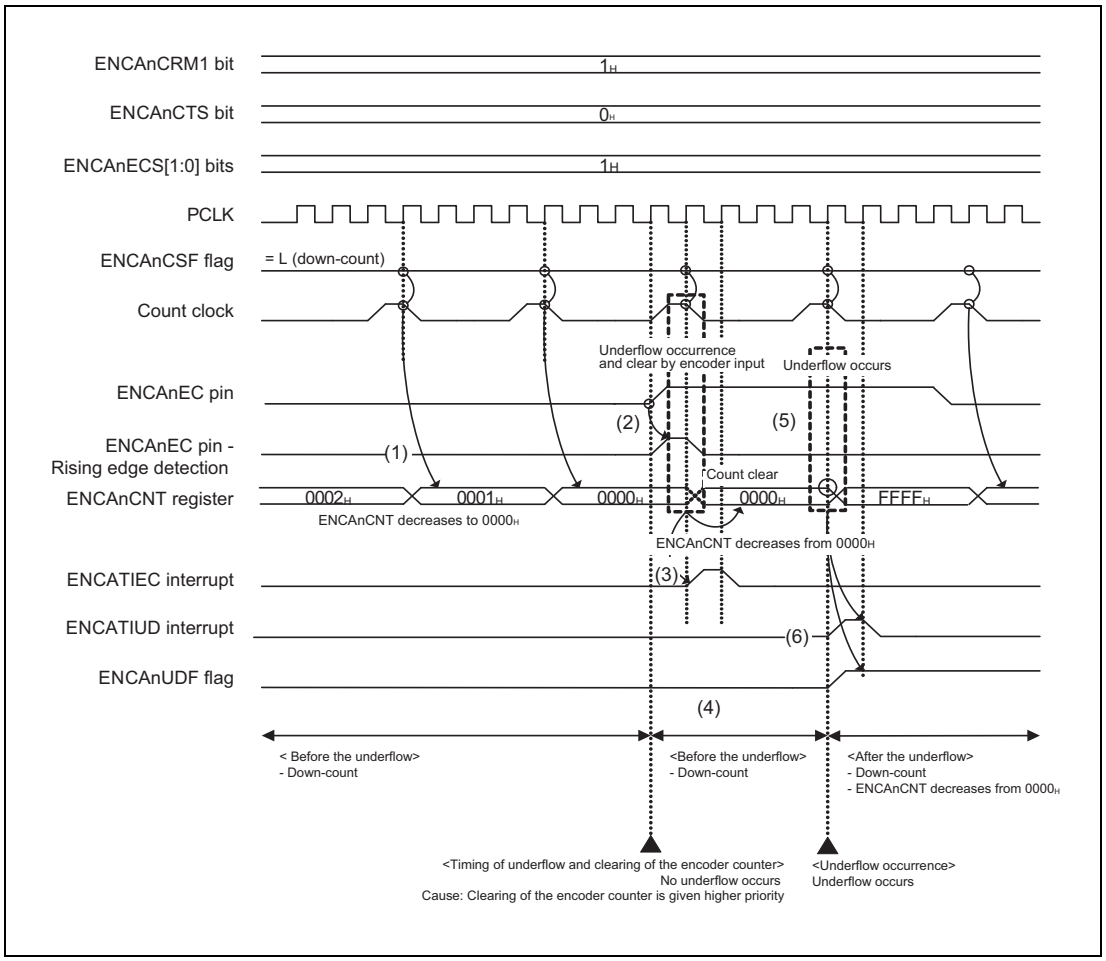


Figure 28.15 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

(1) A down-count from 0002_H is continuously performed.

(2) When an underflow occurs if the count value is 0000_H, and the rising edge of ENCAnEC is detected simultaneously, clear operation by the encoder clear input is performed. Even if the next clock signal is input during clear operation, the counter value remains at 0000_H.

(3) When the counter value is cleared by the encoder clear input, an encoder clear interrupt (ENCATIEC) is output simultaneously. Because a clear operation by the encoder clear input is performed simultaneously with the underflow occurrence, an underflow interrupt is not output (An underflow does not occur. Clear operation is performed by the encoder clear input).

(4) Because an underflow does not occur as is the case with step 3, the underflow flag is not set.

(5) When a further down-count is performed after the counter value changes to 0000_H by clear operation by the encoder clear input, the counter value changes from 0000_H to $FFFF_H$, and an underflow occurs.

(6) When an underflow occurs, an underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

28.6.6 Overflow Operation Immediately after Startup

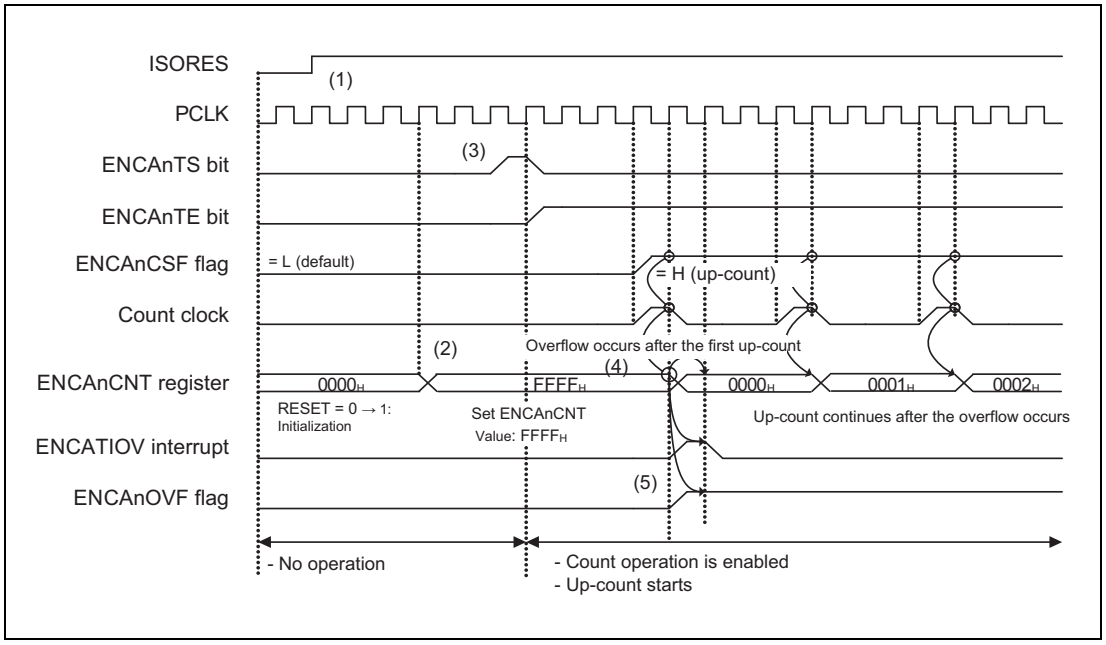


Figure 28.16 Overflow Operation Immediately after Startup

(1) When the ISORES value changes from “0” to “1”, the status changes from reset asserted to reset deasserted.

(2) The timer counter is set to $FFFF_H$ as the initial value.

(3) ENCAnTS is set to “1”, and operation starts. ENCAnTE changes to “1”, which indicates that operation is enabled.

(4) When an up-count is performed from $FFFF_H$ which is the initially set count value, the counter value changes from $FFFF_H$ to 0000_H, and an overflow occurs immediately after operation starts.

(5) At the same time, by an overflow occurrence immediately after operation starts, an overflow interrupt (ENCATIOV) is output, and the overflow flag (ENCAnOVF) is set.

28.6.7 Underflow Operation Immediately after Startup

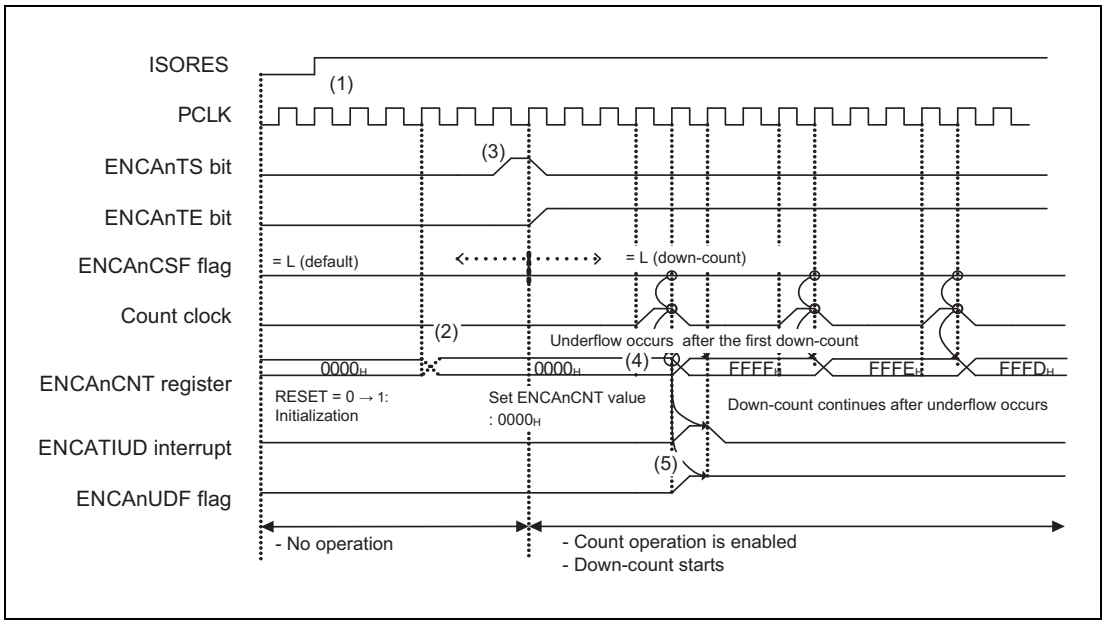


Figure 28.17 Underflow Operation Immediately after Startup

(1) When the ISORES value changes from “0” to “1”, the status changes from reset asserted to reset deasserted.

(2) The timer counter is set to 0000_H as the initial value.

(3) ENCAnTS is set to “1”, and operation starts. ENCAnTE changes to “1”, which indicates that operation is enabled.

(4) When a down-count is performed from 0000_H which is the initially set count value, the counter value changes from 0000_H to $FFFF_H$, and an underflow occurs immediately after operation starts.

(5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

28.6.8 Using the ENCAnLDE Function Immediately after Startup

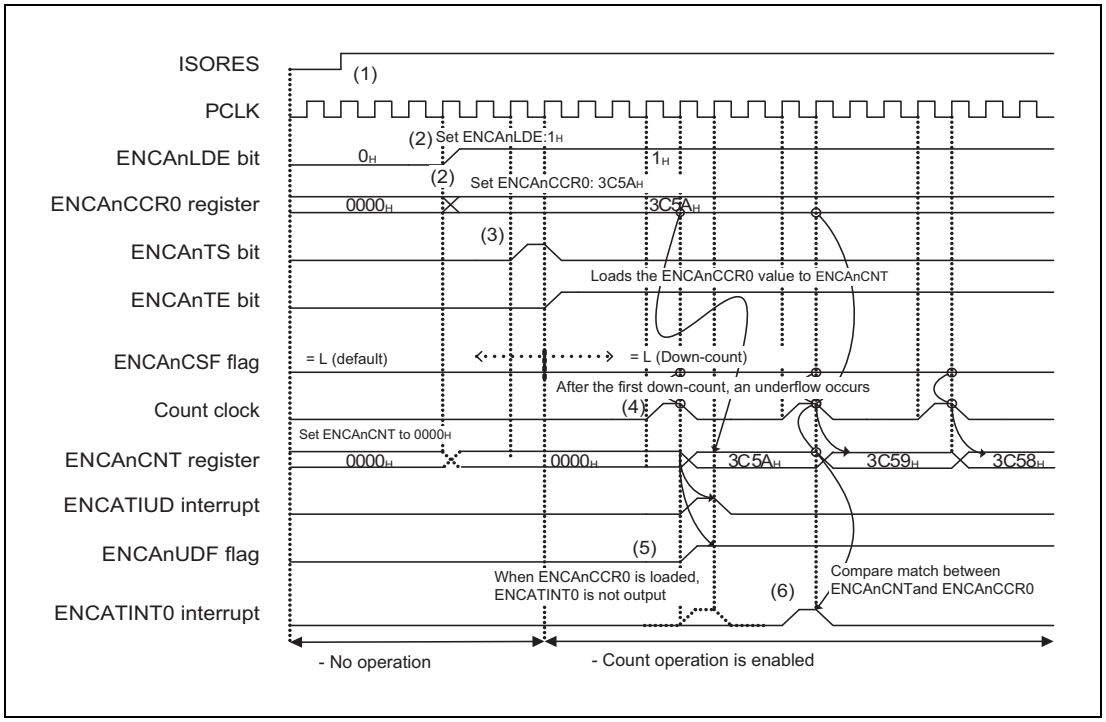


Figure 28.18 Using the ENCAnLDE Function Immediately after Startup

(1) When the ISORES value changes from “0” to “1”, the status changes from reset asserted to reset deasserted.

(2) The load enable bit (ENCAnLDE) is set to “1”, capture/compare register 0 (ENCAnCCR0) is set to $3C5A_H$, and the timer counter is set to the initial value 0000_H.

(3) ENCAnTS is set to “1”, and operation starts. ENCAnTE changes to “1”, which indicates that operation is enabled.

(4) When a down-count is performed from 0000_H which is the initially set count value, an underflow occurs immediately after operation starts. Because ENCAnLDE is set to “1”, the ENCAnCCR0 value, $3C5A_H$, is loaded to the timer counter (ENCATINT0 is not output during loading).

(5) At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set (after an underflow occurs, down-count operation from the loaded value ($3C5A_H$) continues).

(6) After the ENCAnCCR0 value is loaded to ENCAnCNT, a match with ENCAnCCR0 is detected, and ENCATINT0 is output.

28.6.9 ENCAnLDE Function (Loading Count Value)

(1) <When ENCAnLDE = 0>

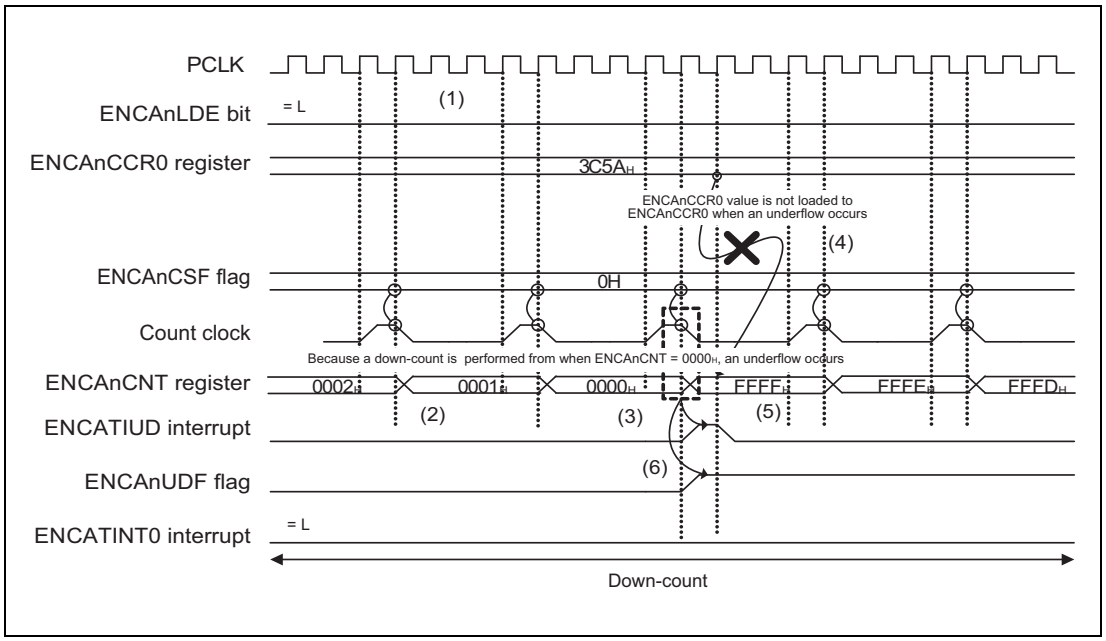


Figure 28.19 ENCAnLDE Function (when ENCAnLDE = 0)

(1) ENCAnLDE is set to “0” (even if an underflow occurs, the ENCAnCCR0 value is not loaded).

(2) A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$

(3) When a further down-count is performed after the counter value changes to 0000_H, an underflow occurs.

(4) Because ENCAnLDE is set to “0”, the setting value of the ENCAnCCR0 register is not loaded to the counter even if an underflow occurs.

(5) Operation changes to underflow operation (counter value: $0000_H \rightarrow FFFF_H$).

(6) An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

(2) <When ENCAnLDE = 1>

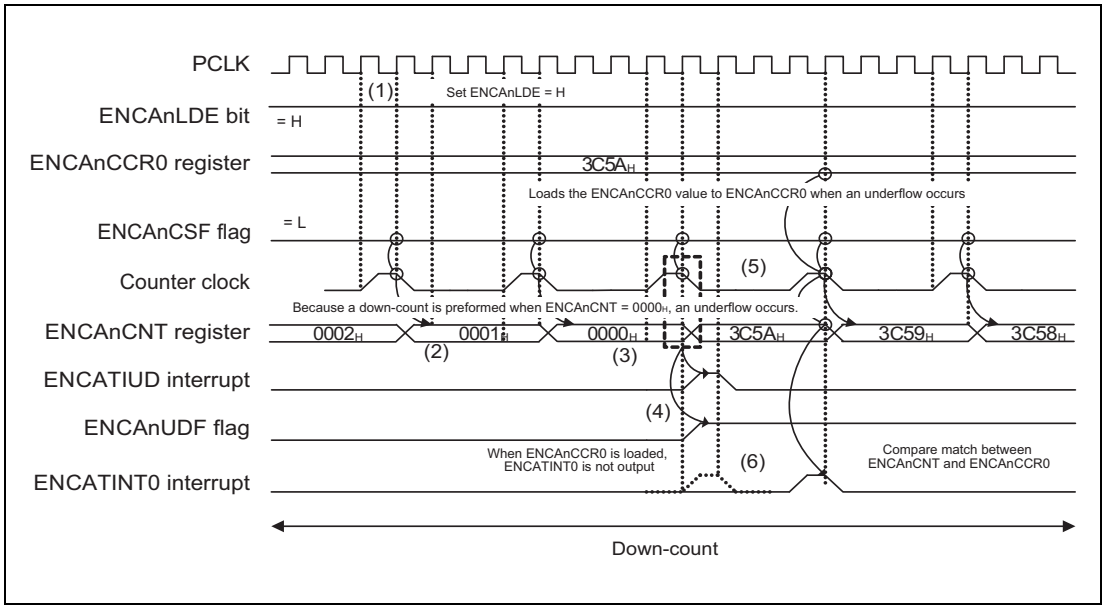


Figure 28.20 ENCAnLDE Function (when ENCAnLDE = 1)

(1) ENCAnLDE is set to “1” (if an underflow occurs, the ENCAnCCR0 value is loaded to the counter).

(2) A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$

(3) When a further down-count is performed after the counter value changes to 0000_H, an underflow occurs.

(4) An underflow interrupt is output, and the underflow flag is set.

(5) Because ENCAnLDE is set to “1”, the setting value of the ENCAnCCR0 register is loaded to the counter if an underflow occurs. ENCAnCNT is set to $3C5A_H$.

(6) After the ENCAnCCR0 value is set to ENCAnCNT, if the ENCAnCNT value matches the ENCAnCCR0 value on a count clock, a compare match interrupt (ENCATINT0) is output.

28.6.10 Conflict between ENCAnLDE Function (Loading Counter Value) and Rewrite of ENCAnCCR0 Register

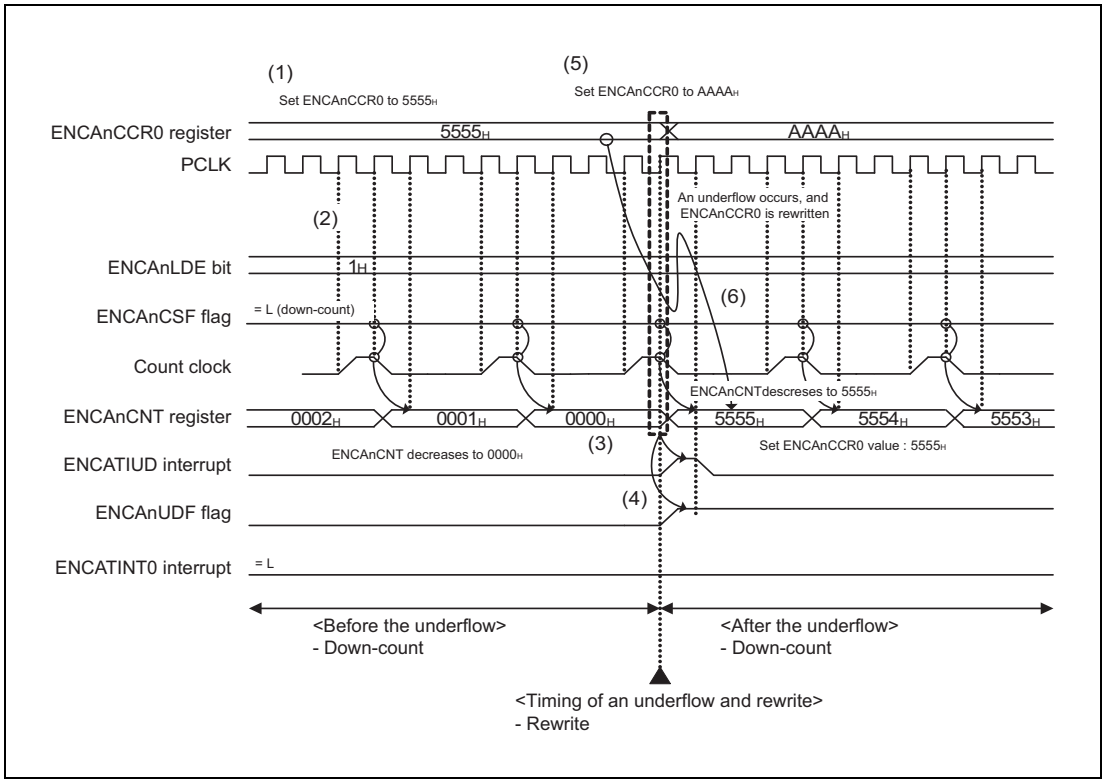


Figure 28.21 Conflict between ENCAnLDE Function and Rewrite of ENCAnCCR0 Register

(1) The ENCAnCCR0 register is currently set to 5555_H.

(2) ENCAnLDE is currently set to “1”.

(3) A down-count is performed ($0002_H \rightarrow 0001_H \rightarrow 0000_H$), and an underflow occurs.

(4) An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

(5) When an underflow occurs, the ENCAnCCR0 register value is changed from 5555_H to $AAAA_H$.

(6) Additionally, when an underflow occurs, the ENCAnCCR0 value before the rewrite was performed (5555_H) is set in ENCAnCNT.

28.6.11 Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

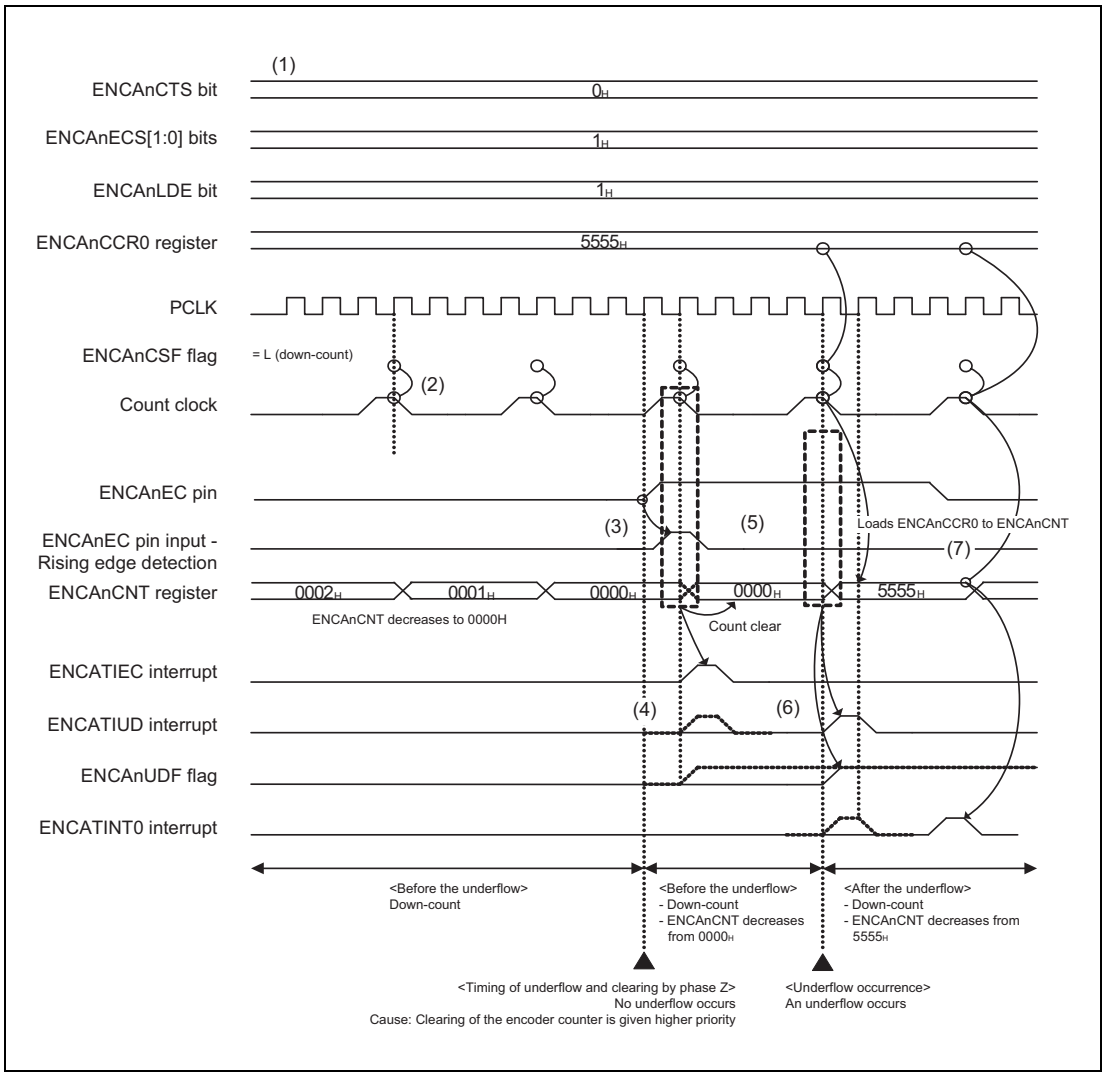


Figure 28.22 Conflict between ENCAnLDE Function and Clear Operation by Encoder Clear Input

(1) The values are set as follows: ENCAnCTS = 0, ENCAnECS[1:0] = 01_B, ENCAnLDE = 1, and ENCAnCCR0 = 5555_H.

(2) A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$

(3) When the count value becomes 0000_H, the rising edge of ENCAnEC pin is detected, and clear operation by the encoder clear input is performed.

(4) Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (ENCATIEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when the count value is 0000_H. Therefore, an underflow interrupt (ENCATIUD) is not output, and the underflow flag (ENCAnUDF) is not set.

(5) After the count value is cleared to 0000_H by clear operation by the encoder clear input, a down-count is performed and an underflow occurs.

(6) An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

(7) Because ENCAnLDE = “1”, if an underflow occurs, the ENCAnCCR0 value is loaded to ENCAnCNT.

(8) After the ENCAnCCR0 value is set to ENCAnCNT, a compare match is detected according to the count clock. If the ENCAnCNT value matches the ENCAnCCR0 value, a compare match interrupt (ENCATINT0) is output.

28.6.12 Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input

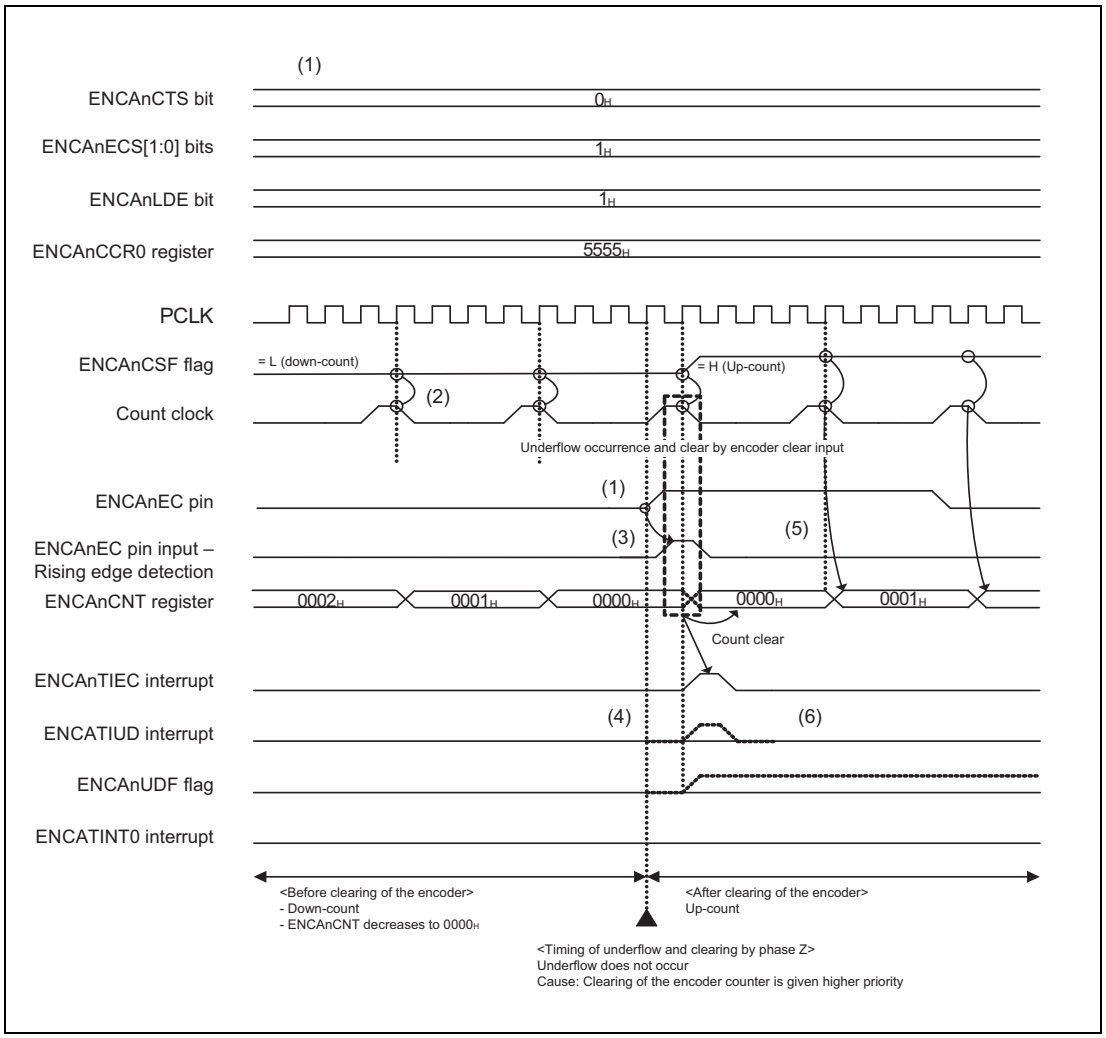


Figure 28.23 Up-count after Conflict between ENCAnLDE Function and Encoder Clear

(1) The values are set as follows: ENCAnCTS = 0, ENCAnECS[1:0] = 01_B, ENCAnLDE = 1, and ENCAnCCR0 = 5555_H.

(2) A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$

(3) When the count value becomes 0000_H, the rising edge of ENCAnEC pin is detected, and clear operation by the encoder clear input is performed.

(4) Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (ENCATIEC) by the encoder clear input is output. An underflow does not occur because a down-count is not performed when the count value is 0000_H. Therefore, an underflow interrupt (ENCATIUD) is not output, and the underflow flag (ENCAnUDF) is not set.

(5) After the count value is cleared to 0000_H by clear operation by the encoder clear input, an up-count is performed.

(6) An underflow interrupt (ENCATIUD) is not output, and the underflow flag (ENCAnUDF) is not set.

28.6.13 Capture Operation between Count Clocks (ENCAnCCR1)

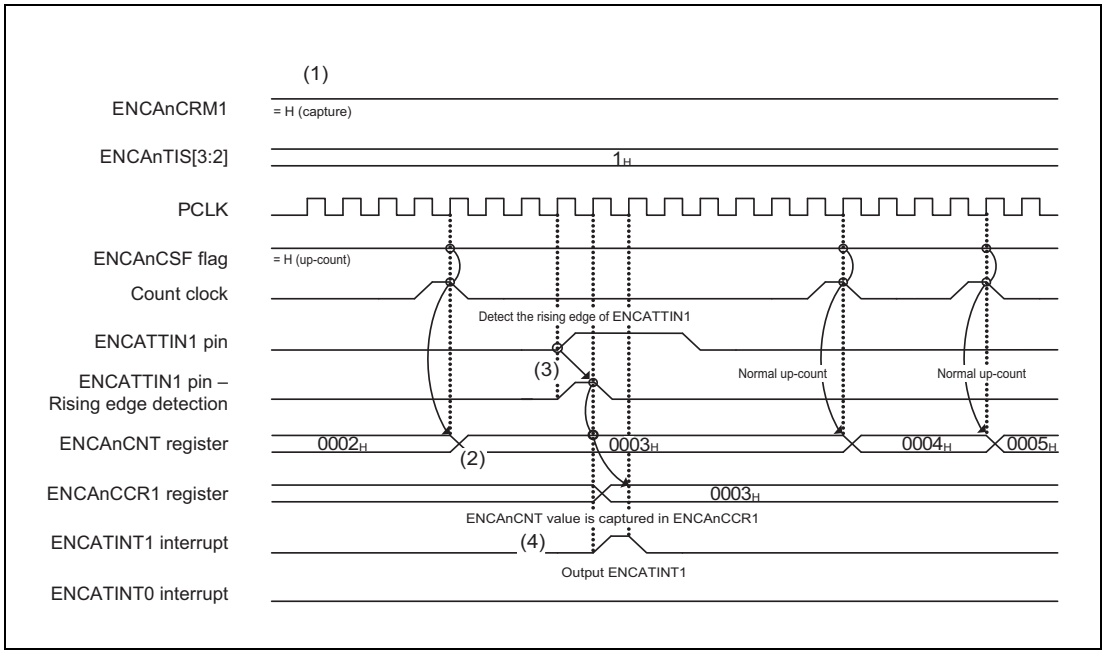


Figure 28.24 Capture Operation between Count Clocks (ENCAnCCR1)

(1) The values are set as follows: ENCAnCRM1 = 1, and ENCAnTIS[3:2] = 01_B.

(2) An up-count is performed.

(3) The rising edge of the ENCATTIN1 input is detected, and the count value is captured in ENCAnCCR1.

(4) An interrupt (ENCATINT1) corresponding to the capture to the ENCAnCCR1 register is output.

28.6.14 Capture Operation between Count Clocks (ENCAnCCR0)

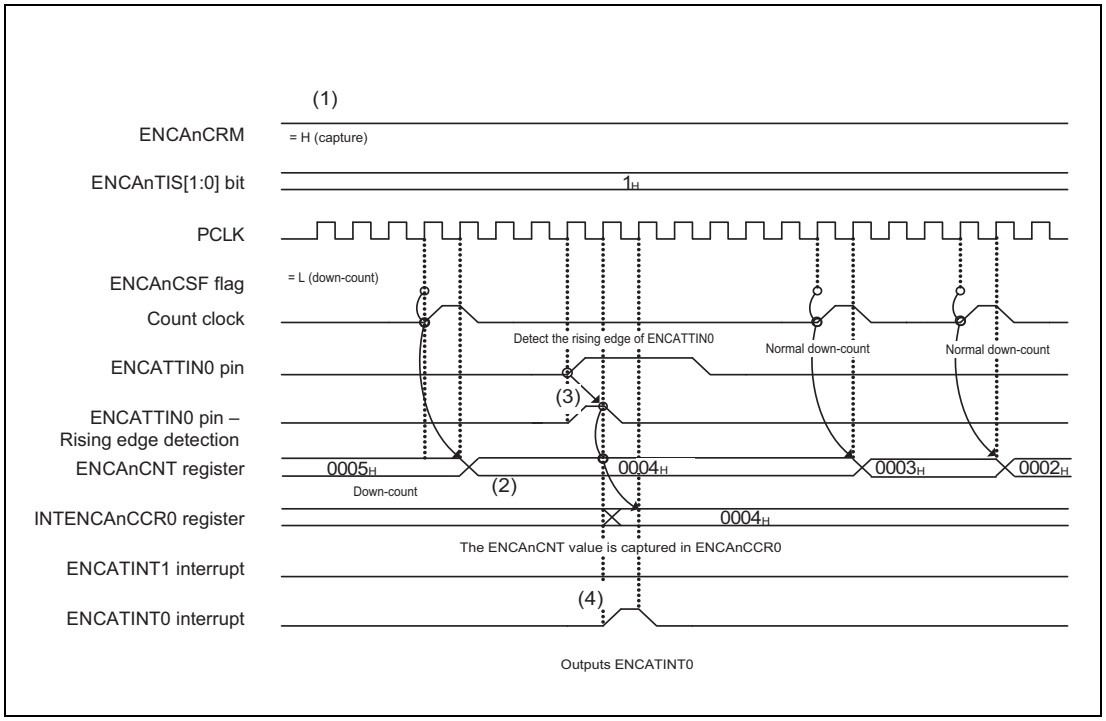


Figure 28.25 Capture Operation between Count Clocks (ENCAnCCR0)

(1) The values are set as follows: ENCAnCRM0 = 1, and ENCAnTIS[1:0] = 01_B.

(2) A down-count is performed.

(3) The rising edge of the ENCATTIN0 input is detected, and the count value is captured in ENCAnCCR0.

(4) An interrupt (ENCATINT0) corresponding to the capture to the ENCAnCCR0 register is output.

28.6.15 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 0

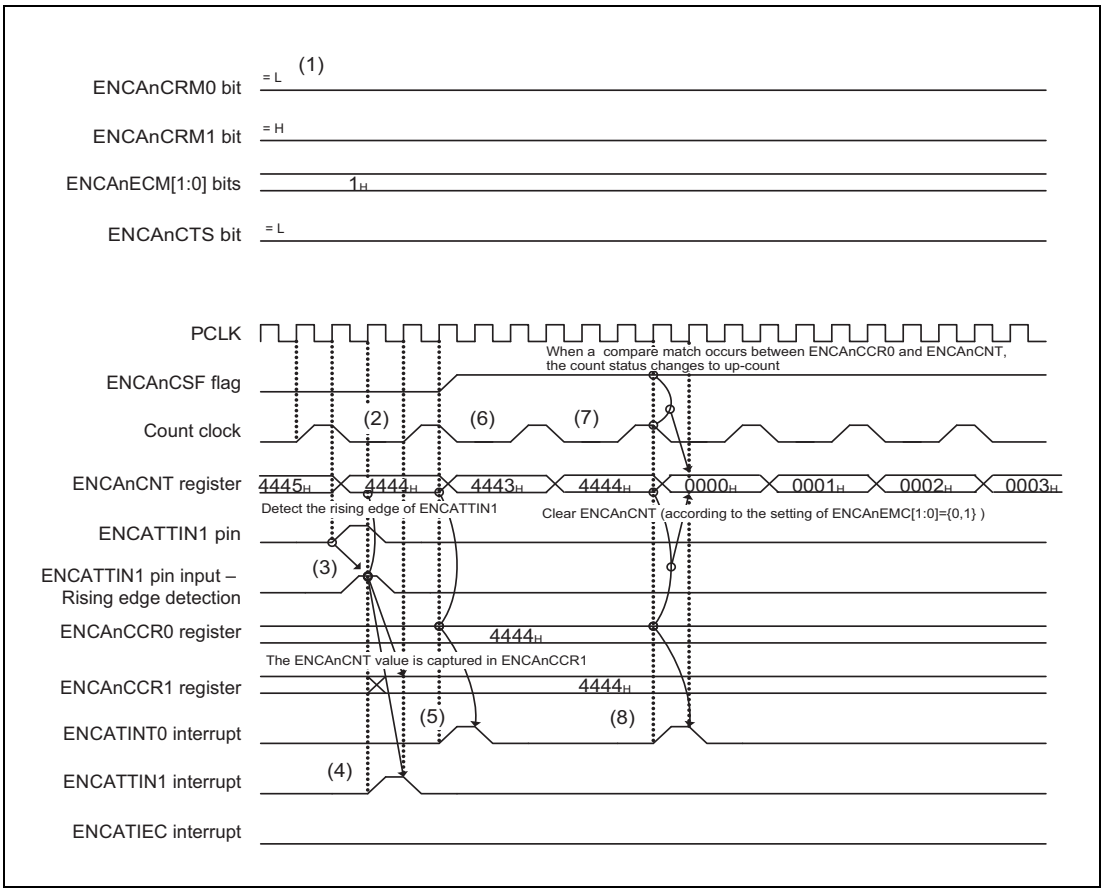


Figure 28.26 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 0

(1) The values are set as follows: ENCAnCCR0 = 4444_H, ENCAnCRM0 = 0, ENCAnCRM1 = 1, ENCAnECM[1:0] = 01_B, and ENCAnCTS = 0.

(2) A down-count is performed.

(3) The rising edge of the ENCATTIN1 input is detected, and the ENCAnCNT value (4444_H) is captured in the ENCAnCCR1 register.

(4) An interrupt signal (ENCATINT1) corresponding to the capture to the ENCAnCCR1 register is output.

(5) When a compare match occurs between ENCAnCNT (counted down from 4445_H to 4444_H) and ENCAnCCR0 (4444_H), a compare match interrupt (ENCATINT0) with ENCAnCCR0 is output.

(6) The count operation changes to up-count.

(7) When ENCAnCNT is counted up from 4443_H to 4444_H, a compare match with ENCAnCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCAnECM1 and ENCAnECM0 (01_B), and the ENCAnCNT value changes to 0000_H.

(8) When ENCAnCNT changes to 4444_H, a compare match interrupt (ENCATINT0) with ENCAnCCR0 is output.

28.6.16 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 1

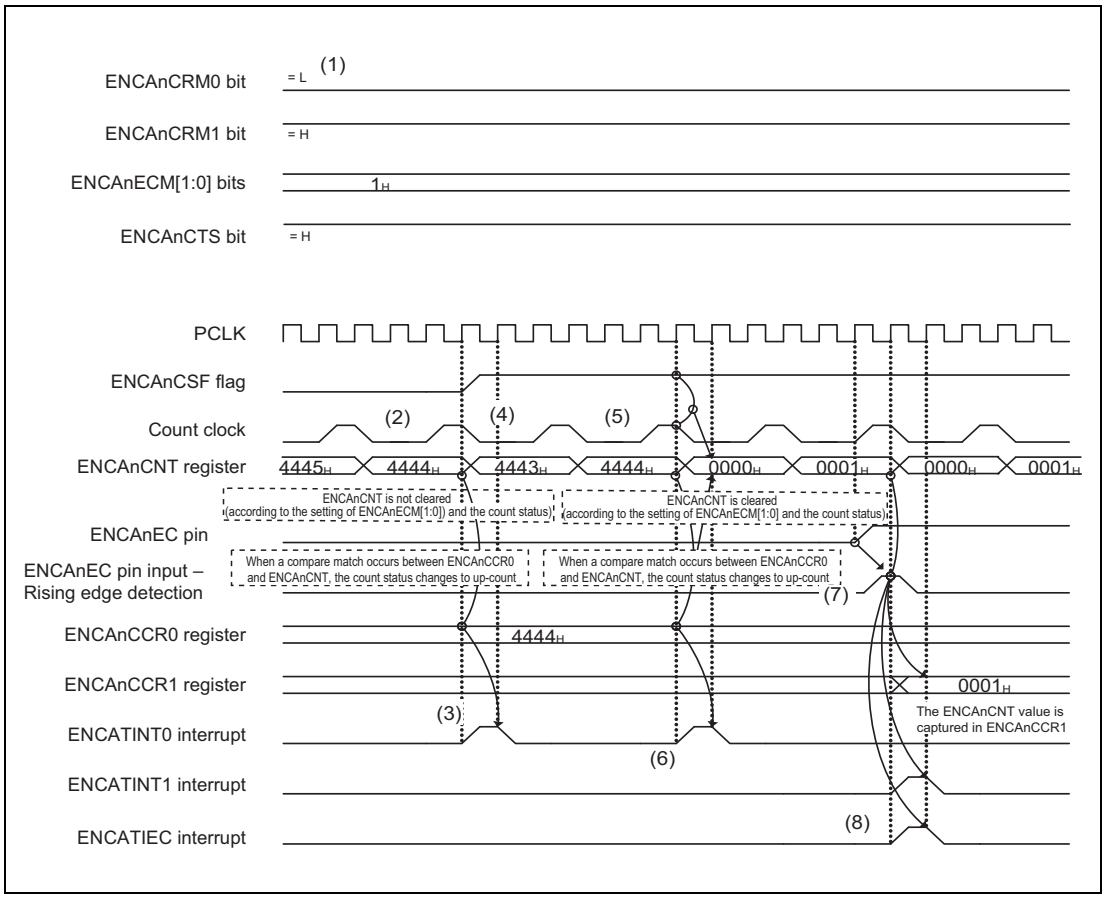


Figure 28.27 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 1

(1) The values are set as follows: ENCAnCCR0 = 4444_H, ENCAnCRM0 = 0, ENCAnCRM1 = 1, ENCAnECM[1:0] = 01_B, and ENCAnCTS = 1.

(2) A down-count is performed.

(3) When a compare match occurs between ENCAnCNT (counted down from 4445_H to 4444_H) and ENCAnCCR0 (4444_H), a compare/capture interrupt (ENCATINT0) is output.

(4) The count operation changes to up-count.

(5) When ENCAnCNT is counted up from 4443_H to 4444_H, a compare match with ENCAnCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCAnECM1 and ENCAnECM0 (01_B), and the ENCAnCNT value changes to 0000_H.

(6) When ENCAnCNT changes to 4444_H, a compare match interrupt (ENCATINT0) with ENCAnCCR0 is output.

(7) After the count value is cleared, an up-count is performed, and the count value changes to 0001_H. At this point, the ENCAnCNT value (0001_H) is captured in ENCAnCCR1 by detecting the rising edge of the ENCAnEC signal, and the counter is cleared to 0000_H.

(8) An interrupt (ENCATINT1) corresponding to the capture to the ENCAnCCR1 register and a clear interrupt (ENCATIEC) by ENCAnEC are output.

28.6.17 Encoder Operation when Compare Match Clear Control is Disabled

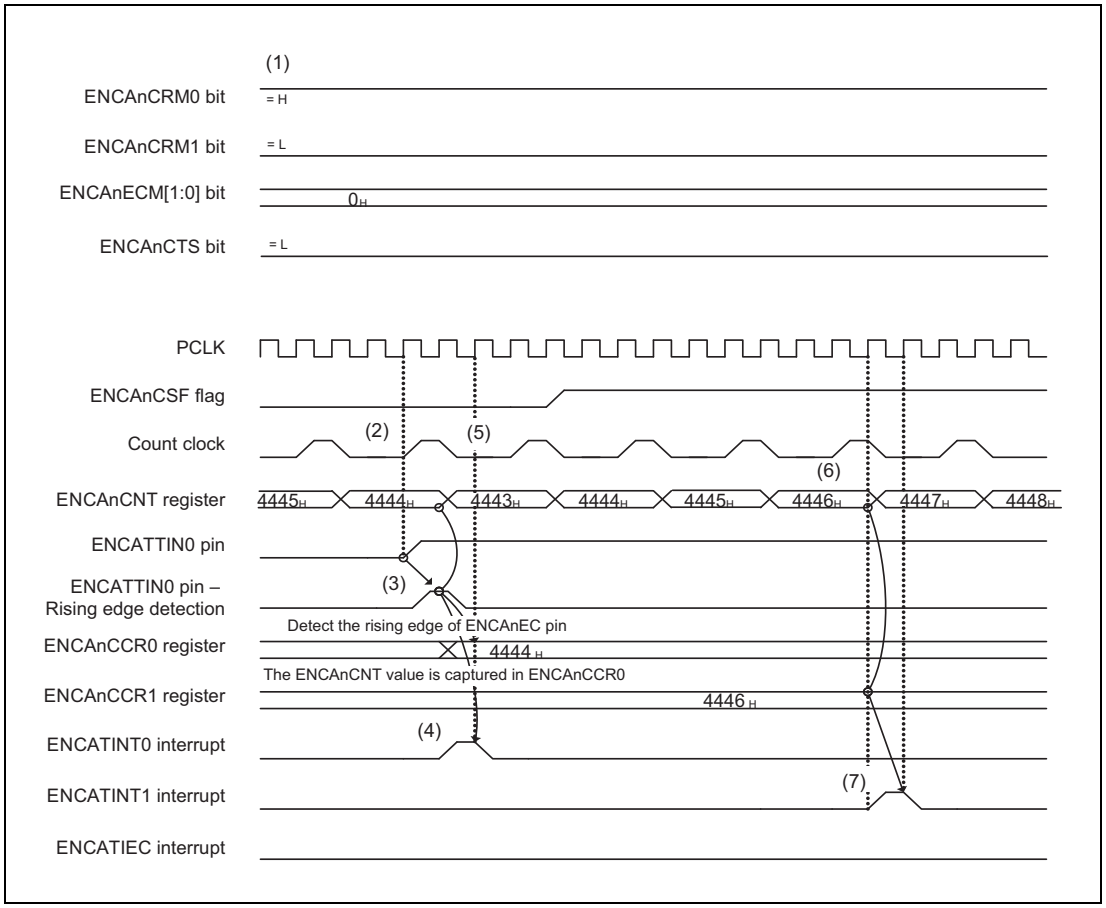


Figure 28.28 Encoder Operation when Compare Match Clear Control is Disabled

(1) The values are set as follows: ENCAnCCR1 = 4446_H, ENCAnCRM0 = 1, ENCAnCRM1 = 0, ENCAnECM[1:0] = 00_B, and ENCAnCTS = 0.

(2) A down-count is performed.

(3) When the rising edge of ENCATTIN0 is detected, the ENCAnCNT value (4444_H) is captured in ENCAnCCR0.

(4) An interrupt signal (ENCATINT0) corresponding to the capture to the ENCAnCCR0 register is output.

(5) The count operation changes to up-count.

(6) When ENCAnCNT changes to 4446_H, a compare match with ENCAnCCR1 is detected.

(7) A compare match interrupt (ENCATINT1) with ENCAnCCR1 is output.

28.6.18 Capture Operation Performed upon Clearing by ENCAnEC, ENCAnE0, or ENCAnE1 when ENCAnSCE = 1

28.6.18.1 Accompanying capture operation

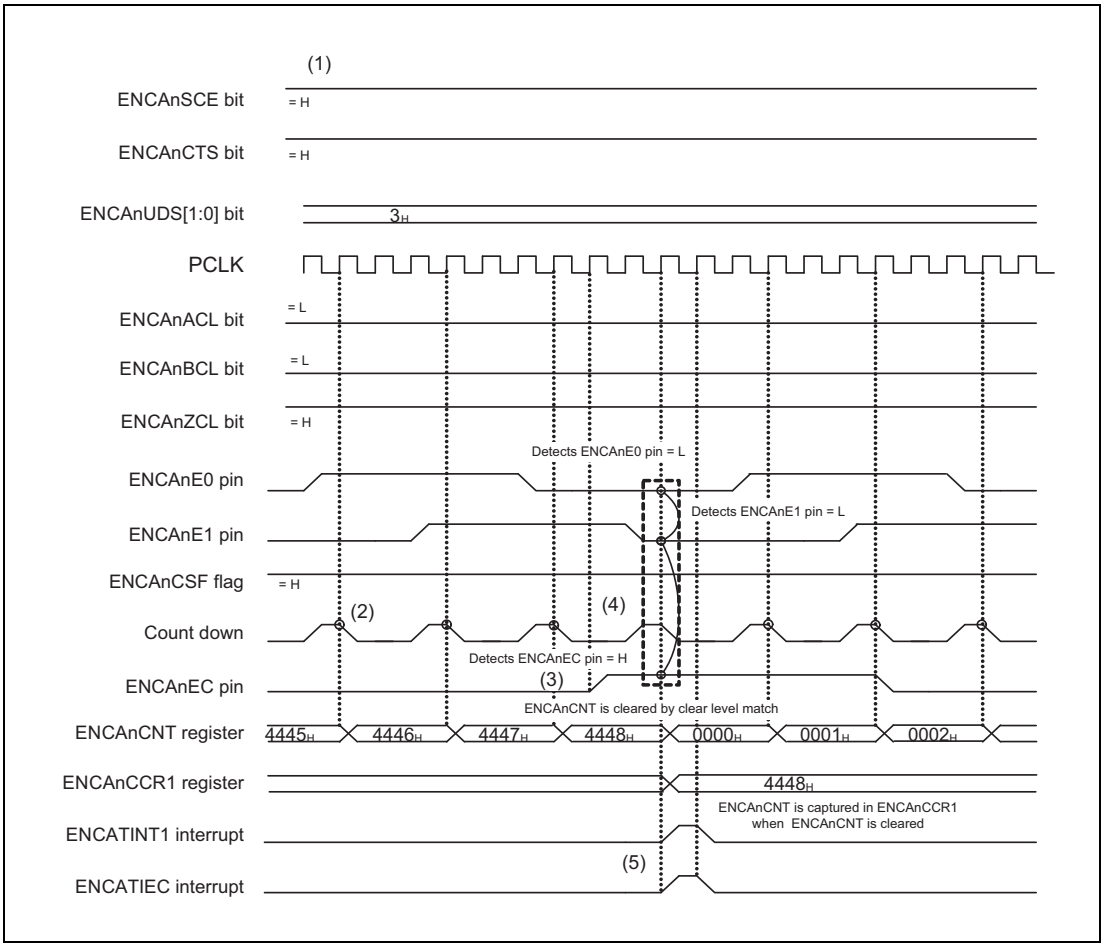


Figure 28.29 Capture Operation Performed upon Clearing by ENCAnEC, ENCAnE0, or ENCAnE1 when ENCAnSCE = 1

(1) The values are set as follows: ENCAnSCE = 1, ENCAnCTS = 1, ENCAnUDS[1:0] =11_B, ENCAnACL = 0, ENCAnBCL = 0, and ENCAnZCL = 1.

(2) An up-count is performed.

(3) The count value is not cleared upon the rising edge of ENCAnEC.

(4) When ENCAnE0, ENCAnE1 and ENCAnEC reach the set clear level, the count value is cleared. The count value is captured in ENCAnCCR1 at the time of the clearing.

(5) At the time of the clearing, an interrupt (ENCATINT1) corresponding to the capture to the ENCAnCCR1 register and a clear interrupt (ENCATIEC) by ENCAnEC are output.

28.6.18.2 When the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Up-count (When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

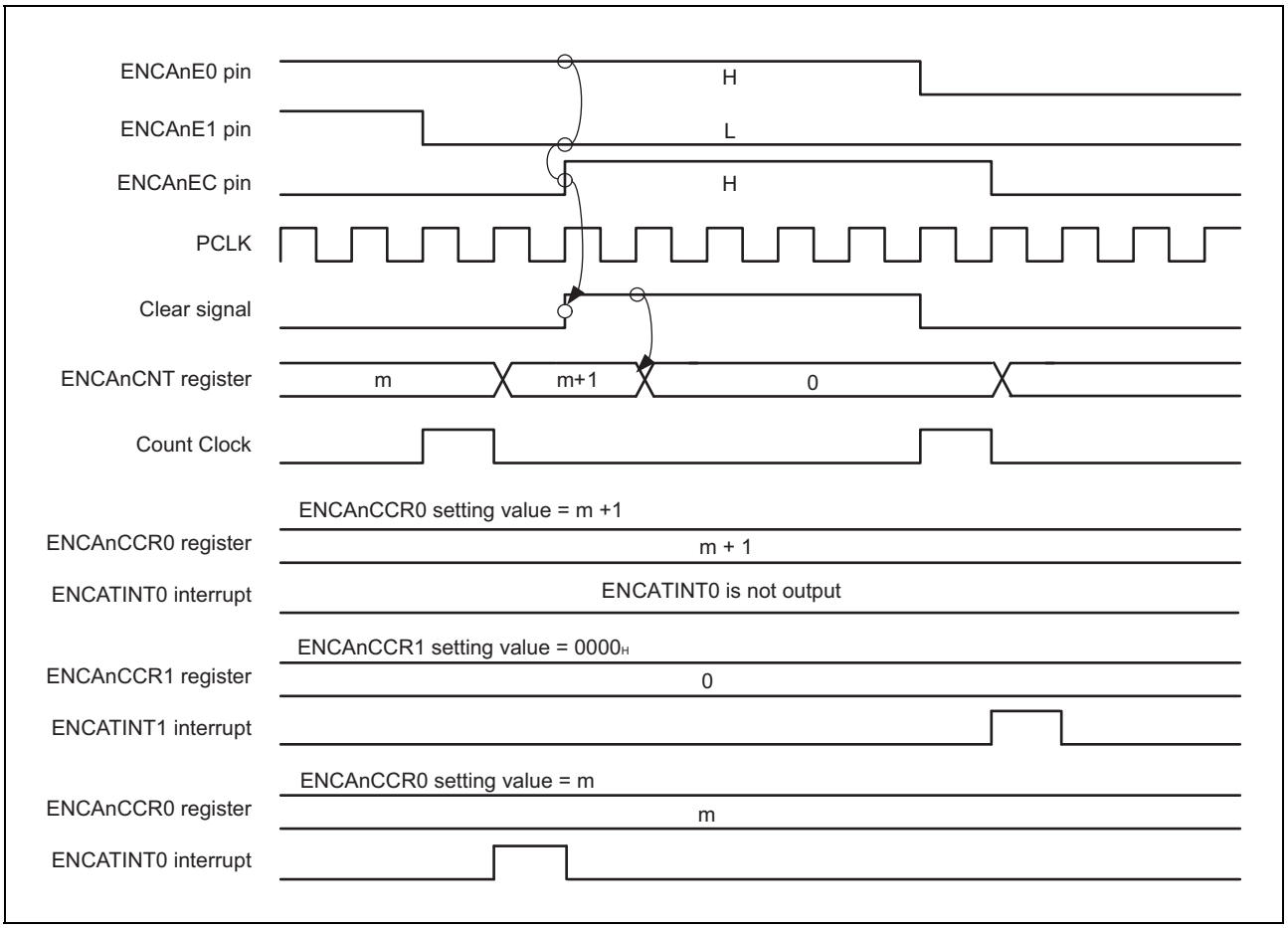


Figure 28.30 Clearing Timing for when the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Up-count

28.6.18.3 When the Timing of the ENCAnEC Input is the Same as that of the ENCAnE1 Input during Up-count (When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

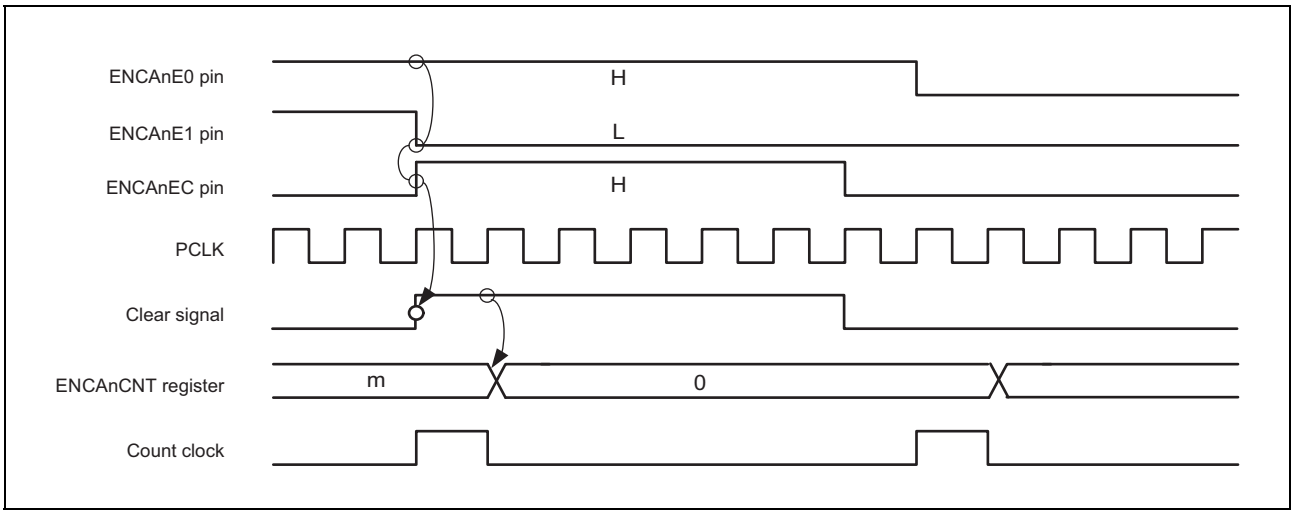


Figure 28.31 Clearing Timing for when the Timing of the ENCAnEC Input is the Same as that of the ENCAnE1 Input during Up-count

28.6.18.4 When the Timing of the ENCAnEC Input is Earlier than that of the ENCAnE1 Input during Up-count (When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

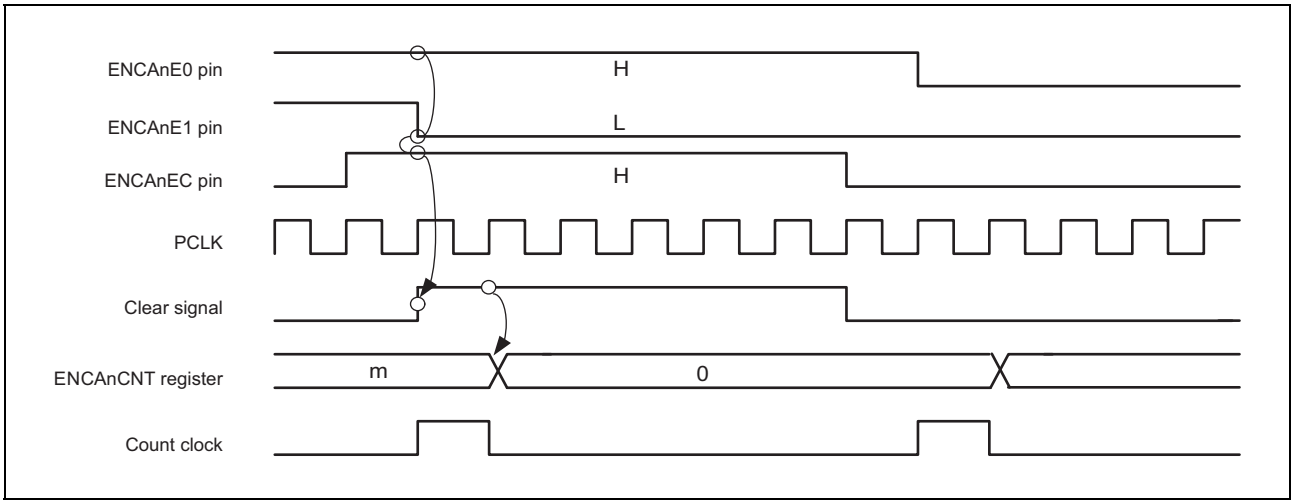


Figure 28.32 Clearing Timing for when the Timing of the ENCAnEC Input is Earlier than that of the ENCAnE1 Input during Up-count

28.6.18.5 When the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Down-count (When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

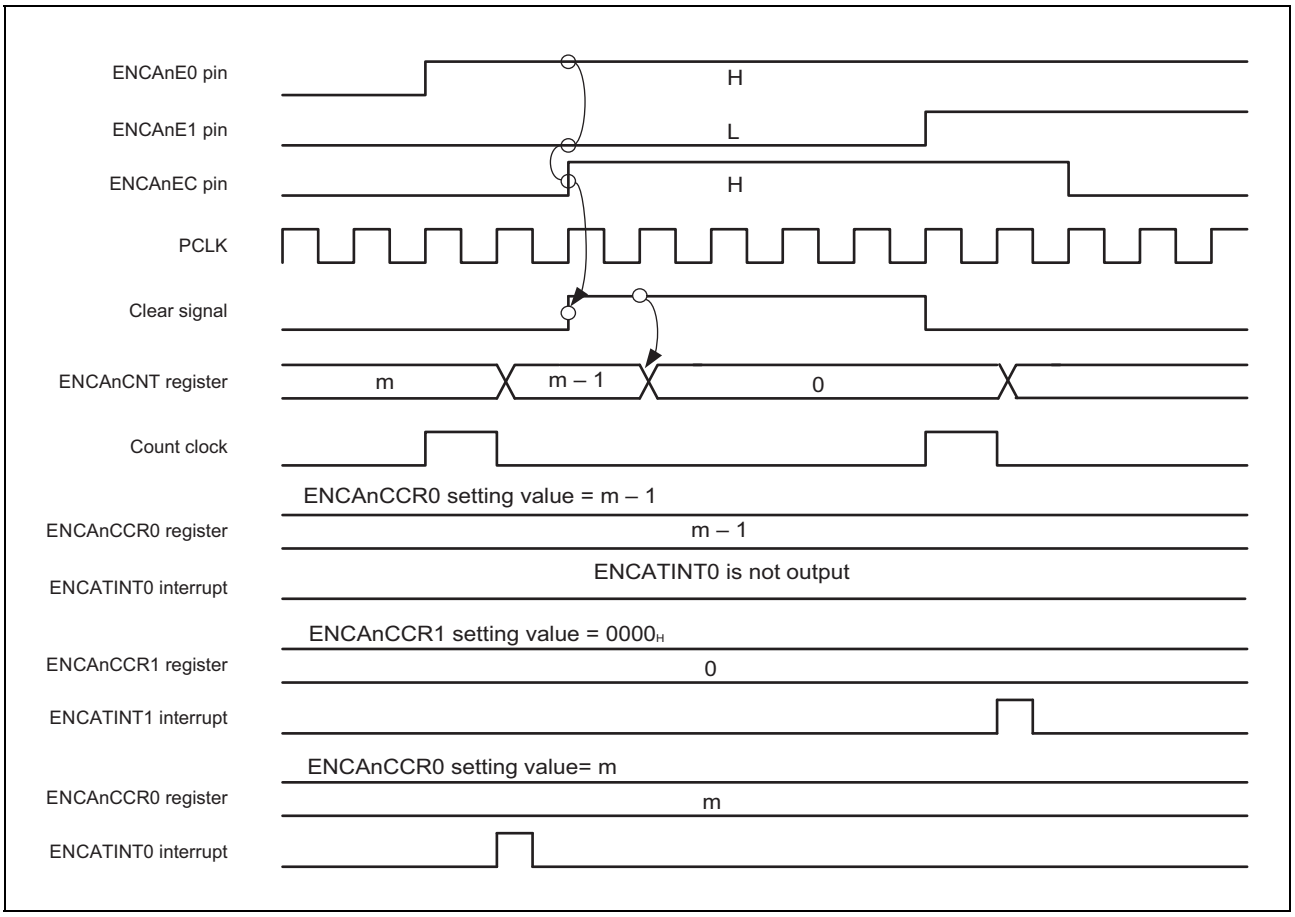


Figure 28.33 Clearing Timing for when the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Down-count

28.6.19 Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0

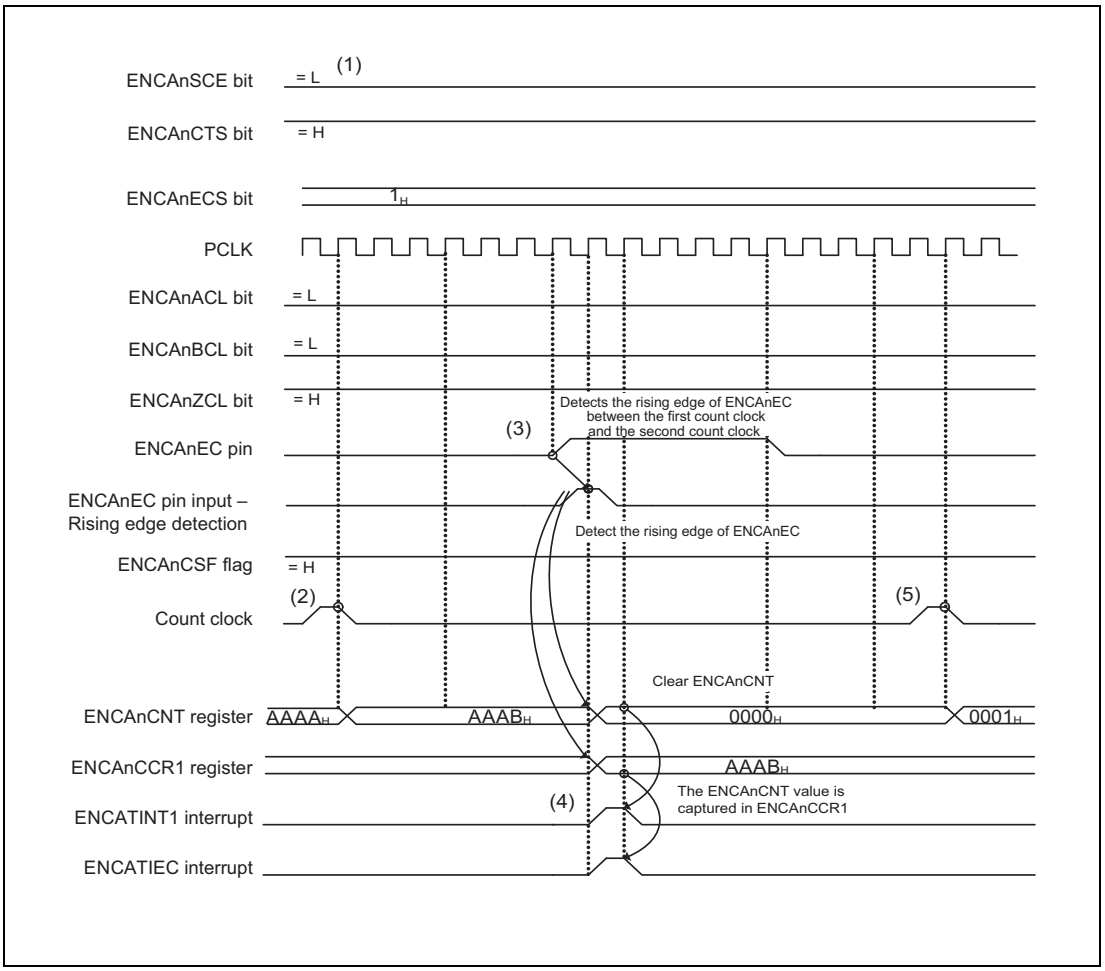


Figure 28.34 Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0

(1) The values are set as follows: ENCAnSCE = 0, ENCAnCTS = 1, and ENCAnECS[1:0] = 01_B.

(2) An up-count is performed.

(3) The rising edge of the ENCAnEC input is detected, and the ENCAnCNT value ($AAAB_H$) is captured in the ENCAnCCR1 register. Concurrently, clear operation by ENCAnEC is performed, and ENCAnCNT is cleared to 0000_H.

(4) A capture interrupt 1 (ENCATINT1) to the ENCAnCCR1 register and an encoder clear interrupt (ENCATIEC) by ENCAnEC are output.

(5) After the count value is cleared, an up-count is performed, and the count value changes to 0001_H.

Section 29 Motor Control

This section contains a generic description of the Motor Control.

The first part in this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the Motor Control.

29.1 Features of RH850/F1M Motor Control

29.1.1 Number of Units and Channels

The motor control function comprises a timer motor control unit (TAPA) and a peripheral interconnection unit (PIC) to connect the TAPA unit to peripheral timers, and generates motor control waveforms by using a combination of peripheral timers and A/D converters.

This microcontroller has the following number of TAPA and PIC units.

Table 29.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
TAPA			
Number of Units	1		
Name	TAPAn (n = 0)		
PIC			
Number of Units	1		
Name	PIC0		

Table 29.2 Indices

Index	Description
n	Throughout this section, the unit of a timer and A/D converter used by TAPA and the motor control function is identified by the index “n” (n = 0): for example, TAPAnCTL0 is the TAPAn control register 0.
m	The channel of a used timer and A/D converter is identified by the index “m”. For example, the TAUDn channel is described as CHm.
x	The scan group of an A/D converter is identified by the index “x” (x = 1 to 3).
j	The scan trigger number of an A/D converter is identified by the index “j” (j = 0 to 2).

The following table lists values indicated by the indices of each product.

Table 29.3 Indices of Products

Indices of Each Product
All Products
m = 0 to 15 (e.g. TAUDn)
x = 1 to 3
j = 0 to 2

29.1.2 Register Base Address

Base addresses of TAPAn and PIC0 are listed in the following table.

Register addresses of TAPAn and PIC0 are given with offsets from the base addresses.

Table 29.4 Register Base Addresses

Base Address Name	Base Address
<TAPA0_base>	FFE9 0000_H
<PIC0_base>	FFDD 0000_H

29.1.3 Clock Supply

The TAPAn and PIC0 clock supplies are listed in the following table.

Table 29.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
TAPAn	PCLK	CKSCLK_IPERI1
	Register access clock	CKSCLK_IPERI1
PIC0	PCLK	CKSCLK_IPERI1
	Register access clock	CKSCLK_IPERI1

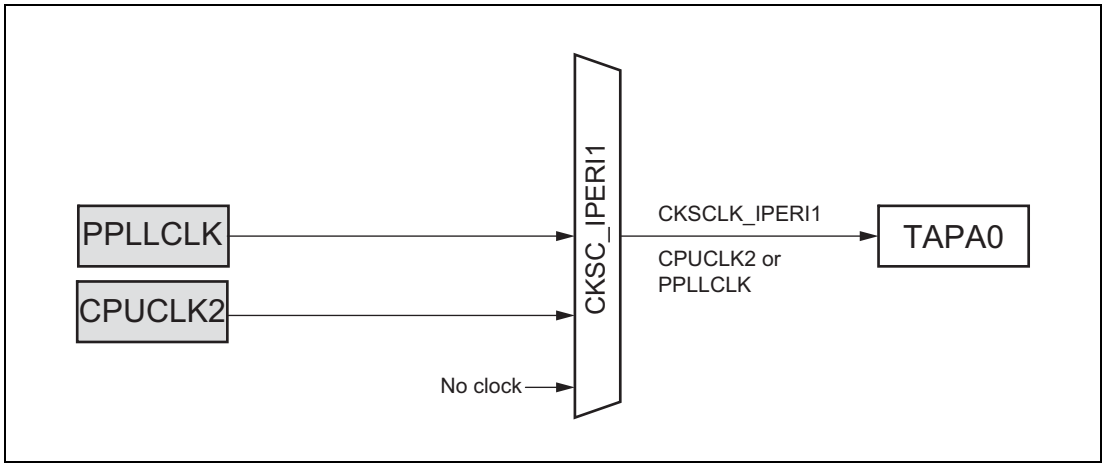


Figure 29.1 TAPA Clock Supply

29.1.4 Interrupt Request

TAPA0 interrupt requests are listed in the following table.

Table 29.6 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
TAPA0			
TAPATIPEK0	Peak interrupt 0	16, 116	—
TAPATIVLY0	Valley interrupt 0	17, 117	—

29.1.5 Reset Sources

Reset sources of TAPAn and PIC0 are listed in the following table. TAPAn and PIC0 are initialized by these reset sources.

Table 29.7 Reset Sources

Unit Name	Reset Source
TAPA0	All reset sources (ISORES)
PIC0	All reset sources (ISORES)

29.1.6 External Input/Output Signal

External output signals of TAPAn and PIC0 are listed below.

Table 29.8 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
PIC		
TOUTU	Motor control output U phase (positive)	TAPA0UP
TOUTUB	Motor control output U phase (negative)	TAPA0UN
TOUTV	Motor control output V phase (positive)	TAPA0VP
TOUTVB	Motor control output V phase (negative)	TAPA0VN
TOUTW	Motor control output W phase (positive)	TAPA0WP
TOUTWB	Motor control output W phase (negative)	TAPA0WN
TAPA		
TAPATHASIN	Motor control output Hi-Z control input	TAPA0ESO

CAUTION

For the port pins that are used as TAPA0UP, TAPA0UN, TAPA0VP, TAPA0VN, TAPA0WP, TAPA0WN, and TAPA0ESO, set the output driving ability to high (PDSCn_m = 1).

29.1.7 Internal Output Signal

Internal output signals of TAPAn and PIC0 are listed below.

Table 29.9 Internal Output Signals

Unit Signal Name	Description	Connected to
TAPA0		
$\overline{\text{TAPATHZOUT0}}$	TAPA0UP/TAPA0UN output buffer Hi-Z control output*1	Port
$\overline{\text{TAPATHZOUT1}}$	TAPA0VP/TAPA0VN output buffer Hi-Z control output*1	Port
$\overline{\text{TAPATHZOUT2}}$	TAPA0WP/TAPA0WN output buffer Hi-Z control output*1	Port
TAPATADOUT0	A/D conversion trigger signal 0 output*2	ADCA0 hardware trigger expansion
TAPATADOUT1	A/D conversion trigger signal 1 output*2	ADCA0 hardware trigger expansion
PIC0		
TAPATHASIN	TAPA0 asynchronous Hi-Z control signal*1,*3	TAPA0
TAPATSIM0	TAUD master channel interrupt signal (TAUD0: INTTAUD0I0,INTTAUD0I2, INTTAUD0I8)	TAPA0
TAPATUDCM0	TAUD master up/down signal (TAUD0: TAUD0UDC0, TAUD0UDC2, TAUD0UDC8)	TAPA0
TAPATCDENS0	TAUD slave 0 match detect*4 (ADCA0 hardware trigger expansion: ADOPA1ADCATTIN00)	TAPA0
TAPATCDENS1	TAUD slave 1 match detect*4 (ADCA0 hardware trigger expansion: ADOPA2ADCATTIN00)	TAPA0

Note 1. See **Section 29.4.6, TAPA0 Hi-Z Control Input Selection** for details.

Note 2. These signals can be used to as a trigger source to start the A/D converter. See **Table 31.47, List of A/D Conversion Hardware Triggers**.

Note 3. This input signal is passed through a noise filter. See **Section 2.12, Noise Filter & Edge/Level Detector** and **Section 2.13, Description of Port Noise Filter & Edge/Level Detection**.

Note 4. These signals are selected by the H/W trigger selection bit of the ADCA0 A/D converter. See **Table 31.47, List of A/D Conversion Hardware Triggers**.

29.2 Overview

29.2.1 Functional Overview

The motor control function provides the following functions by combining the motor control unit (TAPA) and Timer Array Unit D (TAUDn) or A/D (ADCAn):

- Asynchronous Hi-Z control function
  Hi-Z control for TAUDn output can be performed by using pin input or error signals.
- Interrupt signal output function
  Request signals for two types of interrupts, peak interrupts and valley interrupts, can be output by the INTn signals output by TAUDn.
- A/D conversion start trigger selection function
  An A/D conversion start trigger can be output by INTn signals output by TAUDn.

Additionally, the motor control function provides also the following functions by combining the motor control unit (TAPA) and the peripheral interconnection (PIC):

- Timer simultaneous start trigger function
  The respective channels of TAUD0 and TAUJ1, and the ENCAn timer can be started simultaneously.
- Trigger and pulse width measuring function
  Measurement of trigger periods can be performed by inputting ENCAn interrupt signals to TAUDn or TAUJ1.
- A/D trigger encoder capture function
  The value of the ENCAn counter can be captured at the A/D conversion start trigger timing.
- Three-phase PWM output with dead time / High-accuracy triangle PWM output with dead time
  Three-phase PWM output with dead time can be performed by TAUDn.
- Delay pulse output with dead time
  Delay pulses (with dead time) for the cycle timing can be output.

29.2.2 Basic Structure of Motor Control

The peripheral block configuration of the motor control function is shown below.

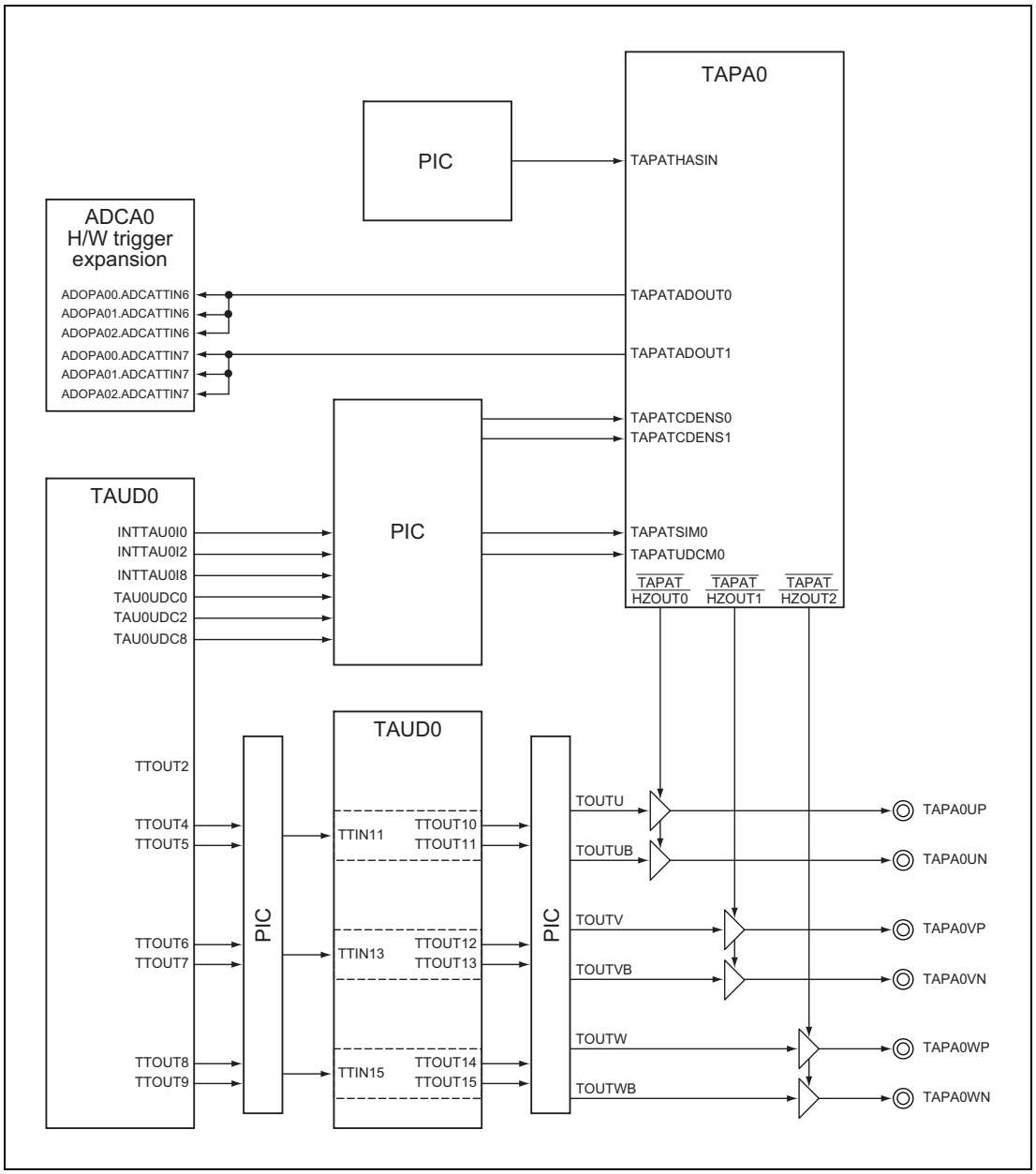


Figure 29.2 Configuration of Motor Control

TAUDn and PIC are used to generate the motor control output signals (three-phase PWM output signals with dead time).

The timer control unit (TAPA) performs Hi-Z control for the motor control output.

Additionally, the PIC can provide functions specific to the motor by combining respective channels of TAUDn and TAUJ1, ENCAn, and TAPA.

29.2.3 Block Diagram

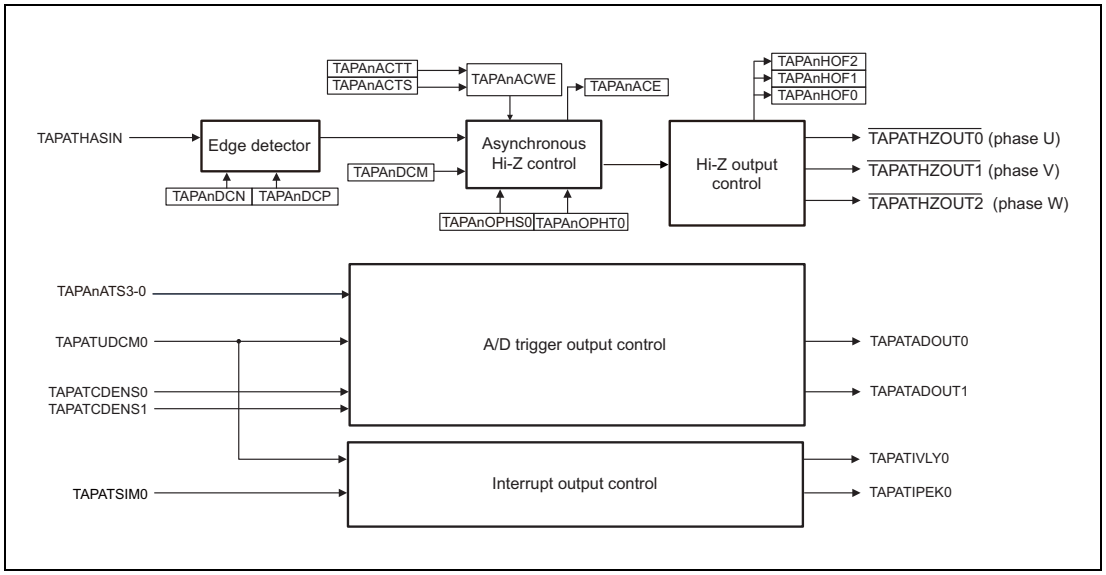


Figure 29.3 TAPA Peripheral Block Diagram

NOTE

For the PIC peripheral block diagram, see the respective section describing each function.

29.2.4 Definition of Terms

Peak and valley interrupts - Peak and valley of timer counter

In this document, the period from a TAUD counting-up status to generation of INT from the master channel is defined as a peak period, and this INT is defined as a peak interrupt.

In contrast, the period from a TAUD counting-down status to generation of INT from the master channel is defined as a valley period, and this INT is defined as a valley interrupt.

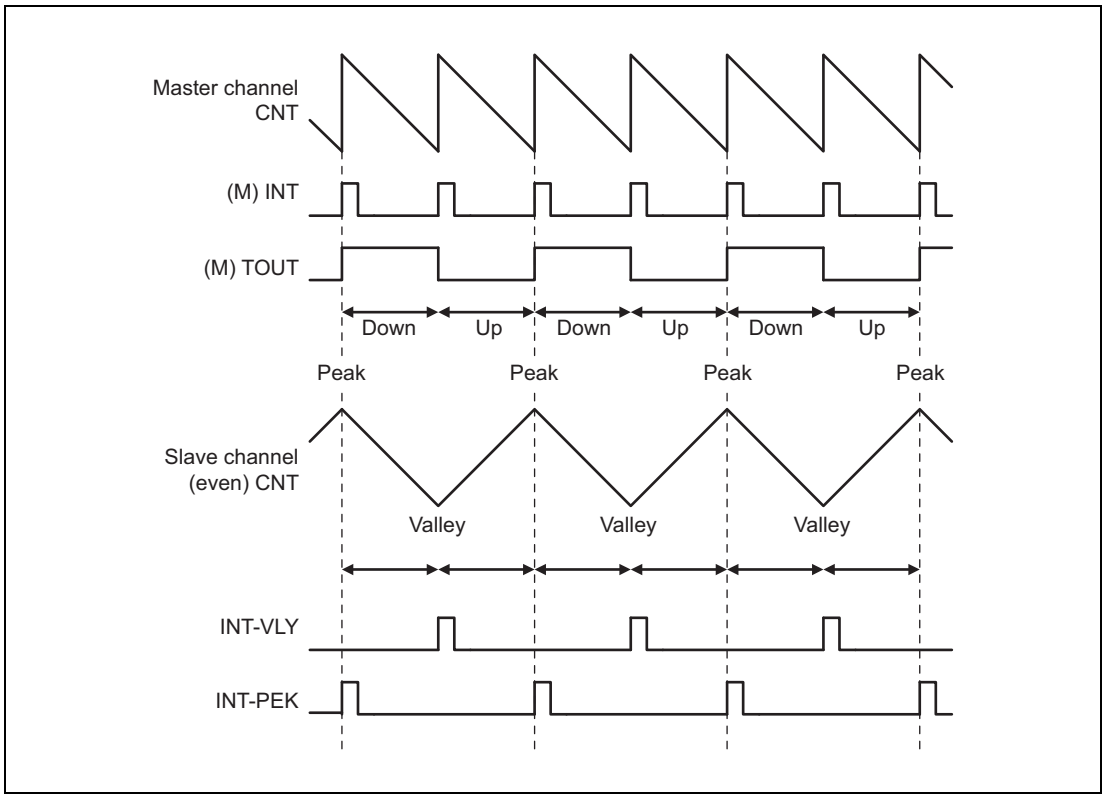


Figure 29.4 Peak and Valley Interrupts

29.3 Registers

29.3.1 List of Registers

Registers of TAPAn and PIC0 are listed in the following table.

For details about <TAPAn_base> and <PIC0_base>, see **Section 29.1.2, Register Base Address**.

Table 29.10 Registers

Module Name	Register Name	Symbol	Address
TAPAn	Control register 0	TAPAnCTL0	<TAPAn_base> + 20_H
TAPAn	Control register 1	TAPAnCTL1	<TAPAn_base> + 24_H
TAPAn	Flag register	TAPAnFLG	<TAPAn_base> + 00_H
TAPAn	Asynchronous Hi-Z control write enable register	TAPAnACWE	<TAPAn_base> + 04_H
TAPAn	Asynchronous Hi-Z control start trigger register	TAPAnACTS	<TAPAn_base> + 08_H
TAPAn	Asynchronous Hi-Z control stop trigger register	TAPAnACTT	<TAPAn_base> + $0C_H$
TAPAn	Hi-Z start trigger register	TAPAnOPHS	<TAPAn_base> + 14_H
TAPAn	Hi-Z stop trigger register	TAPAnOPHT	<TAPAn_base> + 18_H
TAPAn	Emulation register	TAPAnEMU	<TAPAn_base> + 28_H
PIC0	Simultaneous start trigger control register	PIC0SST	<PIC0_base> + 04_H
PIC0	Simultaneous start control register 0	PIC0SSER0	<PIC0_base> + 10_H
PIC0	Simultaneous start control register 2	PIC0SSER2	<PIC0_base> + 18_H
PIC0	Hi-Z output control register 0	PIC0HIZCEN0	<PIC0_base> + 80_H
PIC0	A/D conversion trigger output control register 400	PIC0ADTEN400	<PIC0_base> + 90_H
PIC0	A/D conversion trigger output control register 401	PIC0ADTEN401	<PIC0_base> + 94_H
PIC0	A/D conversion trigger output control register 402	PIC0ADTEN402	<PIC0_base> + 98_H
PIC0	Timer I/O control register 200	PIC0REG200	<PIC0_base> + $C0_H$
PIC0	Timer I/O control register 201	PIC0REG201	<PIC0_base> + $C4_H$
PIC0	Timer I/O control register 202	PIC0REG202	<PIC0_base> + $C8_H$
PIC0	Timer I/O control register 203	PIC0REG203	<PIC0_base> + CC_H
PIC0	Timer I/O control register 30	PIC0REG30	<PIC0_base> + $E8_H$
PIC0	Timer I/O control register 31	PIC0REG31	<PIC0_base> + EC_H

NOTE

For details about PIC-related registers, see the respective section describing each function.

29.3.2 TAPAnCTL0 — TAPA Control Register 0

This register is used to set up the asynchronous Hi-Z control function.

The values of this register can be rewritten only when TAPAnFLG.TAPAnACE is 0 and TAUDnTEm for the corresponding TAUD's master channel is 0.

Access: This register can be read or written in 16-bit units.

Address: <TAPAn_base> + 20_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TAPAn DCM	TAPAn DCN	TAPAn DCP	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 29.11 TAPAnCTL0 Register Contents

Bit Position	Bit Name	Function
15 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	TAPAnDCM	Clear Condition Configuration This control bit specifies the clear conditions for Hi-Z control output. 0: Enables manipulation of TAPAnOPHT0 regardless of the TAPATHASIN signal input level. 1: Enables manipulation of TAPAnOPHT0 only if the TAPATHASIN signal input is inactive.
3, 2	TAPAnDCN, TAPAnDCP	Hi-Z Input Edge Selection These are control bits that specify the valid edge of TAPATHASIN.

TAPAn DCN	TAPAn DCP	Description
0	0	Does not detect valid edges.
0	1	Detects a rising edge as the valid edge (active level = high).
1	0	Detects a falling edge as the valid edge (active level = low).
1	1	Setting prohibited

1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

29.3.3 TAPAnCTL1 — TAPA Control Register 1

This register is used to specify the A/D conversion trigger.

Access: This register can be read or written in 8-bit units.

Address: <TAPAn_base> + 24_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAPAnATS3	TAPAnATS2	TAPAnATS1	TAPAnATS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 29.12 TAPAnCTL1 Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3, 2	TAPAnATS3, TAPAnATS2	A/D Conversion Trigger 1 Selection These are control bits that specify the A/D conversion trigger output 1 (TAPATADOUT1).

TAPAn ATS3	TAPAn ATS2	Description
0	0	INT signal while the triangle wave is falling (counting down)
0	1	Setting prohibited
1	0	INT signal while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt TAPATIVLY0 signal

Bit Position	Bit Name	Function
1, 0	TAPAnATS1, TAPAnATS0	A/D Conversion Trigger 0 Selection These are control bits that specify the A/D conversion trigger output 0 (TAPATADOUT0).

TAPAn ATS1	TAPAn ATS0	Description
0	0	INT signal while the triangle wave is falling (counting down)
0	1	Setting prohibited
1	0	INT signal while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt TAPATIVLY0 signal

29.3.4 TAPAnFLG — TAPA Flag Register

This flag register is for asynchronous Hi-Z control.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <TAPAn_base> + 00_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TAPAn HOF2	TAPAn HOF1	TAPAn HOF0	—	—	—	—	—	—	—	TAPAn ACE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.13 TAPAnFLG Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned.
10 to 8	TAPAnHOF2	W phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of TAPAnTHZOUT2 is high level 1: The present output TAPAnTHZOUT2 is low level.
9	TAPAnHOF1	V phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of TAPAnTHZOUT1 is high level 1: The present output TAPAnTHZOUT1 is low level.
8	TAPAnHOF0	U phase Hi-Z Control Monitor This bit is used to monitor the Hi-Z control status. 0: The present output of TAPAnTHZOUT0 is high level 1: The present output TAPAnTHZOUT0 is low level.
7 to 1	Reserved	When read, the value after reset is returned.
0	TAPAnACE	Asynchronous Hi-Z Control Enable 0: Indicates that the asynchronous Hi-Z control is stopped. 1: Indicates that the asynchronous Hi-Z control is enabled. The conditions for setting or clearing this bit are as follows: Clear condition: Writing 1 to TAPAnACTT while TAPAnACWE = 1 Set condition: Writing 1 to TAPAnACTS while TAPAnACWE = 1

29.3.5 TAPAnACWE — TAPA Asynchronous Hi-Z Control Write Enable Register

This register is used to enable writing for asynchronous Hi-Z control.

Access: This register can be read or written in 8-bit units.

Address: <TAPAn_base> + 04_H

Value after reset: 00_H

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

Table 29.14 TAPAnACWE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TAPAnACWE	Asynchronous Control Write Enable This is a write enable bit for asynchronous Hi-Z control. After 1 is written, this bit is automatically cleared to 0 by writing 1 to TAPAnACTS and TAPAnACTT. 0: Disables writing to TAPAnACTS and TAPAnACTT. 1: Enables writing to TAPAnACTS and TAPAnACTT.

29.3.6 TAPAnACTS — TAPA Asynchronous Hi-Z Control Start Trigger Register

This register is used to enable the start trigger for asynchronous Hi-Z control.

Access: This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address: <TAPAn_base> + 08_H

Value after reset: 00_H

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

Table 29.15 TAPAnACTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnACTS	Asynchronous Hi-Z Control Start Trigger This bit enables the start trigger for asynchronous Hi-Z control. The setting of this bit is valid only when TAPAnACWE = 1. 0: Writing 0 is ignored (no function). 1: Enables asynchronous Hi-Z control when TAPAnACE is 1.

29.3.7 TAPAnACTT — TAPA Asynchronous Hi-Z Control Stop Trigger Register

This bit enables the stop trigger for asynchronous Hi-Z control.

Access: This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address: <TAPAn_base> + $0C_H$

Value after reset: 00_H

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

Table 29.16 TAPAnACTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnACTT	Asynchronous Hi-Z Control Stop Trigger This bit enables the stop trigger for asynchronous Hi-Z control. The setting of this bit is valid only when TAPAnACWE = 1. 0: Writing 0 is ignored (no function). 1: Disables asynchronous Hi-Z control when TAPAnACE is 0.

29.3.8 TAPAnOPHS — TAPA Hi-Z Start Trigger Register

This software trigger register is used to start Hi-Z control for motor control output pins.

Access: This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address: <TAPAn_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnOPHS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 29.17 TAPAnOPHS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnOPHS0	Hi-Z Control Start Trigger This bit starts Hi-Z control for motor control output pins. 0: Writing 0 is ignored (no function). 1: Starts Hi-Z control.

29.3.9 TAPAnOPHT — TAPA Hi-Z Stop Trigger Register

This software trigger register is used to stop Hi-Z control for motor control output pins.

Access: This register is a write-only register that can be written in 8-bit units. This register is always read as 00_H.

Address: <TAPAn_base> + 18_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAPAnOPHT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 29.18 TAPAnOPHT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	TAPAnOPHT0	Hi-Z Control Stop Trigger This bit stops Hi-Z control for motor control output pins. 0: Writing 0 is ignored (no function). 1: Stops Hi-Z control. Whether the setting of this bit is valid or invalid depends on the setting of TAPAnCTL0.TAPAnDCM.

29.3.10 TAPAnEMU — TAPA Emulation Register

This register controls SVSTOP for emulation.

Access: This register can be read or written in 8-bit units. (when SVSTOP = low, rewritten only)

Address: <TAPAn_base> + 28_H

Value after reset: Reading this register returns always 00_H.

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

Table 29.19 TAPAnEMU Register Contents

Bit Position	Bit Name	Function
7	TAPAnSVSDIS	This bit is used to control disabling of SVSTOP. 0: SVSTOP is valid. (Sets Hi-Z control output to low level when SVSTOP = H is input). 1: SVSTOP is invalid. (Hi-Z control output level does not change according to the level of SVSTOP input).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

29.4 Asynchronous Hi-Z Control Function

If the operation of the timer motor control function controlled by the MCU becomes abnormal, the rotation of the external motor also becomes abnormal. This function can forcibly set the motor control output to the Hi-Z state upon detection of abnormal motor operation, independent of the MCU control.

29.4.1 Overview

This function forcibly stops TAPAn output through asynchronous Hi-Z control.

- When the TAPATHASIN signal becomes active, the levels of the motor control output pins are set to Hi-Z, and motor control output is forcibly stopped.
- Motor control output in a Hi-Z state can be resumed by writing the Hi-Z stop trigger register (TAPAnOPHT0).
- The Hi-Z state of motor control output can also be specified by writing the Hi-Z control start trigger register (TAPAnOPHS).
- Setting PIC can enable or disable Hi-Z control input when an error occurs.

29.4.2 System Configuration Example

A system configuration example is shown below, where an external error detection signal (the TAPA0ESO signal) is used for the Hi-Z control of the motor control outputs (TAPA0UP / TAPA0UN / TAPA0VP / TAPA0VN / TAPA0WP / TAPA0WN).

When valid edges of the external error detection signal are detected, the level of the motor control outputs is set to Hi-Z.

Because the microcontroller might freeze when an error occurs, external error detection signals are continuously processed so that the motor control timer outputs can be set to Hi-Z even if no clock is supplied.

Note that an error is detected only when the valid edge of the error detection signal is detected. Therefore, no error is detected if the output level is fixed and the signal level does not change.

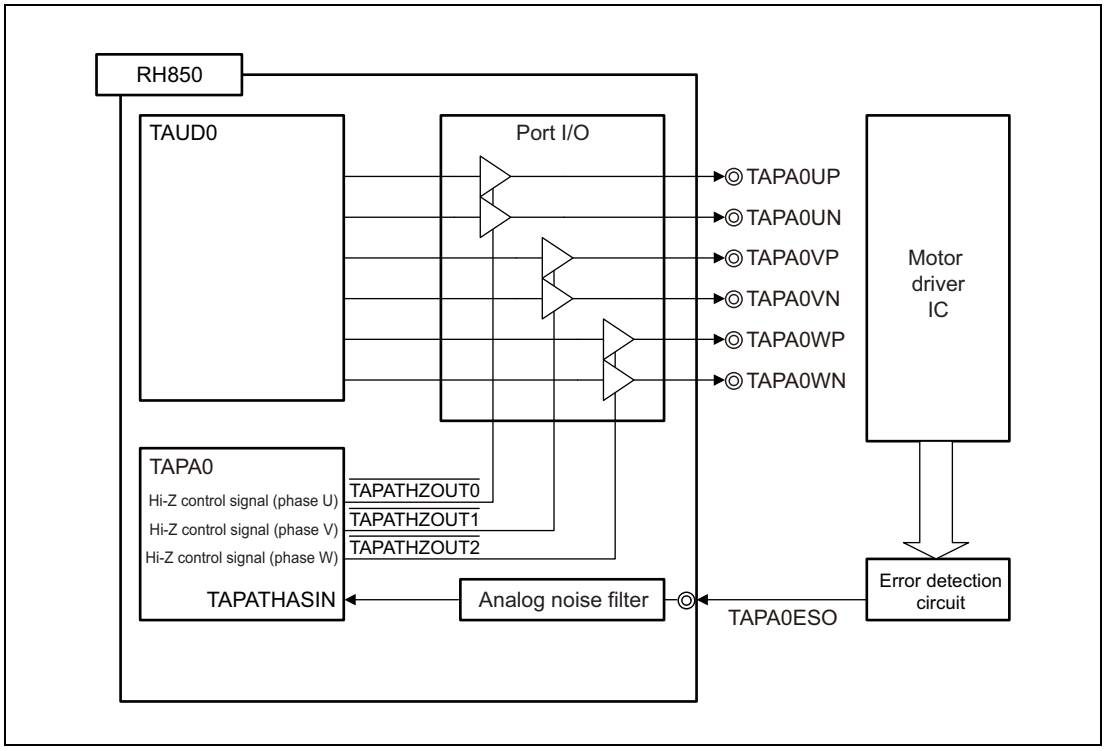


Figure 29.5 System Configuration Example of Asynchronous Hi-Z Control for Pin Input

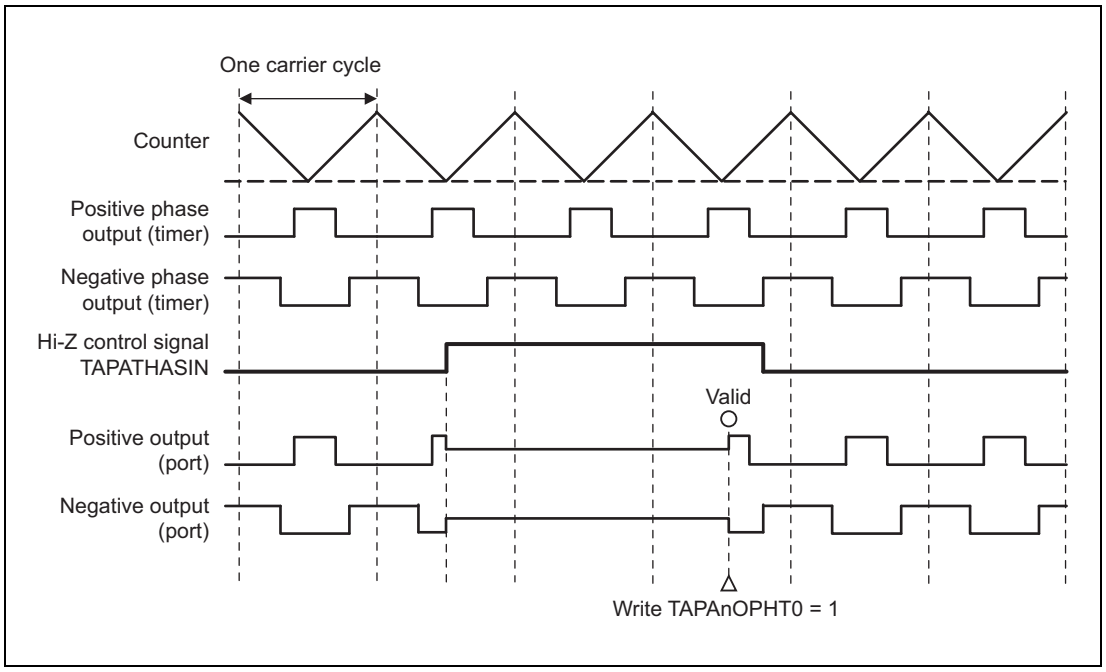
29.4.3 Basic Operation

Hi-Z control for motor control output pins can be started as follows:

- Detecting the valid edge of asynchronous Hi-Z control signal (TAPATHASIN)
- Setting the start trigger bit TAPAnOPHS.TAPAnOPHS0 of the Hi-Z control signal

The levels of the motor control output pins are set to Hi-Z until the stop trigger bit of the Hi-Z control signal (TAPAnOPHT.TAPAnOPHT0) is set. Note that whether the setting of TAPAnOPTH0 is valid or invalid depends on the setting of TAPAnCTL0.TAPAnDCM.

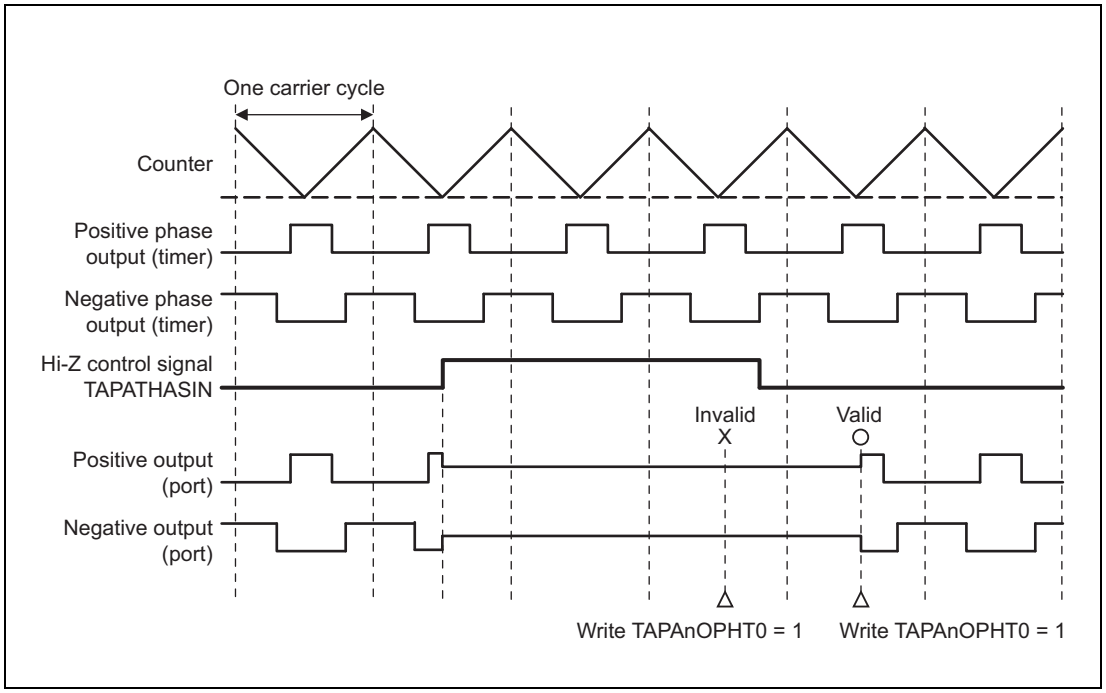
(1) Operation when TAPAnCTL0.TAPAnDCM = 0, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control outputs are forcibly stopped (Hi-Z output) when the valid edge of TAPATHASIN is detected.

The motor control outputs restart when 1 is written to TAPAnOPHT.TAPAnOPHT0, regardless of the level of TAPATHASIN.

(2) Operation when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control outputs are forcibly stopped (Hi-Z output) when the valid edge of TAPATHASIN is detected.

Writing 1 to the stop trigger bit (TAPAnOPHT.TAPAnOPHT0) of the Hi-Z control signal is ignored while TAPATHASIN is active (high level because TAPAnCTL0.TAPAnDCP is 1).

The motor control outputs restart when 1 is written to TAPAnOPHT.TAPAnOPHT0 after TAPATHASIN becomes inactive (low level because TAPAnCTL0.TAPAnDCP is 1).

29.4.4 Asynchronous Hi-Z Control Using Software Trigger

Hi-Z control for motor control output is possible by using the Hi-Z control start trigger bit TAPAnOPHS.TAPAnOPHS0 and Hi-Z control stop trigger bit TAPAnOPHT.TAPAnOPHT0.

(1) Function of Hi-Z control start trigger bit TAPAnOPHS.TAPAnOPHS0

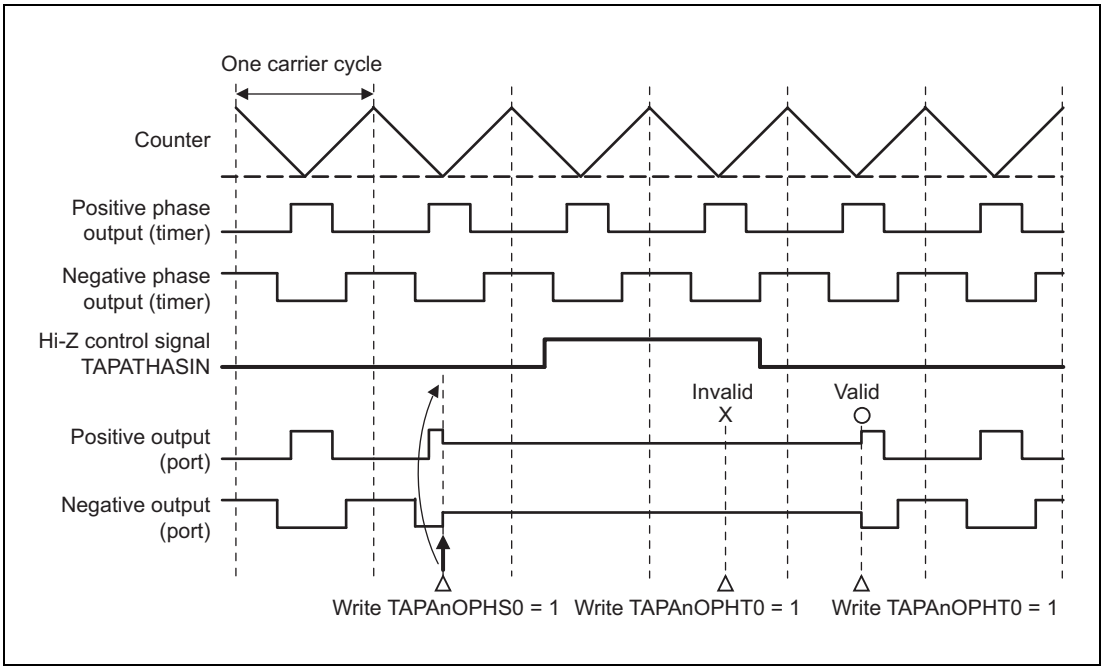
TAPAnDCM	Function
0/1	Writing 1 to TAPAnOPHS0 starts Hi-Z control and forcibly stops the motor control output (Hi-Z output).

(2) Function of Hi-Z control stop trigger bit TAPAnOPHT.TAPAnOPHT0

Whether the Hi-Z control stop trigger is enabled or disabled depends on the conditions below:

TAPAnDCM	Function
0	Writing 1 to TAPAnOPHT0 stops Hi-Z control and restarts motor control output.
1	If TAPATHASIN is inactive, writing 1 to TAPAnOPHT0 stops Hi-Z control and restarts motor control output. If TAPATHASIN is active, writing 1 to TAPAnOPHT0 is ignored.

(3) Operation when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control output (Hi-Z output) is forcibly stopped when 1 is written to TAPAnOPHS0.

After that, the levels of the motor control outputs remain Hi-Z even if the rising edge of TAPATHASIN is detected.

Writing to TAPAnOPHT0 is ignored while TAPATHASIN is active (high level because TAPAnDCN is 0 and TAPAnDCP is 1).

After detection of the falling edge of TAPATHASIN, the motor control outputs restart when 1 is written to TAPAnOPHT0 while TAPATHASIN is inactive (low level because TAPAnDCN is 0 and TAPAnDCP is 1).

29.4.5 Operating Procedure

The operating procedure for the asynchronous input Hi-Z control function is shown below:

	Operation	Status of TAPA
Initial setup	Set up the TAPAnCTL0 register. Specify TAPAnDCP and TAPAnDCN to select the input edge. Specify TAPAnDCM to select the clear mode.	Asynchronous Hi-Z control stopped (TAPAnFLG.TAPAnACE = 0)
Start operation	Set up the TAPAnACWE register. Set TAPAnACWE to 1. Set up the TAPAnACTS register. Set TAPAnACTS to 1.	 Writing to TAPAnACTS is enabled. Asynchronous Hi-Z control enabled (TAPAnFLG.TAPAnACE = 1)
During operation	Hi-Z control for the timer function outputs can be started by controlling the following: • TAPAnOPHS register • Asynchronous Hi-Z control signal (TAPATHASIN) Hi-Z control for the timer function outputs can be stopped by controlling the following: • TAPAnOPHT register (If TAPAnDCM is 1, control by the TAPAnOPHT register is enabled only while TAPATHASIN is inactive.) The TAPA operating status can always be read using the TAPAnFLG register.	Hi-Z control for the motor control output pins is started by detecting the valid edge of the asynchronous Hi-Z control signal (TAPATHASIN) or by setting the Hi-Z control start trigger bit TAPAnOPHS0 to 1. Hi-Z control for the motor control output pins is stopped by setting the Hi-Z control stop trigger bit TAPAnOPHT0 to 1 according to the operation mode specified by the TAPAnDCM bit.
Stop operation	Set up the TAPAnACWE register. Set TAPAnACWE to 1. Set up the TAPAnACTT register. Set TAPAnACTT to 1.	 Writing to TAPAnACTT is enabled. Asynchronous Hi-Z control stopped (TAPAnFLG.TAPAnACE = 0)

Restart

29.4.6 TAPA0 Hi-Z Control Input Selection

In order to stop the motor control outputs in case of errors, error events are selected in PIC and the levels of the motor control outputs are set to Hi-Z in TAPA0, as shown in the diagram below.

The TAPA function can be stopped by setting TAPA0ACTT = 01_H after setting PIC0HIZCEN0 = 00_H or TAPA0ACWE = 01_H.

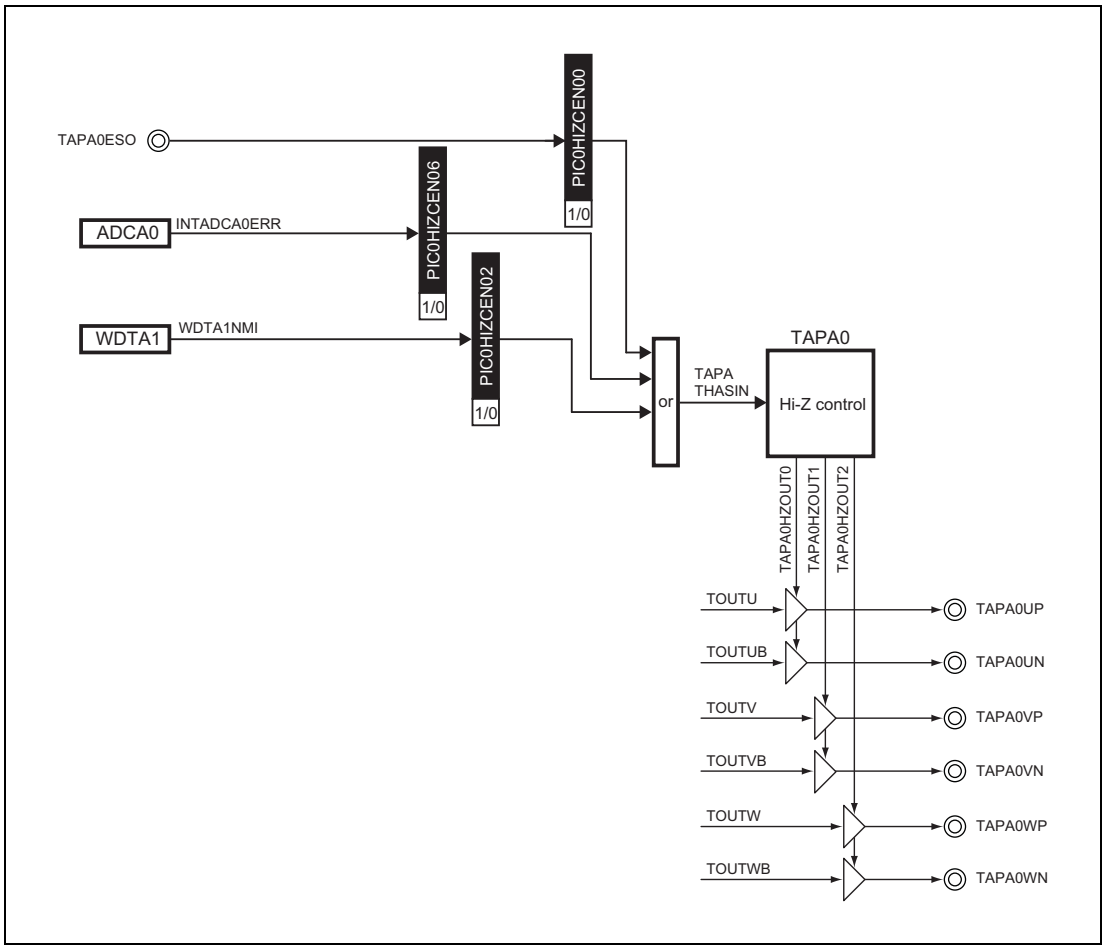


Figure 29.6 Hi-Z Control Block Diagram

Switching into the Hi-Z state can be performed by the following:

- TAPA0ESO pin input
- A/D converter ADCA0 error signal ADCA0ERR
- Window Watchdog Timer WDTA1 non maskable interrupt WDTA1NMI

For details about these signals, see their respective descriptions.

29.4.7 Registers

29.4.7.1 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

The PIC0HIZCENn register selects the Hi-Z output control signal of TAPAn.

Access: This register can be read or written in 8-bit units.

Address: <PIC0_base> + 80_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R	R/W

Table 29.20 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Hi-Z Output Control by INTADCA0ERR Interrupt Signal Enable 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	Selects whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Hi-Z Output Control by the TAPA0ESO Pin Input Enable 0: Disable 1: Enable

29.5 INT Signal Output Selection Function

29.5.1 Configuration of the INT Signal Output Selection Function

This function generates the peak interrupt TAPATIPEK0 and valley interrupt TAPATIVLY0 by using the TAPATSIM0 signal, which is connected to the INT signal on the TAUD's triangular carrier cycle generation channel (master) and TAPATUDCM0 signal, which is connected to the counter up/down signal.

For the connection destination of TAPATSIM0, see **Section 29.1.7, Internal Output Signal**.

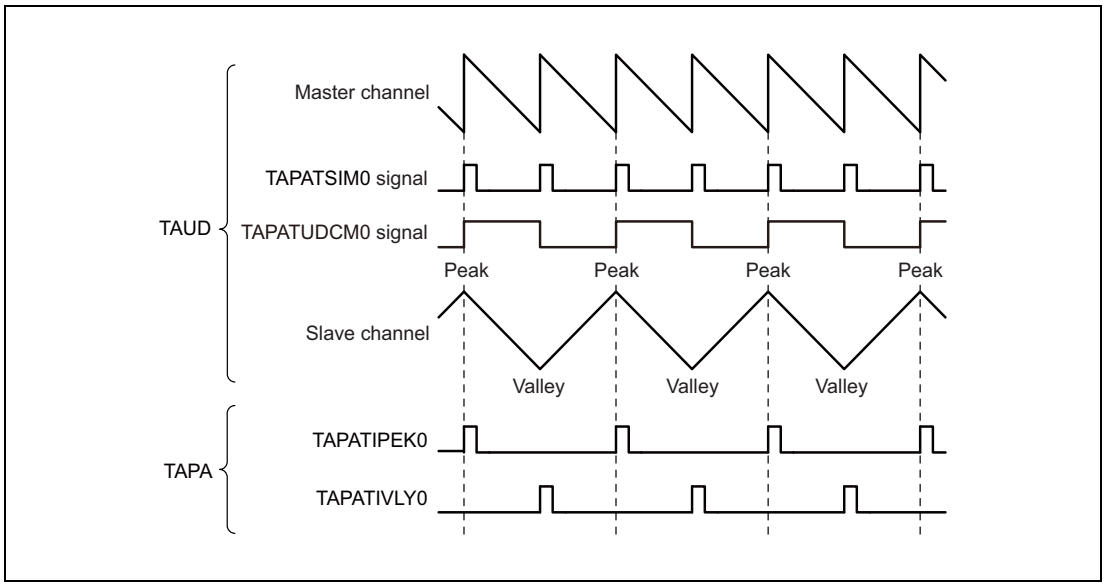


Figure 29.7 Basic Timing of Signals for the INT Signal Output Selection Function

Triangular carrier cycles are generated on the master channel.

The INT signal generated on the master channel in each half triangular carrier cycle is input to TAPAn as TAPATSIM0 signal. TAPAn generates TAPATIPEK0 signal (peak interrupt) during high level of the TAPATUDCM0 signal and TAPATIVLY0 signal (valley interrupt) during low level of the TAPATUDCM0 signal by using TAPATSIM0 and TAPATUDCM0 input signal.

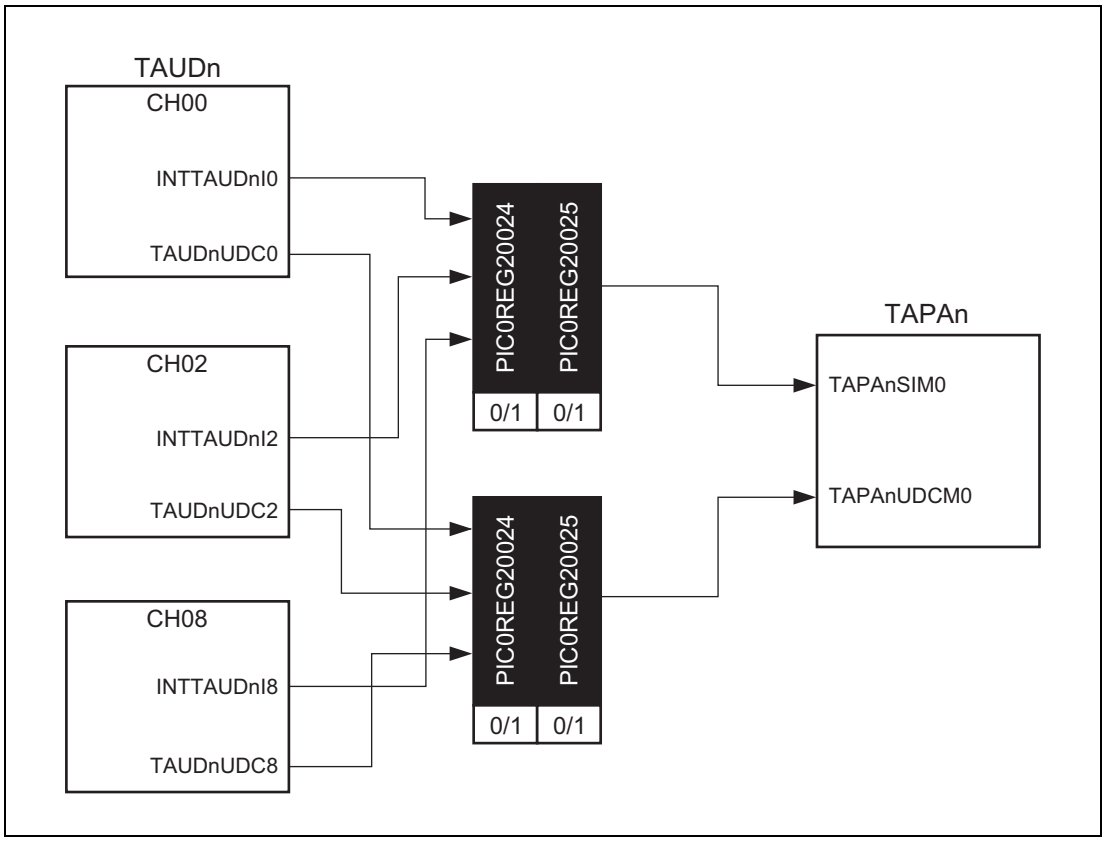
CAUTION

The peak interrupt TAPATIPEK0 and valley interrupt TAPATIVLY0 are generated regardless of the function of the master channel of the TAUD.

When not using these peak and valley interrupts, mask them by using the ICTAPAnIPEK0 and ICTAPAnIVLY0 registers, respectively.

29.5.2 Block Diagram

The INT signal output selection function connects TAUDn and TAPAn by using the registers shown below.



29.5.3 Registers

29.5.3.1 PIC0REG2n0 — Timer I/O Control Register 2n0 (n = 0)

This register selects TAPA0 input.

Access: This register can be read or written in 32-bit units.

Address: PIC0REG200: FFDD 00C0$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	PIC0REG2n025	PIC0REG2n024	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.21 PIC0REG2n0 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	*1
25, 24	PIC0REG2n025 PIC0REG2n024	Select the TAUDn channel used for TAPATSIM0 and TAPATUDCM0. 00: Not selected 01: TAUD0 channel 0 10: TAUD0 channel 2 11: TAUD0 channel 8
23 to 0	Reserved	*1

Note 1. Bits defined as 0 in the PIC0REG2n0 register may be defined by another timer connection function. For such bits, use the bit definition of the timer connection function.

29.6 A/D Conversion Trigger Selection Function

This function outputs the A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1 from the TAPATCDENS0 and TAPATCDENS1 signals, which are connected to a compare match interrupt based on the triangular carrier cycle of TAUD, or a valley interrupt signal (TAPATIVLY0).

29.6.1 Configuration of A/D Conversion Trigger Selection Function

Table 29.22 Signals Used for TAPATADOUT Generation

Output Signal	Slave Match Detection Signal	Valley Interrupt Signal
TAPATADOUT0	TAPATCDENS0	TAPATIVLY0
TAPATADOUT1	TAPATCDENS1	TAPATIVLY0

Table 29.23 Operation of TAPATADOUT1 According to the Setting of TAPAnCTL1.TAPAnATS[3:2]

TAPAnATS3	TAPAnATS2	Description
0	0	Outputs the INT signal from TAPATADOUT1 while the triangle wave is falling (counting down).
0	1	Outputs the INT signal from TAPATADOUT1 while the triangle wave is rising (counting up).
1	0	Outputs the INT signal from TAPATADOUT1 while the triangle wave is rising (counting up) or falling (counting down).
1	1	Outputs the INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt TAPATIVLY0 from TAPATADOUT1.

Table 29.24 Operation of TAPATADOUT0 According to the Setting of TAPAnCTL1.TAPAnATS[1:0]

TAPAnATS1	TAPAnATS0	Description
0	0	Outputs the INT signal from TAPATADOUT0 while the triangle wave is falling (counting down).
0	1	Outputs the INT signal from TAPATADOUT0 while the triangle wave is rising (counting up).
1	0	Outputs the INT signal from TAPATADOUT0 while the triangle wave is rising (counting up) or falling (counting down).
1	1	Outputs the INT signal while the triangle wave is rising (counting up) or falling (counting down) and valley interrupt TAPATIVLY0 from TAPATADOUT0.

29.6.2 Block Diagram

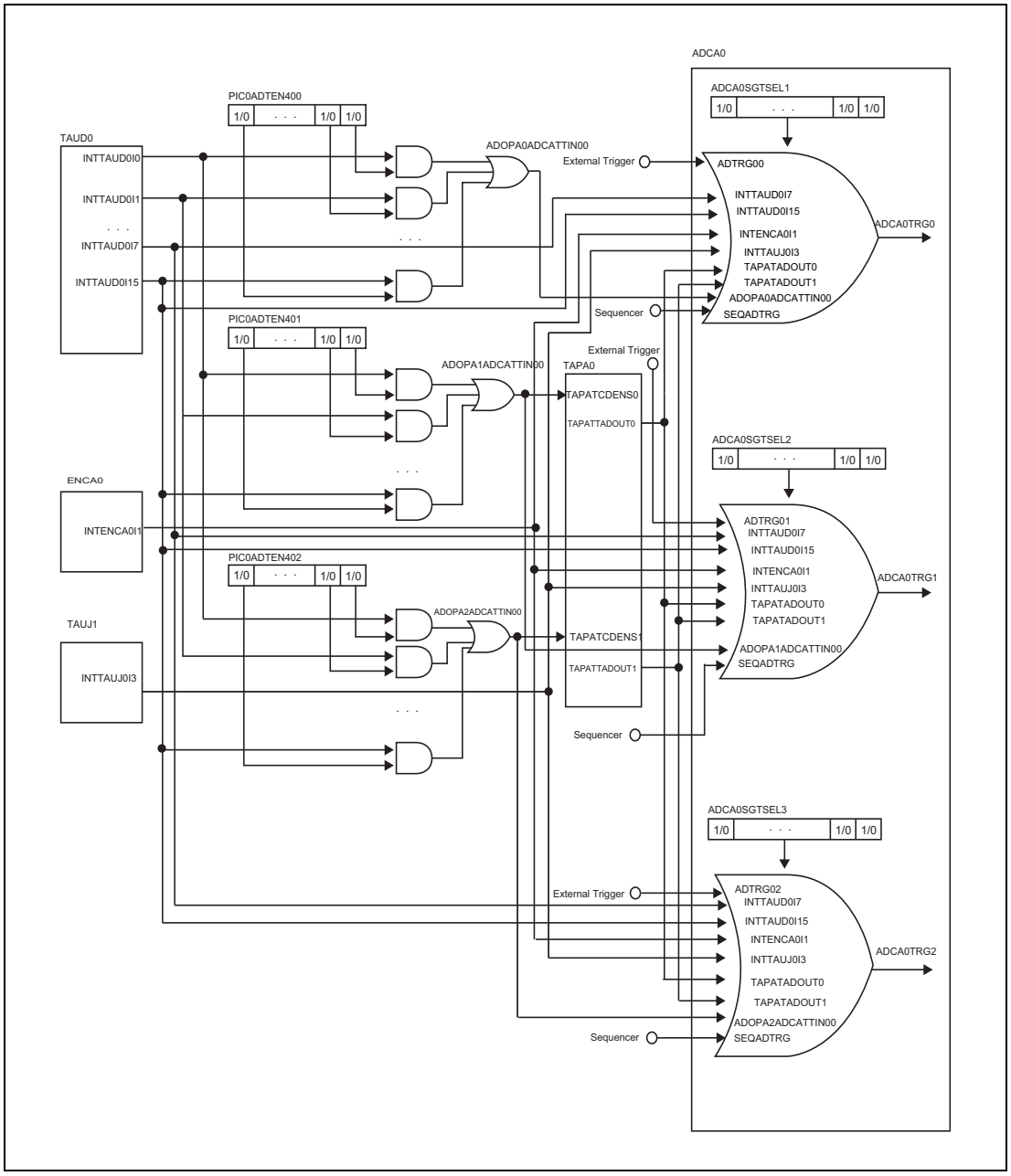


Figure 29.8 Block Diagram of A/D Conversion Trigger Selection Function

NOTE

See **Section 31.3.4.1, ADCAnSGTSELx — Scan Group x Start Trigger Control Register x** for the details on the settings of ADCA0SGTSEL register.

29.6.3 Waveforms of A/D Conversion Trigger Output Control Operation in Triangle PWM Mode

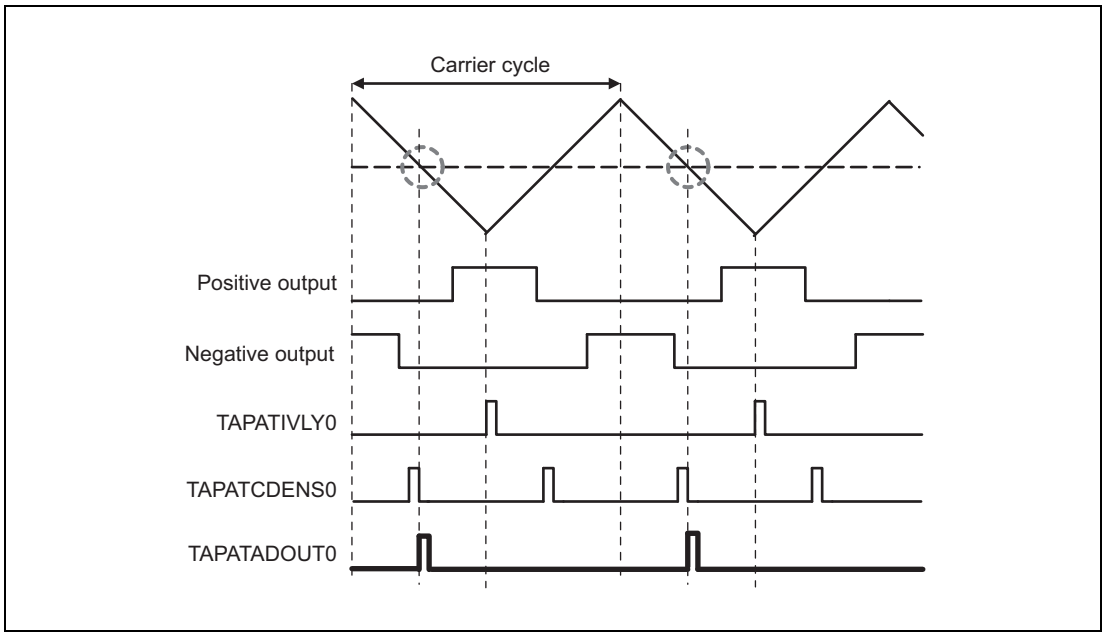


Figure 29.9 TAPAnATS[1:0] bits = 00_B: Output of INT Signal while the Triangle Wave is Falling (Counting Down)

While the triangle wave is falling (counting down), the signals TAPATCDENS0 and TAPATCDENS1 are output as the A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

No A/D conversion trigger signal is output while the triangle wave is rising (counting up).

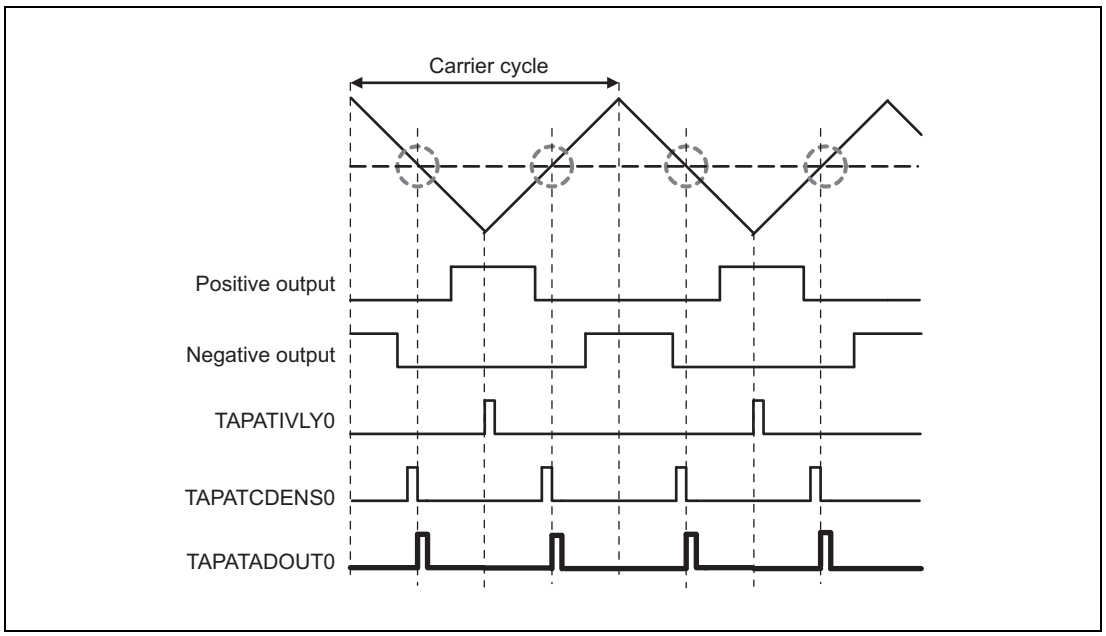


Figure 29.10 TAPAnATS[1:0] bits = 10_B: Output of INT Signal while the Triangle Wave is Rising (Counting Up) or Falling (Counting Down)

The signals TAPATCDENS0 and TAPATCDENS1 are output as the A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

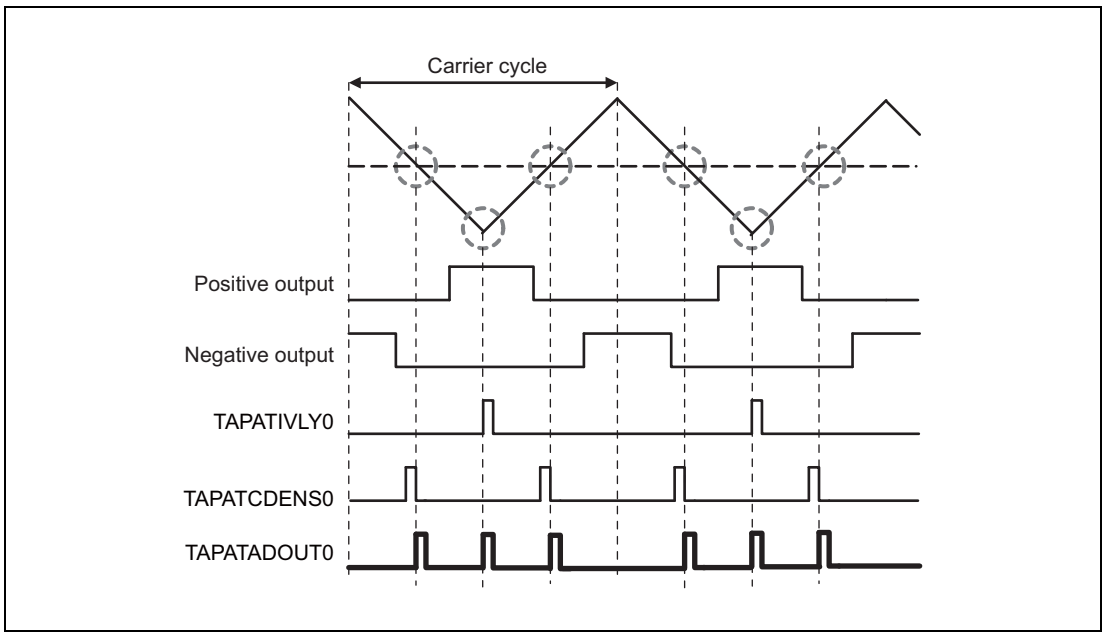


Figure 29.11 TAPAnATS[1:0] bits = 11_B: Output of INT Signal and Valley Interrupt while the Triangle Wave is Rising (Counting Up) or Falling (Counting Down)

The signals TAPATCDENS0 and TAPATCDENS1 and valley interrupt TAPATIVLY0 are output as the A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

29.6.4 Operating Procedure for A/D Conversion Trigger Selection Function

The operating procedure for the A/D conversion trigger selection function is shown below.

	Operation	Status of TAUD and TAPA
Initial setup	Initialize TAUD. Specify the timer operation mode. Set up the TAPAnCTL1 register. Specify TAPAnATS[1:0] (TAPATADOUT0 setting). Specify TAPAnATS[3:2] (TAPATADOUT1 setting). Set up the PIC0ADTEN4nj and PIC0REG2n0 registers according to the signal to be used. Specify PIC0ADTEN4nj (TAPATCDENS0 or TAPATCDENS1 setting). Specify PIC0REG2n0 (TAPATIVLY0 setting).	TAUD and TAPA stop the operation.
Start operation	Start the TAUD operation.	TAUD starts count operation.
During operation	TAUD operates according to the setting of each function.	The A/D conversion trigger selection function outputs either TAPATADOUT0 according to the setting of TAPAnATS[1:0] or TAPATADOUT1 according to the setting of TAPAnATS[3:2], based on the interrupt TAPATCDENS1 or TAPATCDENS0, which is input from TAUD, and the valley interrupt TAPATIVLY0, which is generated by TAPA.
Stop operation	Stop the TAUD operation.	TAUD stops the count operation.

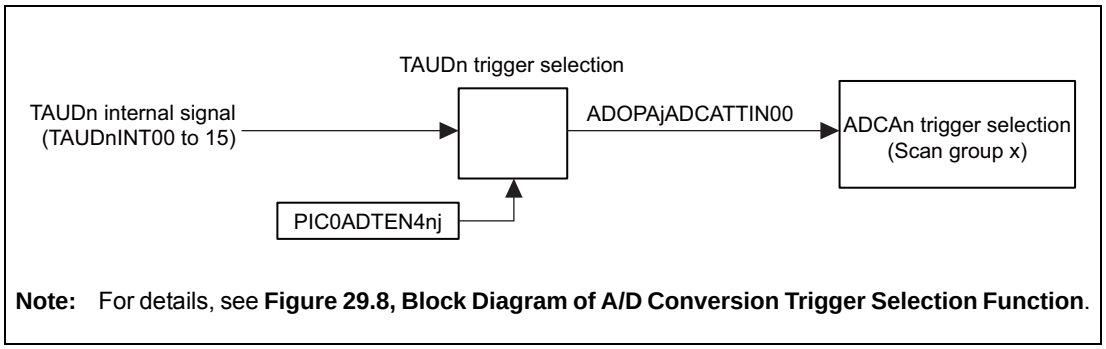
Restart

29.7 ADCA Trigger Selection Function

29.7.1 Functional Overview

This function generates ADCA hardware trigger signals by using TAUDn channel output.

29.7.2 Configuration



Note: For details, see **Figure 29.8, Block Diagram of A/D Conversion Trigger Selection Function**.

Figure 29.12 Block Diagram of ADCA Trigger Selection Function

29.7.3 Registers

29.7.3.1 PIC0ADTEN4nj — A/D Conversion Trigger Output Control Register 4nj (n = 0, j = 0 to 2)

This register selects an ADCA0 start trigger source from TAUDn channel m. (m = 0 to 15)

Access: This register can be read or written in 16-bit units.

Address: <PIC0_base> + 90H + 4 × j

Value after reset: 0000 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC0 ADTEN 4nj15	PIC0 ADTEN 4nj14	PIC0 ADTEN 4nj13	PIC0 ADTEN 4nj12	PIC0 ADTEN 4nj11	PIC0 ADTEN 4nj10	PIC0 ADTEN 4nj09	PIC0 ADTEN 4nj08	PIC0 ADTEN 4nj07	PIC0 ADTEN 4nj06	PIC0 ADTEN 4nj05	PIC0 ADTEN 4nj04	PIC0 ADTEN 4nj03	PIC0 ADTEN 4nj02	PIC0 ADTEN 4nj01	PIC0 ADTEN 4nj00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.25 PIC0ADTEN4nj Register Contents

Bit Position	Bit Name	Function
15 to 0	PIC0ADTEN4nj 15 to PIC0ADTEN4nj 00	Sets a trigger source of CHm (m = 0 to 15) in the TAUDn timer. 0: A/D trigger source of CHm in the TAUDn timer is disabled. 1: A/D trigger source of CHm in the TAUDn timer is enabled.

29.7.4 Example of Operation

(1) Initial setting: Set the function of each channel of the TAUD0 timer to be used.

(2) Setting of the A/D conversion trigger output control register 4nj (PIC0ADTEN4nj):
Setting of the bits of the A/D conversion trigger output control register 4nj ((PIC0ADTEN4nj) to 1 enables selection of an interrupt request signal from each channel in the TAUD0 timer as the trigger of the A/D conversion scan group.

– Register setting should be performed when the A/D conversion is stopped.

(3) Setting of the A/D conversion trigger selection control register (ADCA0SGTSELx):
Setting the bits corresponding to each trigger to 1 enables to use the signal generated by executing the logical OR of each trigger as the start trigger of the A/D conversion scan group.

– Register setting should be performed when the A/D conversion is stopped.

(4) Enabling of TAUD0 timer operation
Each channel in the TAUD0 timer set in (1) starts.

29.7.5 Setup Flow

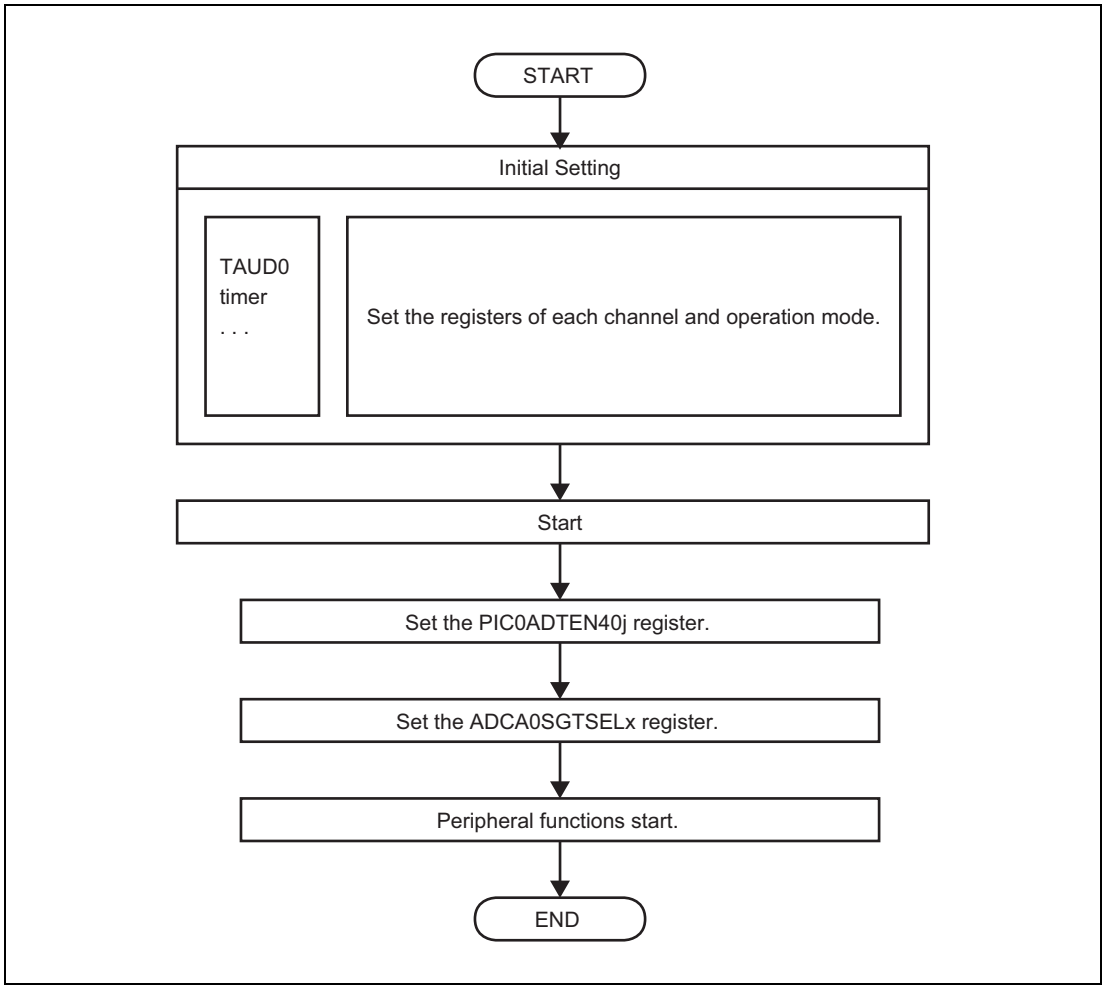


Figure 29.13 Setup Flow (j = 0 to 2)

29.8 Simultaneous Start Trigger Function

29.8.1 Functional Overview

The timers (TAUD0, TAUJ1, ENCA0) can be simultaneously started in any combinations.

29.8.2 Configuration

(1) Configuration

Table 29.26 Configuration of Simultaneous Start Trigger Function

Configuration/Timer Function	Timer
Configuration of Timer	TAUD0, TAUJ1, ENCA0

(2) Block Diagram

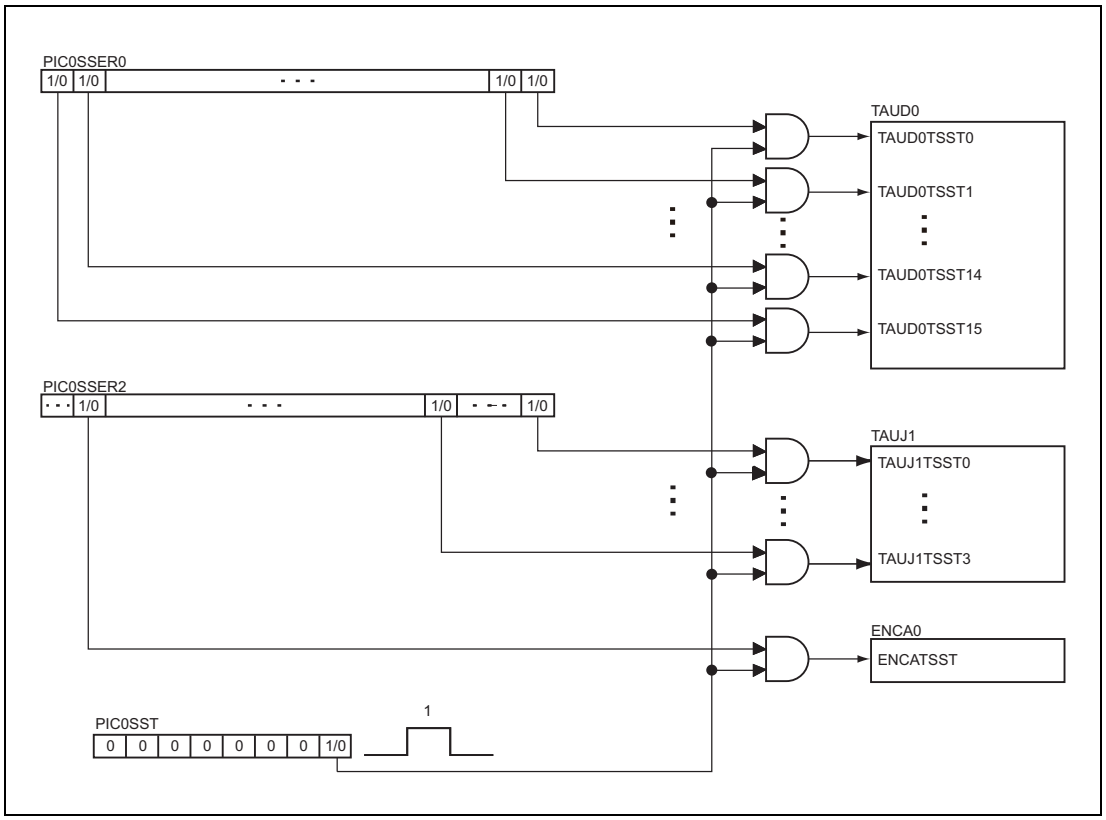


Figure 29.14 Block Diagram of Simultaneous Start Trigger

29.8.3 Registers

29.8.3.1 PIC0SSER0 — Simultaneous Start Control Register 0

The PIC0SSER0 register enables a start trigger for each channel of the TAUD0 timer.

Access: This register can be read or written in 16-bit units.

Address: <PIC0_base> + 10_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIC0SSER015	PIC0SSER014	PIC0SSER013	PIC0SSER012	PIC0SSER011	PIC0SSER010	PIC0SSER009	PIC0SSER008	PIC0SSER007	PIC0SSER006	PIC0SSER005	PIC0SSER004	PIC0SSER003	PIC0SSER002	PIC0SSER001	PIC0SSER000
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.27 PIC0SSER0 Register Contents

Bit Position	Bit Name	Function
15 to 0	PIC0SSER015 to PIC0SSER000	Enable a simultaneous start trigger for the CHm in the TAUD0 timer. 0: Simultaneous start trigger is disabled. 1: Simultaneous start trigger is enabled.

29.8.3.2 PIC0SSER2 — Simultaneous Start Control Register 2

The PIC0SSER2 register enables a start trigger for ENCA0 and TAUJ1.

Access: This register can be read or written in 16-bit units.

Address: <PIC0_base> + 18_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PIC0SSER214(ENCA0)	—	—	—	—	—	—	—	—	—	—	PIC0SSER203	PIC0SSER202	PIC0SSER201	PIC0SSER200
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 29.28 PIC0SSER2 Register Contents

Bit Position	Bit Name	Function
15	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
14	PIC0SSER214	Enables a simultaneous start trigger for the ENCA0 timer. 0: Simultaneous start trigger is disabled. 1: Simultaneous start trigger is enabled.
13 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	PIC0SSER203 to PIC0SSER200	Set a simultaneous start trigger for the CHm in the TAUJ1 timer. 0: Simultaneous start trigger is disabled. 1: Simultaneous start trigger is enabled.

29.8.3.3 PIC0SST — Simultaneous Start Trigger Control Register

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PIC0_base> + 04_H

Value after reset: 00_H

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

Table 29.29 PIC0SST Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	SYNCTRG	Generates a start trigger for the timer whose simultaneous start is enabled. When read, this bit is always read as 0. 0: Disabled. 1: Simultaneous start trigger is generated (the pulse in the width of 1PCLK is output).

29.8.4 Example of Operation

(1) Operation example of timer configuration:
The timers that operates in operation mode to be selected can be simultaneously started in any combinations.

(2) Simultaneous start enable:
Setting the relevant bits in the PIC0SSER0 and PIC0SSER2 registers of the target timers to be simultaneously started to 1 enables these timers to simultaneously start.

(3) Start trigger output:
Writing 1 to the SYNCTRG bit in the PIC0SST register enables the target timers set in (2) to simultaneously start.

(4) Repeating (2) and (3) for the channels that have not started yet enables the different target timers to simultaneously start in multiple batches.

29.8.5 Setup Flow

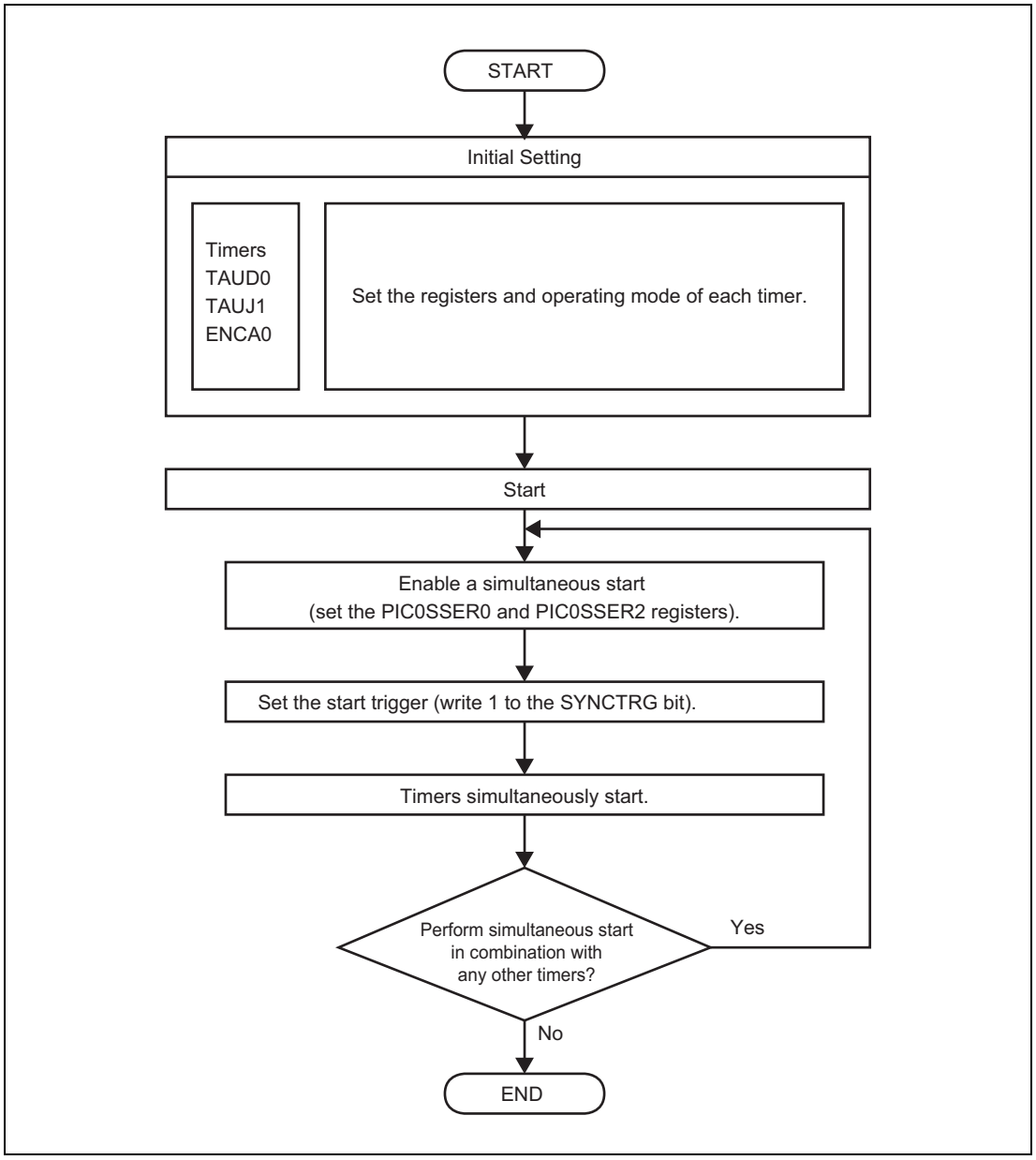


Figure 29.15 Setup Flow

29.9 Trigger & Pulse Width Measuring Function

29.9.1 Functional Overview

This function allows measurement of trigger periods by inputting the trigger signal output from ENCA0 to TAUJ1 and TAUD0.

29.9.2 Configuration

(1) Configuration

Table 29.30 Configuration of Trigger & Pulse Width Measuring Function

Configuration/Timer Function	Timer
Configuration of Timer	ENCA0, TAUD0, TAUJ1

Table 29.31 Setting Functions of TAUJ1/TAUD0 Channels

TAU	Channels	Functions Name	M/S*1	Target Trigger of Pulse Width Measurement
TAUJ1	00	TINm input pulse interval measurement function	S	ENCAT0IEC*2
	01	TINm input pulse interval measurement function	S	ENCAT0IEC*2
TAUD0	00	TINm input pulse interval measurement function	S	ENCAT0EQ0, ENCAT0EQ1
	01	TINm input pulse interval measurement function	S	ENCAT0EQ1
	02	TINm input pulse interval measurement function	S	ENCAT0EQ0

Note 1. M: Master channel, S: Slave channel

Note 2. Read ENCAT0IEC as ENCATIEC (encoder clear interrupt) in **Table 28.5, Interrupt Requests**.

(2) Block Diagram

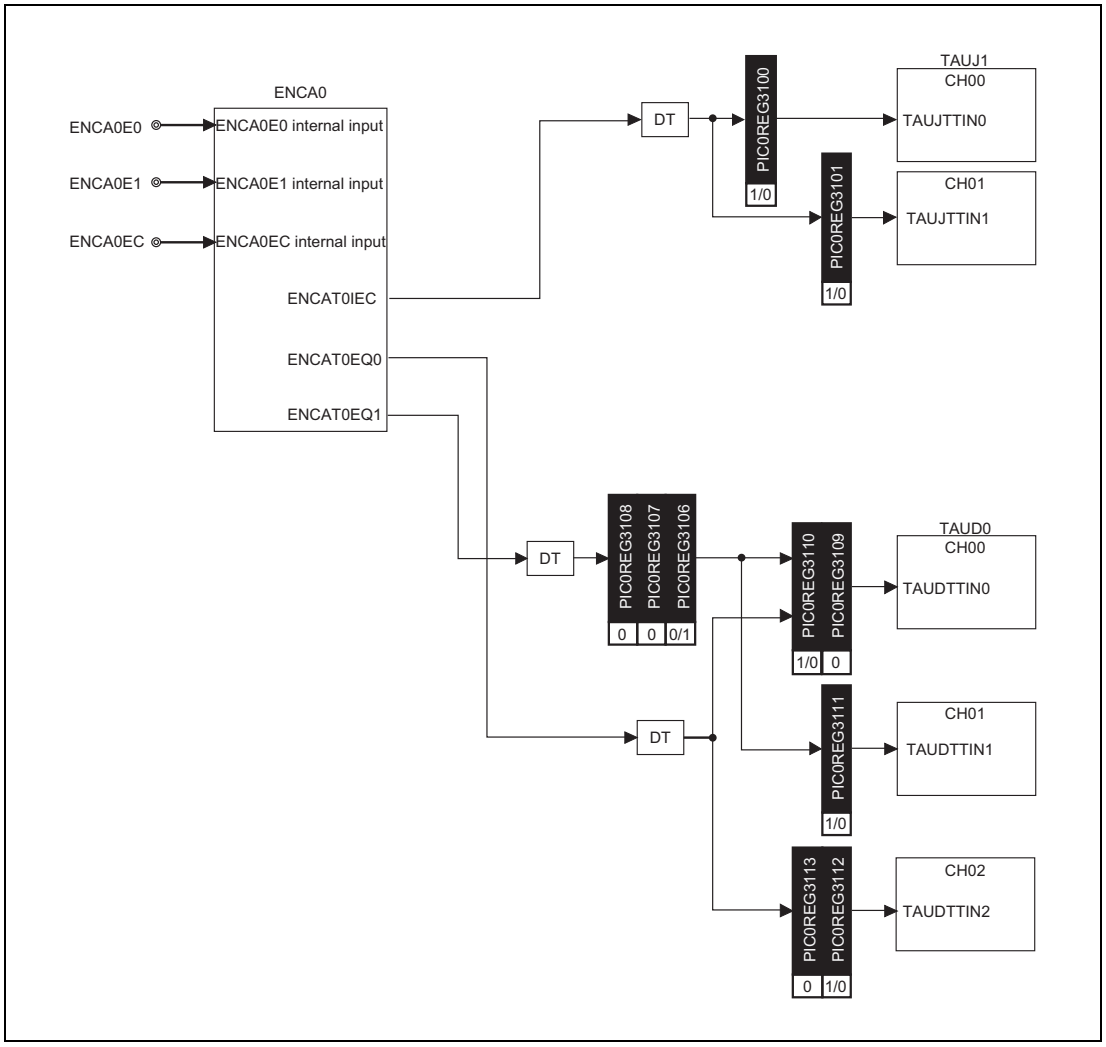


Figure 29.16 Block Diagram of Trigger & Pulse Width Measuring Function

29.9.3 Registers

29.9.3.1 PIC0REG31 — Timer I/O Control Register 31

Access: This register can be read or written in 32-bit units.

Address: <PIC0_base> + EC_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PIC0REG3113	PIC0REG3112	PIC0REG3111	PIC0REG3110	PIC0REG3109	PIC0REG3108	PIC0REG3107	PIC0REG3106	—	—	—	—	PIC0REG3101	PIC0REG3100
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 29.32 PIC0REG31 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
13, 12	PIC0REG3113 to PIC0REG3112	Select a TIN input signal to the CH2 of the TAUD0.

PIC0REG3113	PIC0REG3112	Input Signal
0	0	CH2 of TAUD0 is not used for trigger width measurement.
0	1	DT output signal of ENCAT0EQ0
Other than the above		Setting prohibited

Bit Position	Bit Name	Function
11	PIC0REG3111	Select a TIN input signal to the CH1 of the TAUD0. 0: CH1 of TAUD0 is not used for trigger width measurement. 1: Signal selected in the PIC0REG3106 to PIC0REG3108 (when measuring the ENCATEQ1 signal)
10, 9	PIC0REG3110 to PIC0REG3109	Select a TIN input signal to the CH0 of the TAUD0.

PIC0REG3110	PIC0REG3109	Input Signal
0	0	Signal selected in the PIC0REG3106 to PIC0REG3108
1	0	DT output signal of ENCAT0EQ0
Other than the above		Setting prohibited

Bit Position	Bit Name	Function
8 to 6	PIC0REG3108 to PIC0REG3106	Select a TIN input signal to the CH0 and CH1 of the TAUD0.

PIC0REG3108	PIC0REG3107	PIC0REG3106	Input Signal
0	0	0	CH0 of TAUD0 is not used for trigger width measurement.
0	0	1	DT output signal of ENCAT0EQ1
Other than the above			Setting prohibited

Bit Position	Bit Name	Function
5 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 29.32 PIC0REG31 Register Contents (2/2)

Bit Position	Bit Name	Function
1	PIC0REG3101	Selects a TIN input signal to the CH1 of the TAUJ1. 0: CH1 of TAUJ1 is not used for trigger width measurement. 1: DT output signal of ENCAT0IEC
0	PIC0REG3100	Selects a TIN input signal to the CH0 of the TAUJ1. 0: CH0 of TAUJ1 is not used for trigger width measurement. 1: DT output signal of ENCAT0IEC

29.9.4 Example of Operation

The trigger and pulse width measurement function is achieved by combining the ENCA0 trigger signals (ENCAT0IEC, ENCAT0EQ0, ENCAT0EQ1) and the following function of TAUD0 and TAUJ1.

- TAUDTTINm input pulse interval measurement function (TAUD0)
- TAUJTTINm input pulse interval measurement function (TAUJ1)

Also, the function of PIC, described below, is used to convert the trigger signal input to TINm into a level-sensitive toggle signal.

- DT circuit

The trigger and pulse width measurement function implements measurement of the ENCA0 output trigger signal interval using the TAUDTTINm input pulse interval measurement function of TAUD0 and the TAUJTTINm input pulse interval measurement function of TAUJ1.

(1) TAUDTTINm input pulse interval measurement function, TAUJTTINm input pulse interval measurement function

When the valid TINm edge of TAUD0 or TAUJ1 is detected, the CNTm value is captured into CDRm and the CNTm is cleared.

CAUTION

Set the both edges (rising and falling edges) of TINm to be detected as valid (TAUD0CMURm.TAUD0TIS[1:0] = 10_B, TAUJ1CMURm.TAUJ1TIS[1:0] = 10_B) for this function.

For details of the TAUD and TAUJ functions, see the corresponding sections.

(2) DT circuit

The DT circuit is used to convert the trigger signal output from ENCA0 into a level-sensitive toggle signal.

As shown in **Figure 29.17, Operation of DT Circuit**, the output signal is toggled on each input trigger signal generation.

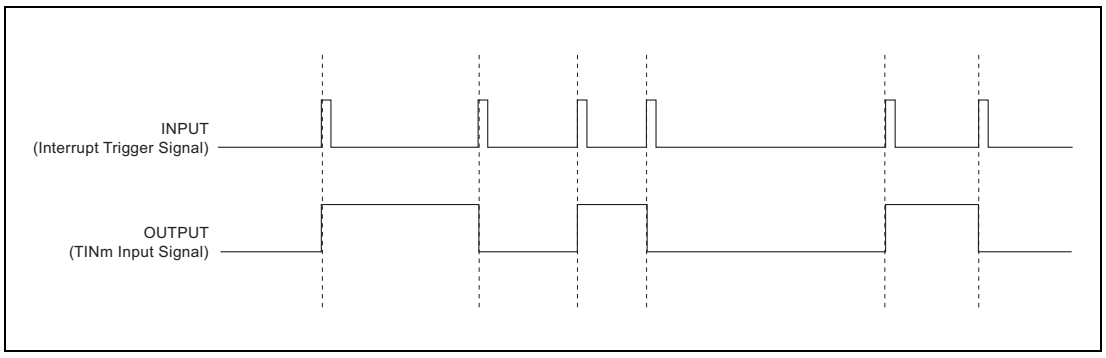


Figure 29.17 Operation of DT Circuit

The PIC provides the input signal conversion and signal connection to TAUD0 and TAUJ1 to measure the generation interval of trigger signals from ENCA0.

The timing chart of the trigger and pulse width measurement function is shown below.

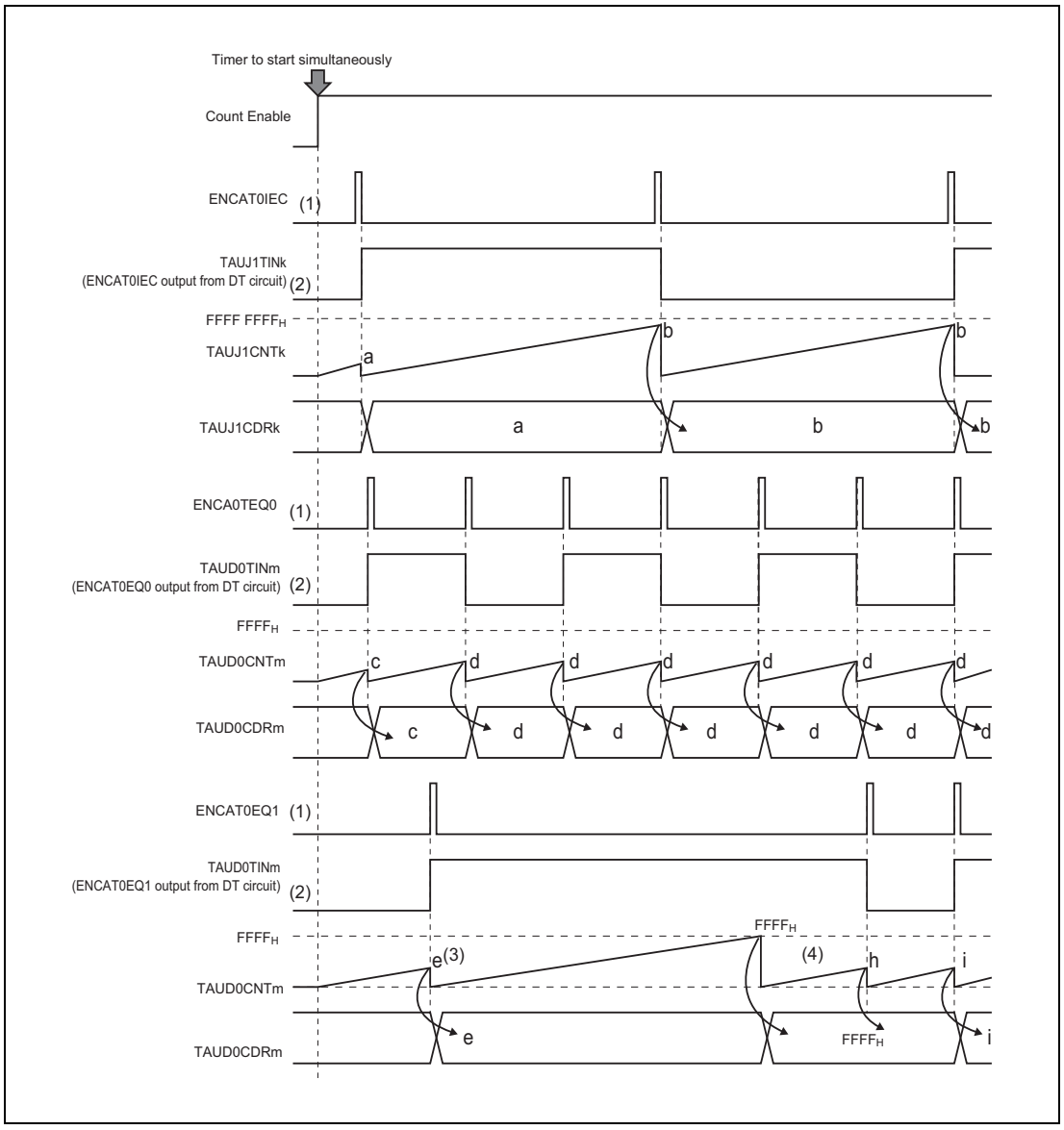


Figure 29.18 Operation Example of the Function of Trigger and Pulse Width Measurement (m = 0 to 2, k = 0, 1)

(1) Following signals are output from ENCA0 as triggers:

- ENCAT0IEC (interrupt trigger signal output when timer counter value is cleared by ENCA0EC input)
- ENCAT0EQ0 (trigger signal output according to timing of a match of timer counter value and value of compare register 0)
- ENCAT0EQ1 (trigger signal output according to timing of a match of timer counter value and value of compare register 1)

(2) The trigger signal output from ENCA0 is converted to a level-sensitive toggle signal by the DT circuit and is output to TINm of TAUD0 and TAUJ1.

(3) Setting both TINm edges for TAUD0 and TAUJ1 as effective edges makes the CNTm value loaded into CDRm at the TINm toggle timing. At the same time, the operation to clear CNTm to 0000_H is repeated.

The first captured value (shown as “a” in the figure) from the start of operation indicates the interval from the start of TAUJ operation until trigger input.

(4) When an overflow occurs, the count value $FFFF_H$ ($FFFF\ FFFF_H$ for TAUJ) is captured but the count value is not captured on the first trigger after the overflow.

With the above operation, the trigger generation interval can be measured.

The following table shows the combinations of the trigger signals and measurement timers, and the bit settings of the pertinent PIC registers for setting the signal paths and the I/O selection registers. Appropriately set these bits according to the trigger signal to be measured and the measurement timer to be used.

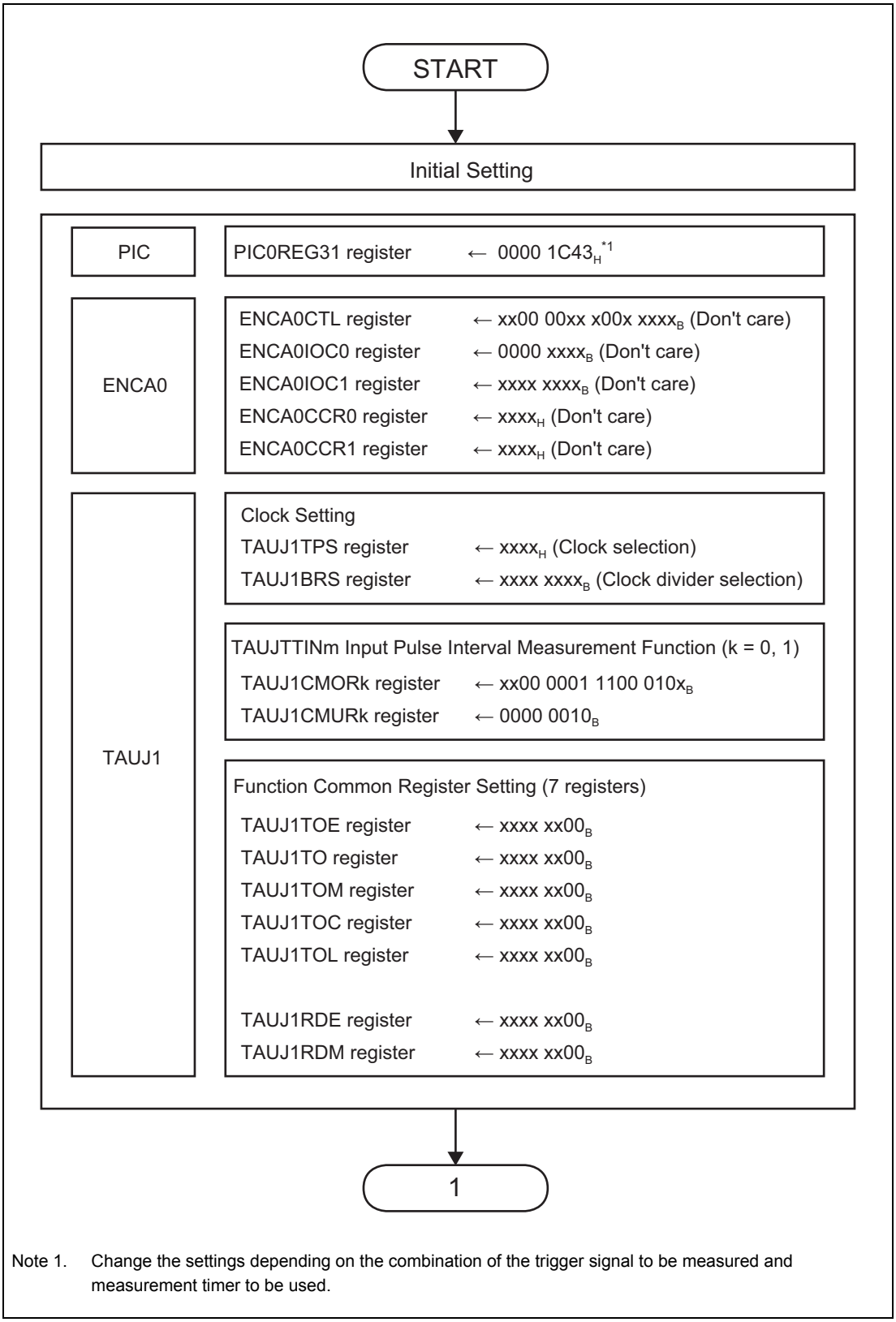
Table 29.33 Combinations of Trigger Signals and Measurement Timers

Interrupt Trigger Signal	Measurement Timer	PIC Register Bit Setting	
ENCAT0IEC	TAUJ1 CH0	PIC0REG3100 = 1	
	TAUJ1 CH1	PIC0REG3101 = 1	
ENCAT0EQ0	TAUD0 CH0	PIC0REG3109 = 0 PIC0REG3110 = 1	
	TAUD0 CH2	PIC0REG3112 = 1 PIC0REG3113 = 0	
ENCAT0EQ1	TAUD0 CH0	PIC0REG3106 = 1 PIC0REG3107 = 0 PIC0REG3108 = 0	PIC0REG3109 = 0 PIC0REG3110 = 0
	TAUD0 CH1		PIC0REG3111 = 1

29.9.5 Setup Flow

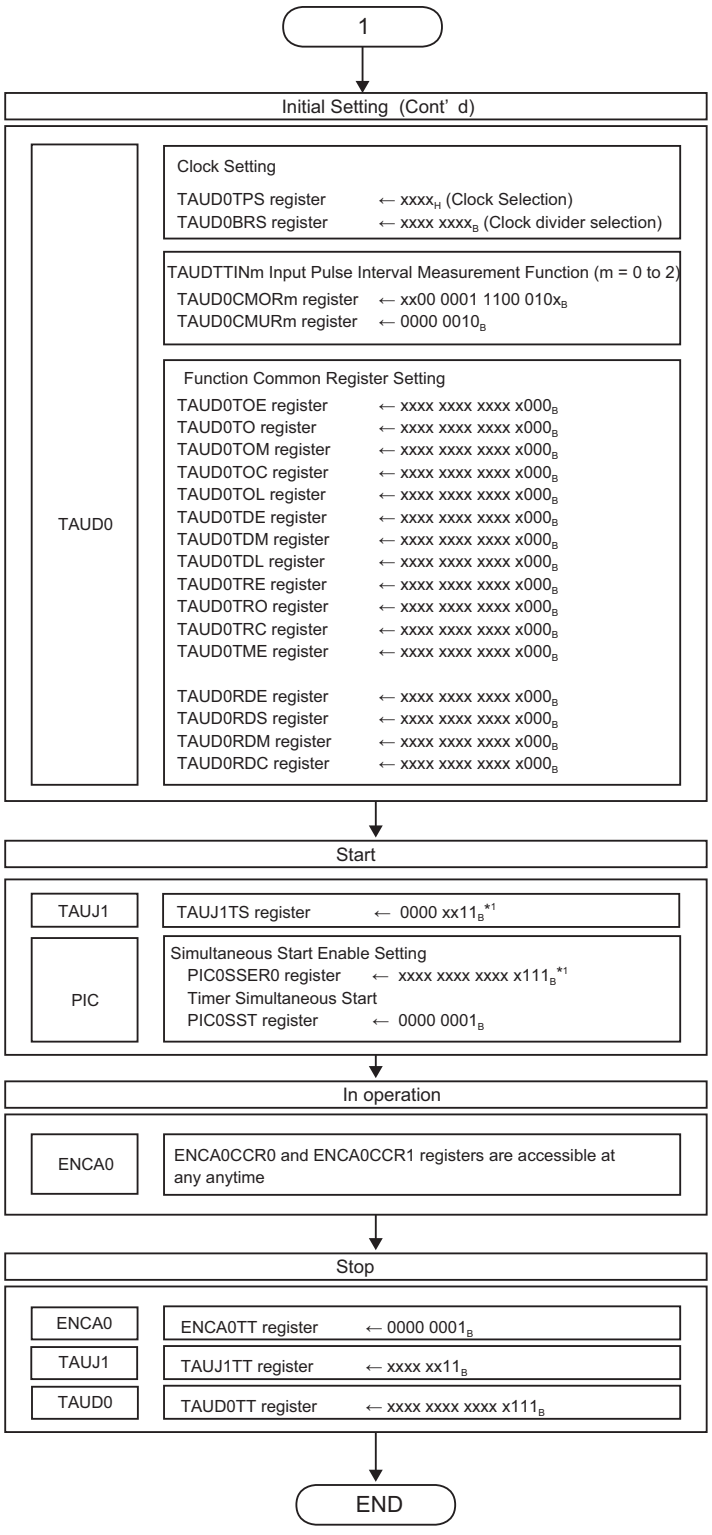
The setup flow in this section shows the general setup flow to measure the pulse interval, which applies to all the following combinations. For the combinations of the trigger signals and measurement timers, see **Table 29.33, Combinations of Trigger Signals and Measurement Timers**.

Encoder Timer	Trigger Signal	Measurement Timer
ENCA0	ENCAT0IEC	TAUJ1 CH0, TAUJ1 CH1
	ENCAT0EQ0	TAUD0 CH0, TAUD0 CH2
	ENCAT0EQ1	TAUD0 CH0, TAUD0 CH1



Note 1. Change the settings depending on the combination of the trigger signal to be measured and measurement timer to be used.

Figure 29.19 Setup Flow



Note 1. Change the settings depending on the combination of the trigger signal to be measured and measurement timer to be used.

Figure 29.20 Setup Flow (Cont'd)

29.9.6 Setting Examples for Operation Functions

This section provides example settings for each register.

The setup example shown in this section describes how to set up measurement of the pulse interval for all the combinations below. For the combinations of the trigger signals and measurement timers, see **Table 29.33, Combinations of Trigger Signals and Measurement Timers**.

Encoder Timer	Trigger Signal	Measurement Timer
ENCA0	ENCAT0IEC	TAUJ1 CH0, TAUJ1 CH1
	ENCAT0EQ0	TAUD0 CH0, TAUD0 CH2
	ENCAT0EQ1	TAUD0 CH0, TAUD0 CH1

Table 29.34 ENCA0 Setting

Register	Bit Position	Bit Name	Setting Value	Note
ENCA0CTL	15	ENCA0CME	Don't care	Enables or disables compare match interrupt detection mask
	14	ENCA0MCS	Don't care	Selects a cancelation trigger for compare match interrupt detection mask
	13 to 10		0	Fixed to 0
	9	ENCA0CRM1	Don't care	Selects the ENCA0CCR1 register function
	8	ENCA0CRM0	Don't care	Selects the ENCA0CCR0 register function
	7	ENCA0CTS	Don't care	Selects trigger for capture operation of ENCA0CCR1.
	6, 5		0	Fixed to 0
	4	ENCA0LDE	Don't care	Enables or disables reload operation when underflow is generated
	3	ENCA0ECM1	Don't care	Enables or disables clearing of the counter on compare match of ENCA0CCR1
	2	ENCA0ECM0	Don't care	Enables or disables clearing of the counter on compare match of ENCA0CCR0
	1, 0	ENCA0UDS[1:0]	Don't care	Selects the counter up/down control by ENCA0E0 and ENCA0E1
ENCA0IOC0	7 to 4		0	Fixed to 0
	3, 2	ENCA0TIS[3:2]	Don't care	Selects the valid edge for capture trigger 1 (ENCA0I1)
	1, 0	ENCA0TIS[1:0]	Don't care	Selects the valid edge for capture trigger 0 (ENCA0I0)
ENCA0IOC1	7	ENCA0SCE	Don't care	Enables encoder special-clear
	6	ENCA0ZCL	Don't care	Selects the clear level of Z phase for encoder special-clear
	5	ENCA0BCL	Don't care	Selects the clear level of B phase for encoder special-clear
	4	ENCA0ACL	Don't care	Selects the clear level of A phase for encoder special-clear
	3, 2	ENCA0ECS[1:0]	Don't care	Selects encoder clear input (Z phase) edge
	1, 0	ENCA0EIS[1:0]	Don't care	Selects encoder input (A or B phase) edge

Table 29.35 TAUJ1 Setting (k = 0, 1)
TAUJ1 (TAUJTTINm Input Pulse Interval Measurement Function)

Register	Bit Position	Bit Name	Setting Value	Note
TAUJ1CMORk	15,14	TAUJ1CKS[1:0]	Don't care	Operation Clock Setting
	13,12	TAUJ1CCS[1:0]	00	
	11	TAUJ1MAS	0	
	10, 9, 8	TAUJ1STS[2:0]	001	
	7, 6	TAUJ1COS[1:0]	11	
	5		0	Fixed to 0
	4, 3, 2, 1	TAUJ1MD[4:1]	0010	
	0	TAUJ1MD0	Don't care	
TAUJ1CMURk	1, 0	TAUJ1TIS[1:0]	10	

NOTE

When TAUJ1CMORk is used for the TAUDTTINm input pulse interval measurement function, the TAUJ1CKS[1:0] (operating clock selection) and TAUJ1MD0 (INTm output control at the start of counting) bits can be set arbitrarily.

Though the TAUJ1COS[1:0] (overflow mode selection) bits can also be set arbitrarily, these bits should be fixed values as specified above for this function.

Other control bits have fixed values as specified above. For details, see **Section 26, Timer Array Unit J (TAUJ)**.

For TAUJ common registers (TAUJ1TOE, TAUJ1TO, TAUJ1TOM, TAUJ1TOC, TAUJ1TOL,TAUJ1RDE, and TAUJ1RDM), only set the bits corresponding to the used channels to 0.

Table 29.36 TAUD0 Setting (m = 0 to 2)
TAUD0 (TAUDTTINm Input Pulse Interval Measurement Function)

Register	Bit Position	Bit Name	Setting Value	Note
TAUD0CMORm	15, 14	TAUD0CKS[1:0]	Don't care	Operation Clock Setting
	13, 12	TAUD0CCS[1:0]	00	
	11	TAUD0MAS	0	
	10 to 8	TAUD0STS[2:0]	001	
	7, 6	TAUD0COS[1:0]	11	
	5		0	Fixed to 0
	4 to 1	TAUD0MD[4:1]	0010	
	0	TAUD0MD0	Don't care	
TAUD0CMURm	1, 0	TAUD0TIS[1:0]	10	

NOTE

When TAUD0CMORm is used for the TAUDTTINm input pulse interval measurement function, the TAUD0CKS[1:0] (operating clock selection) and TAUD0MD0 (INTm output control at the start of counting) bits can be set arbitrarily.

Though the TAUD0COS[1:0] (overflow mode selection) bits can also be set arbitrarily, these bits should be fixed values for this function.

Other control bits have fixed values as specified above. For details, see **Section 25, Timer Array Unit D (TAUD)**

For TAUD common registers (TAUD0TOE, TAUD0TO, TAUD0TOM, TAUD0TOC,TAUD0TOL, TAUD0TDE, TAUD0TDM, TAUD0TDL, TAUD0TRE, TAUD0TRO,TAUD0TRC, TAUD0TME, TAUD0RDE, TAUD0RDS, TAUD0RDM, and TAUD0RDC), only set the bits corresponding to the used channels to 0.

Table 29.37 PIC Setting

Register	Bit Position	Bit Name	Setting Value	Note
PIC0REG31	13, 12	PIC0REG3113 PIC0REG3112	0 1	Selects the DT output signal from ENCAT0EQ0 as TAUD0TTIN2 input signal
	11	PIC0REG3111	1	Selects the signal selected with PIC0REG3106 to PIC0REG3108 (DT output signal from ENCAT0EQ1) as TAUD0TTIN1 input signal
	10, 9	PIC0REG3110 PIC0REG3109	1 0	Selects the DT output signal from ENCAT0EQ0 as TAUD0TTIN0 input signal
	8 to 6	PIC0REG3108 PIC0REG3107 PIC0REG3106	0 0 1	Selects the DT output signal from ENCAT0EQ1 as TAUD0TTIN1 or TAUD0TTIN0 input signal
	1	PIC0REG3101	1	Selects the DT output signal from ENCAT0IEC as TAUJ1TTIN1 input signal
	0	PIC0REG3100	1	Selects the DT output signal from ENCAT0IEC as TAUJ1TTIN0 input signal

29.10 A/D Trigger Encoder Capture Function

29.10.1 Functional Overview

The value of encoder counter synchronized with A/D conversion can be obtained by using an A/D conversion trigger signal as a capture signal of the ENCA0.

29.10.2 Configuration

(1) Configuration

Table 29.38 Configuration of A/D Trigger Encoder Capture Function

A/D Converter	Encoder Timer
ADCA0	ENCA0

(2) Block Diagram

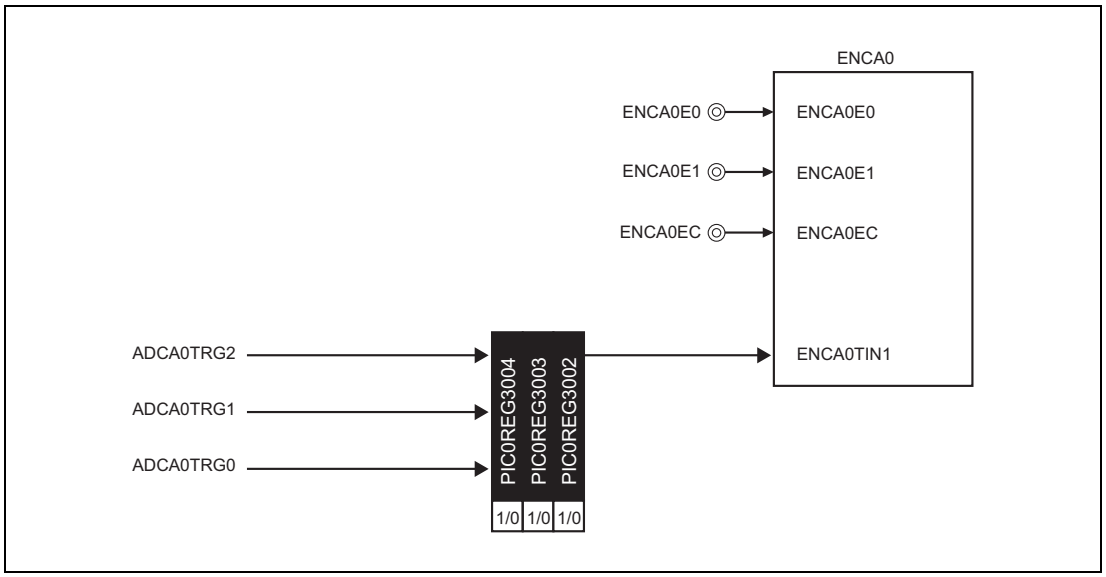


Figure 29.21 Block Diagram of A/D Trigger Encoder Capture Function

CAUTIONS

1. **It takes ENCA0 one additional clock cycle to capture the input signal of the ADCA0TRG0, ADCA0TRG1, and ADCA0TRG2 pins compared with the number of clock cycles it takes to capture the input signal of the ENCA0TIN1 pin when using CKSCLK_IPERI1, and three additional clock cycles when using CKSCLK_AADCA. Be sure to take this into account when configuring your system.**
2. **Configure the edge detection function by using the edge detection function registers of the digital noise filter, which are FCLA0CTL0_ADC0, FCLA0CTL1_ADC0, and FCLA0CTL2_ADC0 (see Section 2.12.1.4, Input Pins that Incorporate Digital Filter Type D, for details), and specify "rising edge" for edge detection of the ENCA0TIN1 capture trigger input of ENCA0 (ENCA0IOC0.ENCA0TIS[3:2] = 01_B). Do not set ENCA0IOC0.ENCA0TIS[3:2] to 10_B (falling edge) or 11_B (both edges).**

29.10.3 Registers

29.10.3.1 PIC0REG30 — Timer I/O Control Register 30

Access: This register can be read or written in 32-bit units.

Address: <PIC0_base> + $E8_H$

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PIC0REG3004	PIC0REG3003	PIC0REG3002	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 29.39 PIC0REG30 Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 2	PIC0REG3004 to PIC0REG3002	Selects an input signal to ENCA0TIN1.
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

PIC0REG3004	PIC0REG3003	PIC0REG3002	Input signal
0	0	0	Capture is not performed by an A/D trigger signal in the ENCA0.
0	1	0	ADCA0TRG2
0	1	1	ADCA0TRG1
1	0	0	ADCA0TRG0
Other than the above			Setting prohibited

29.10.4 Example of Operation

The A/D trigger encoder capture function is implemented by connecting the A/D conversion trigger signal ADCAnTRGi (n = 0, i = 0 to 2) to ENCA0.

CAUTION

When using this function, the ENCA0 interrupt signal ENCA0INT1 should not be selected as the A/D converter trigger. If selected, the correct operation cannot be performed because the following loop occurs: ADCAnTRG1 generation → ENCA0 capture operation → ENCA0INT1 generation by capture operation → ADCAnTRG1 generation.

The following shows a timing chart of the A/D trigger encoder capture function using the ADCA0TRG1 as a trigger.

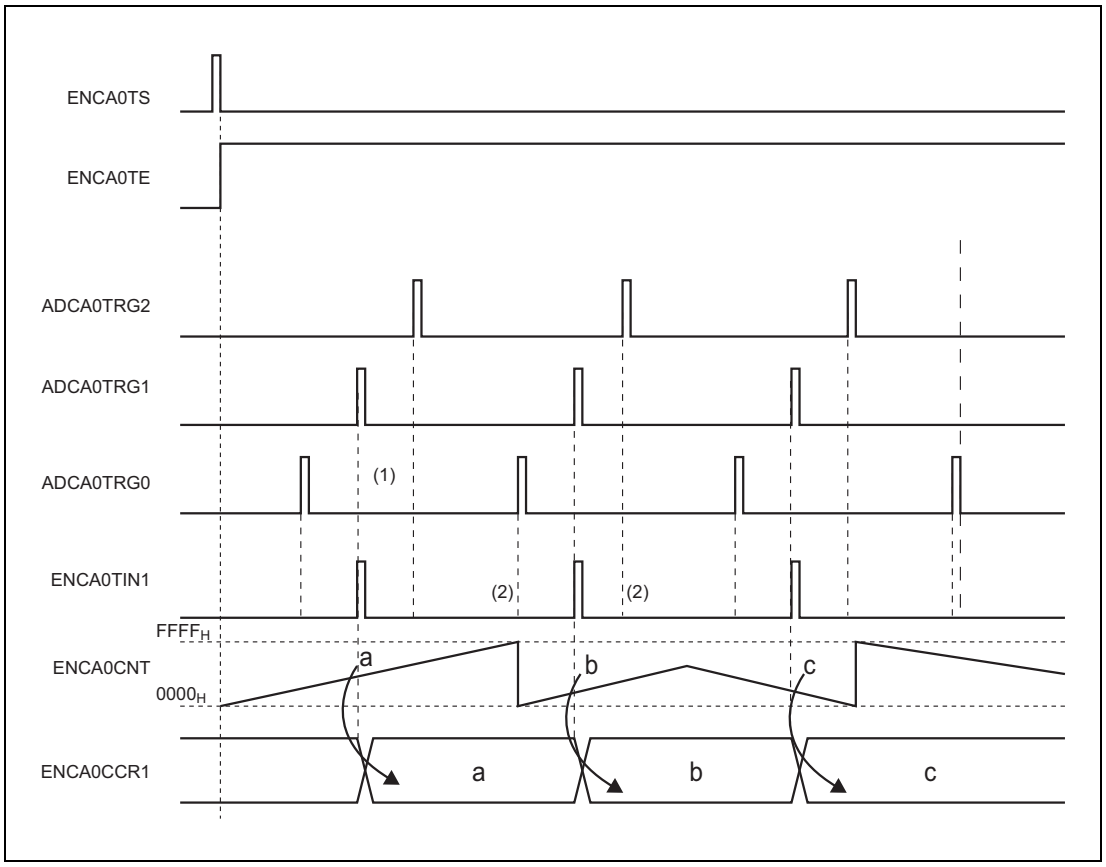


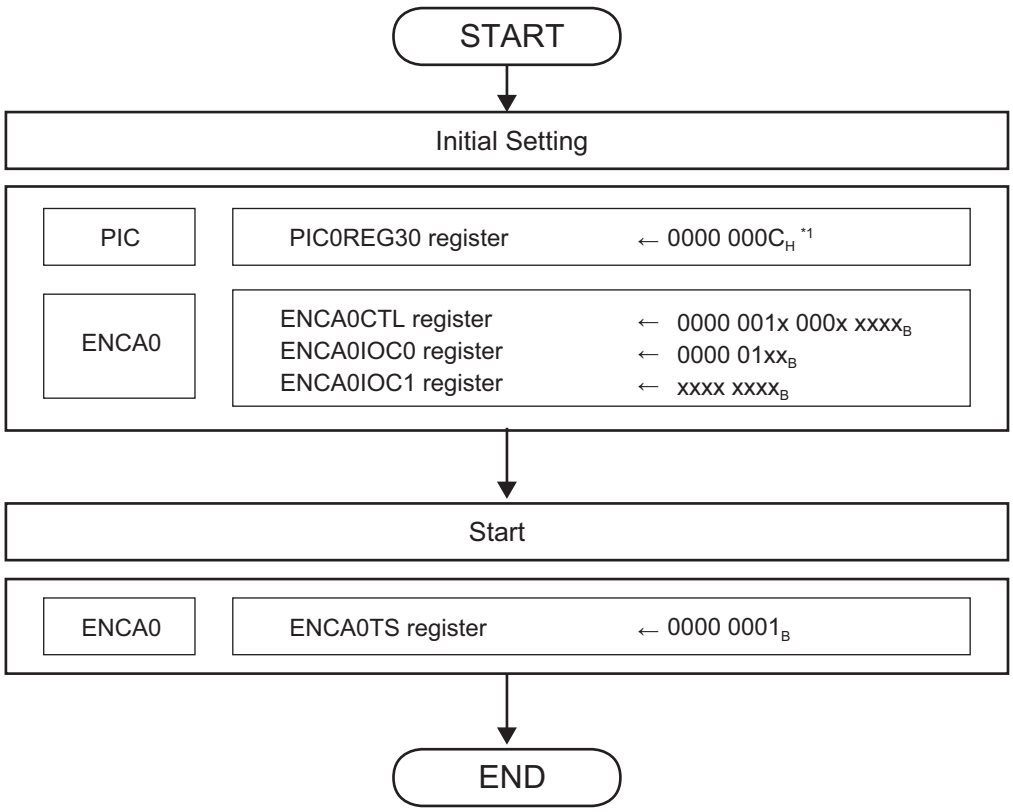
Figure 29.22 Operation Example of Trigger Encoder Capture function

(1) When ADCA0TRG1 is selected as the ENCA0 capture trigger 1 signal ENCA0TIN1, ENCA0 capture occurs to let the effective ADCA0TRG1 be input to ENCA0 as the ENCA0TIN1.

(2) When a hardware trigger signal (ADCA0TRG0, ADCA0TRG2) other than ADCA0TRG1 is generated, the ENCA0TIN1 signal is not generated and ENCA0 is not captured.

29.10.5 Setup Flow

The setup flow in this section shows the general setup flow to perform capture operation of encoder timer ENCA0 with the ADCA0TRG1 signal.



Note 1. Change the settings depending on the hardware trigger to perform the capture operation.

Figure 29.23 Setup Flow

29.10.6 Setting Examples for Operation Functions

This section provides example settings for each register.

The setup example shown in this section describes how to set up capture operation of encoder timer ENCA0 based on the ADCA0TRG1 signal. Change the settings depending on the hardware trigger to perform the capture operation.

Table 29.40 ENCAn Setting

Register	Bit Position	Bit Name	Setting Value	Remark
ENCAnCTL	15	ENCAnCME	0	Disables compare match interrupt detection masking
	14	ENCAnMCS	0	Selects release trigger for compare match interrupt detection masking
	13 to 10		0	Fixed to 0
	9	ENCAnCRM1	1	Sets the ENCAnCCR1 register for capture operation
	8	ENCAnCRM0	Don't care	Selects the function of ENCAnCCR0 register
	7	ENCAnCTS	0	Selects ENCATTIN1 as trigger for capture operation
	6, 5		0	Fixed to 0
	4	ENCAnLDE	Don't care	Enables or disables reload operation when ENCAnCCR0 register underflow occurs
	3	ENCAnECM1	Don't care	Enables or disables clearing of the counter on compare match of ENCAnCCR1 register
	2	ENCAnECM0	Don't care	Enables or disables clearing of the counter on compare match of ENCAnCCR0 register
	1,0	ENCAnUDS[1:0]	Don't care	Selects the counter up/down control by ENCAnE0 and ENCAnE1
ENCAnIOC0	7 to 4		0	Fixed to 0
	3, 2	ENCAnTIS[3:2]	0[*1] 1[*1]	Selects the valid edge of capture trigger 1 (ENCATTIN1) for the rising edge detection
	1, 0	ENCAnTIS[1:0]	Don't care	Selects the valid edge of capture trigger 0 (ENCATTIN0)
ENCAnIOC1	7	ENCAnSCE	Don't care	Enables encoder special-clear
	6	ENCAnZCL	Don't care	Selects the clear level (input level) of Z phase for encoder special-clear
	5	ENCAnBCL	Don't care	Selects the clear level (input level) of B phase for encoder special-clear
	4	ENCAnACL	Don't care	Selects the clear level (input level) of A phase for encoder special-clear
	3, 2	ENCAnECS[1:0]	Don't care	Selects encoder clear input (Z phase) edge
	1, 0	ENCAnEIS[1:0]	Don't care	Selects encoder input (A or B phase) edge
PIC0REG30	4	PIC0REG3004	Don't care	Selects ADCA0 trigger signal of ENCA0TIN1
	3	PIC0REG3003	Don't care	
	2	PIC0REG3002	Don't care	

Note 1. Change the setting depending on the hardware trigger to perform the capture operation.

NOTE

Bits ENCA0CRM1 and ENCA0CTS in ENCA0CTL are fixed: ENCA0CRM1 = 1 (ENCA0CCR1 register function) and ENCA0CTS = 0 (trigger source of capture to the ENCA0CCR1 register). All the other bits can be set arbitrarily.

29.11 Three-Phase PWM Output with Dead Time

29.11.1 Functional Overview

This feature generates each of the set signals (active level change timing signals) and clear signals (inactive level change timing signals) once or less per cycle and then uses the results to output a three-phase PWM waveform with dead time.

For the PWM output feature of TAUD, only the clear timing used during each cycle is specified by specifying the duty value, but for the feature described here, the set timing can also be specified, which makes more flexible PWM output with dead time possible.

29.11.2 Configuration

The unit and channel configuration for this feature is shown below. (n = 0)

Table 29.41 Configuration of Three-Phase PWM Output with Dead Time

Timer	Timer Motor Control Function
TAUD0 CH2, CH4 to CH15 (used channels fixed)	TAPA0

The signal names used in the descriptions below are abbreviations. The actual signal names corresponding to each abbreviation are as follows:

- INTm → INTTAUDnIm (TAUDn channel m interrupt)
- TINm → TAUDTTINm (TAUDn channel m input)
- TOUTm → TAUDTTOUTm (TAUDn channel m output)
- CDRm → TAUDnCDRm (TAUDn channel m data register)
- CNTm → TAUDnCNTm (TAUDn channel m counter register)

(1) TAUDn configuration

Because CH10, CH12, and CH14 are only used for TOUTm, these channels can be used for features that do not use TOUTm (m = 10, 12, 14).

Table 29.42 TAUDn Configuration (1/2)

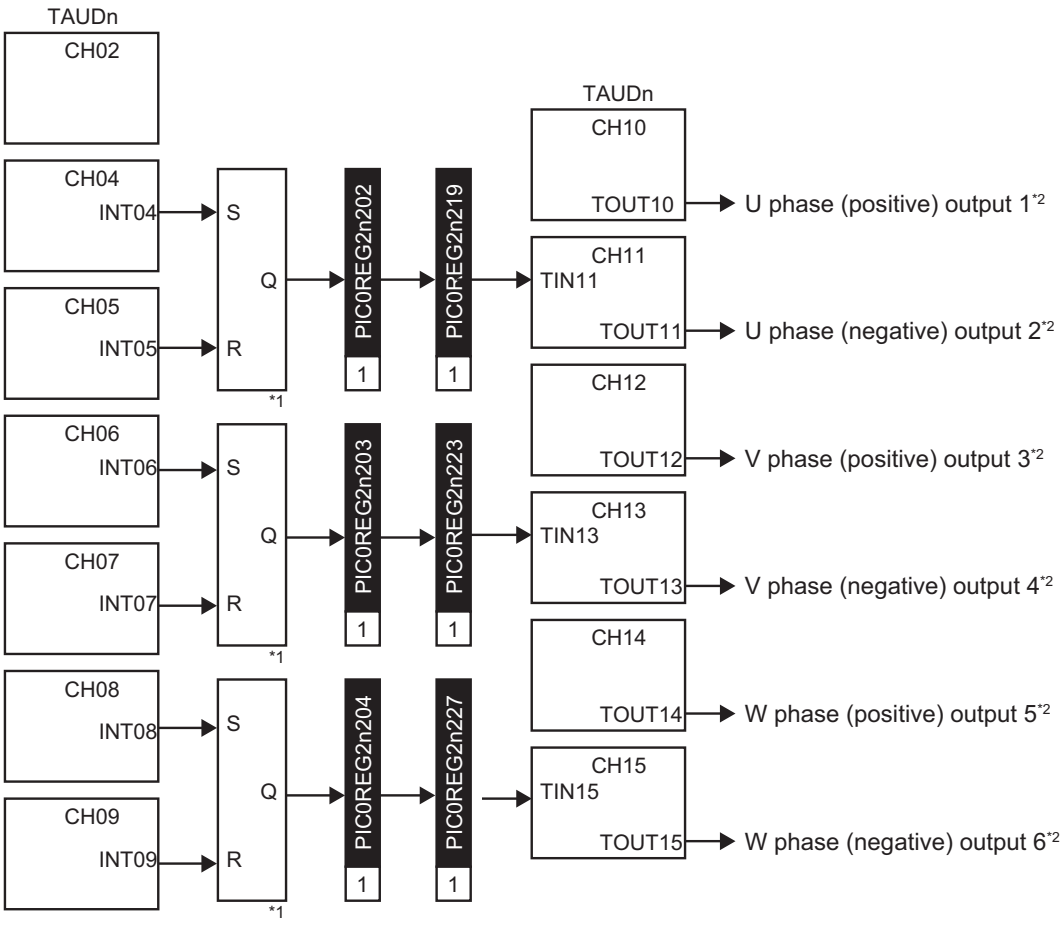
CH	Function Name	M/S	CDR Setting	Description
2	PWM output (CH2 is the master channel for CH4 to CH9.)	M	Cycle	
4		S	Duty (U phase setting)	
5		S	Duty (U phase clearing)	
6		S	Duty (V phase setting)	
7		S	Duty (V phase clearing)	
8		S	Duty (W phase signal setting)	
9		S	Duty (W phase clearing)	
10	Any for features that do not use TOUT10	S		TOUT10: U phase output
11	One-phase PWM output	S	Dead time (U phase)	TOUT11: UB-phase output
12	Any for features that do not use TOUT12	S		TOUT12: V phase output

Table 29.42 TAUDn Configuration (2/2)

CH	Function Name	M/S	CDR Setting	Description
13	One-phase PWM output	S	Dead time (V phase)	TOUT13: VB-phase output
14	Any for features that do not use TOUT14	S		TOUT14: W phase output
15	One-phase PWM output	S	Dead time (W phase)	TOUT15: WB-phase output

Note: M: Master channel, S: Slave channel

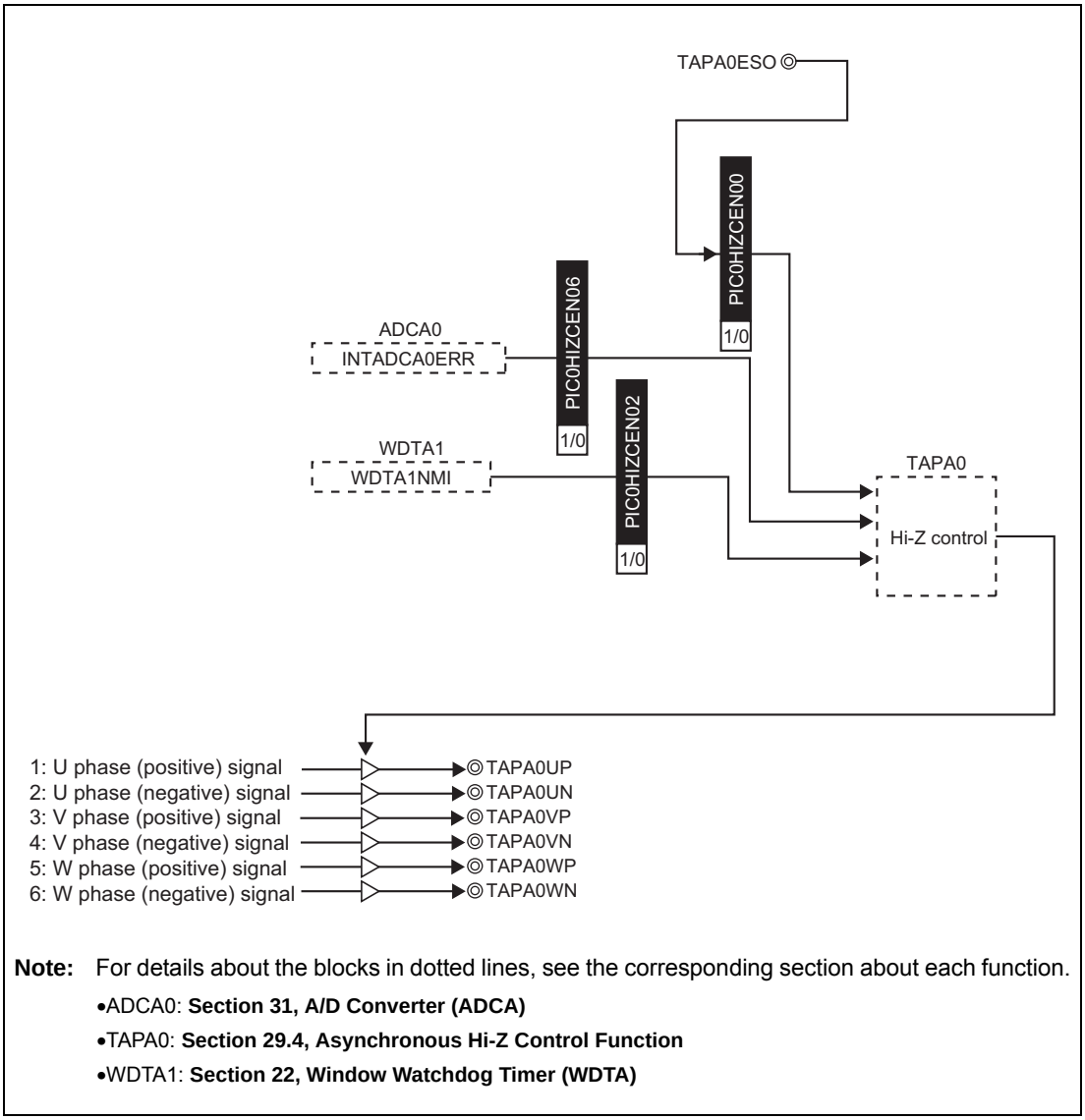
(2) Block diagram



Note 1. SR flip-flop circuit

Note 2. For the connection destination, see **Figure 29.25, Block Diagram: Motor Output Buffer Control**.

Figure 29.24 Block Diagram: Three-Phase PWM Output with Dead Time



Note: For details about the blocks in dotted lines, see the corresponding section about each function.

- ADCA0: **Section 31, A/D Converter (ADCA)**
- TAPA0: **Section 29.4, Asynchronous Hi-Z Control Function**
- WDTA1: **Section 22, Window Watchdog Timer (WDTA)**

Figure 29.25 Block Diagram: Motor Output Buffer Control

29.11.3 Registers

29.11.3.1 PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0)

Access: This register can be read or written in 32-bit units.

Address: PIC0REG202: FFDD 00C8$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PIC0REG2n227	—	—	—	PIC0REG2n223	—	—	—	PIC0REG2n219	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PIC0REG2n204	PIC0REG2n203	PIC0REG2n202	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 29.43 PIC0REG2n2 Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	*1
27	PIC0REG2n227	Selects the signal input to TAUDTTIN15. **PIC0REG2n227** / **Input signal** 1 / Signal selected by the PIC0REG2n204 bit. Other than the above / Setting prohibited
26 to 24	Reserved	*1
23	PIC0REG2n223	Selects the signal input to TAUDTTIN13. **PIC0REG2n223** / **Input signal** 1 / Signal selected by the PIC0REG2n203 bit. Other than the above / Setting prohibited
22 to 20	Reserved	*1
19	PIC0REG2n219	Selects the signal input to TAUDTTIN11. **PIC0REG2n219** / **Input signal** 1 / Signal selected by the PIC0REG2n202 bit Other than the above / Setting prohibited
18 to 5	Reserved	*1
4	PIC0REG2n204	Selects the TIN input signal to TAUDTTIN15. 0: Setting prohibited 1: Select the set/clear output according to INTTAUDnI8 and INTTAUDnI9.
3	PIC0REG2n203	Selects the TIN input signal to TAUDTTIN13. 0: Setting prohibited 1: Select the set/clear output according to INTTAUDnI6 and INTTAUDnI7.
2	PIC0REG2n202	Selects the TIN input signal to TAUDTTIN11. 0: Setting prohibited 1: Select the set/clear output according to INTTAUDnI4 and INTTAUDnI5.
1, 0	Reserved	*1

Note 1. Bits defined as 0 in the PIC0REG2n2 register may be defined by another timer connection function. For such bits, use the bit definition of the timer connection function.

29.11.3.2 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

The PIC0HIZCENn register selects the Hi-Z output control input signal of TAPAn.

Access: This register can be read or written in 8-bit units.

Address: <PIC0_base> + 80_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R	R/W

Table 29.44 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Selects whether to enable or disable Hi-Z output control by the INTADCA0ERR interrupt signal. 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	Selects whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Selects whether to enable or disable Hi-Z output control by the TAPAnESO pin input. 0: Disable 1: Enable

29.11.4 Operation Example

This example shows how to generate each of the set signals and clear signals once or less per cycle and then use the results to output a three-phase PWM waveform with dead time.

This is achieved by combining the following TAUD features:

- PWM output
- One-phase PWM output

In addition, the following function of PIC is used to create the PWM waveform to supply to the input TINm signal (m = 11, 13, or 15) of one-phase PWM output function from the set and clear signals generated by the PWM output function:

- SR flip-flop circuit

Three-phase PWM output is achieved by assigning the one-phase PWM output with dead time achieved using the above features to the U, V, and W phases. Therefore, the set and clear signals of PWM output can be freely specified for each PWM phase. Because the only difference among phases is the assigned channel, only one phase (the U phase) is described below.

29.11.4.1 Pwm Output

PWM output uses a combination of CH2, CH4, and CH5.

By specifying the cycle for CDR02, the U phase set value for CDR04, and the U phase clear value for CDR05, a set/clear signal is generated for the SR flip-flop circuit that generates the input TIN11 signal of one-phase PWM output from INT04 and INT05.

Instead of CH4 and CH5, which are used for the above described U phase set/clear signal generation, the V phase uses CH6 and CH7, and the W phase uses CH8 and CH9.

29.11.4.2 One-Phase PWM Output

One-phase PWM output is generated from TOUT10 and TOUT11 by using a combination of CH10 and CH11.

By specifying the dead time value for CDR11, a one-phase PWM signal with dead time is output for the TIN11 input.

Similarly, the V phase uses CH12 and CH13 to output a one-phase PWM signal with dead time, while the W phase uses CH14 and CH15.

CAUTION

Specify the same clock for each TAUDn channel that uses the PWM output and one-phase PWM output features.

For details about the TAUD functions, see **Section 25, Timer Array Unit D (TAUD)**.

29.11.4.3 SR Flip-Flop Circuit

The PWM waveform supplied to the input TIN11 of one-phase PWM output is generated by using the U phase set signal generated by CH4 of TAUD and the U phase clear signal generated by CH5.

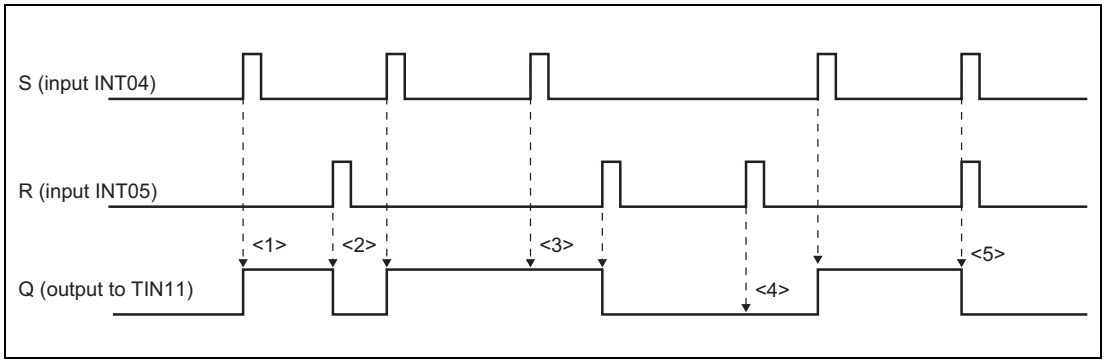


Figure 29.26 SR Flip-Flop Circuit Operation Timing Chart (U phase example)

<1> When a signal is input to input S, output Q goes to the high level at the rising edge of S.

<2> When a signal is input to input R, output Q goes to the low level at the rising edge of R.

<3> If a signal is input to input S while output Q is at the high level, output Q is not affected.

<4> If a signal is input to input R while output Q is at the low level, output Q is not affected.

<5> If a signal is input to input S and input R at the same time, input R takes priority and output Q goes to the low level at the rising edge of R.

The V phase uses INT06 and INT07 as input to supply a PWM waveform to TIN13, and the W phase uses INT08 and INT09 as input to supply a PWM waveform to TIN15.

The output change timing of the PWM waveform generated during one-phase PWM output is based on PWM output.

The active level output timing set signal and inactive level output timing clear signal of PWM are generated during PWM output. By inputting these signals to the SR flip-flop circuit, a PWM signal that can be changed at any time is generated.

A one-phase PWM signal is output by generating a positive or negative PWM waveform and then adding dead time to it according to changes in the generated PWM signal.

The PIC is used to set/clear signal generated during PWM output as the TIN input for one-phase PWM output through the SR flip-flop circuit.

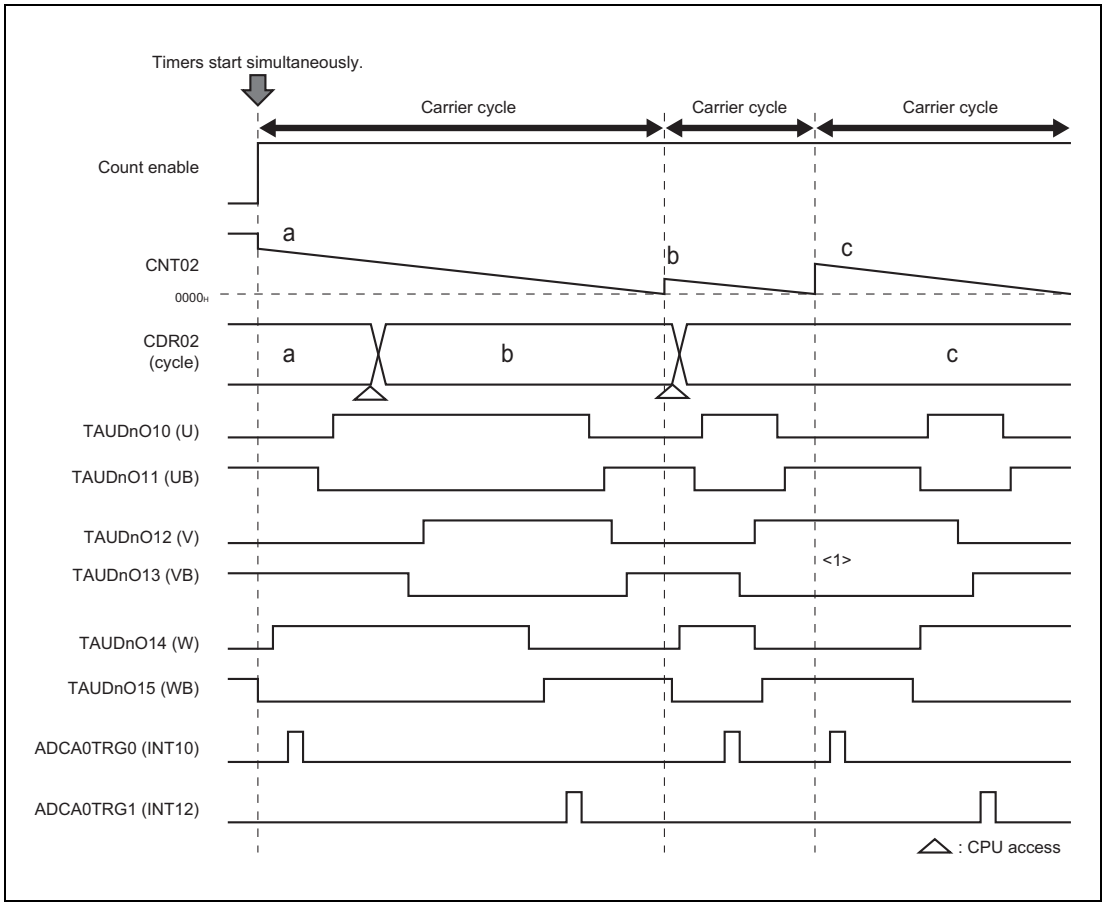


Figure 29.27 Example of Three-Phase PWM (U/UB, V/VB, W/WB) Output with Dead Time

Figure 29.27 shows a typical example of three-phase PWM output with dead time.

By appropriately setting up the set/clear signal output timing, PWM output that extends across carrier cycles (point <1>) and other types of output are also possible.

In this example, ADCA0TRG0 and ADCA0TRG1 (which are at the bottom) use the CNT and INT signals of CH10 and CH12, which are not used for one-phase PWM output, and the A/D trigger signal is output by performing type-1 A/D trigger output.

As shown, the one-phase PWM output function uses only TOUTm that outputs signals on channels that make phase output, thus functions that use CNTm, CDRm, or INTm can be set. For details, see **Section 25, Timer Array Unit D (TAUD)** (m = 10, 12, or 14).

The following figures show timing charts for outputting a three-phase PWM signal with dead time.

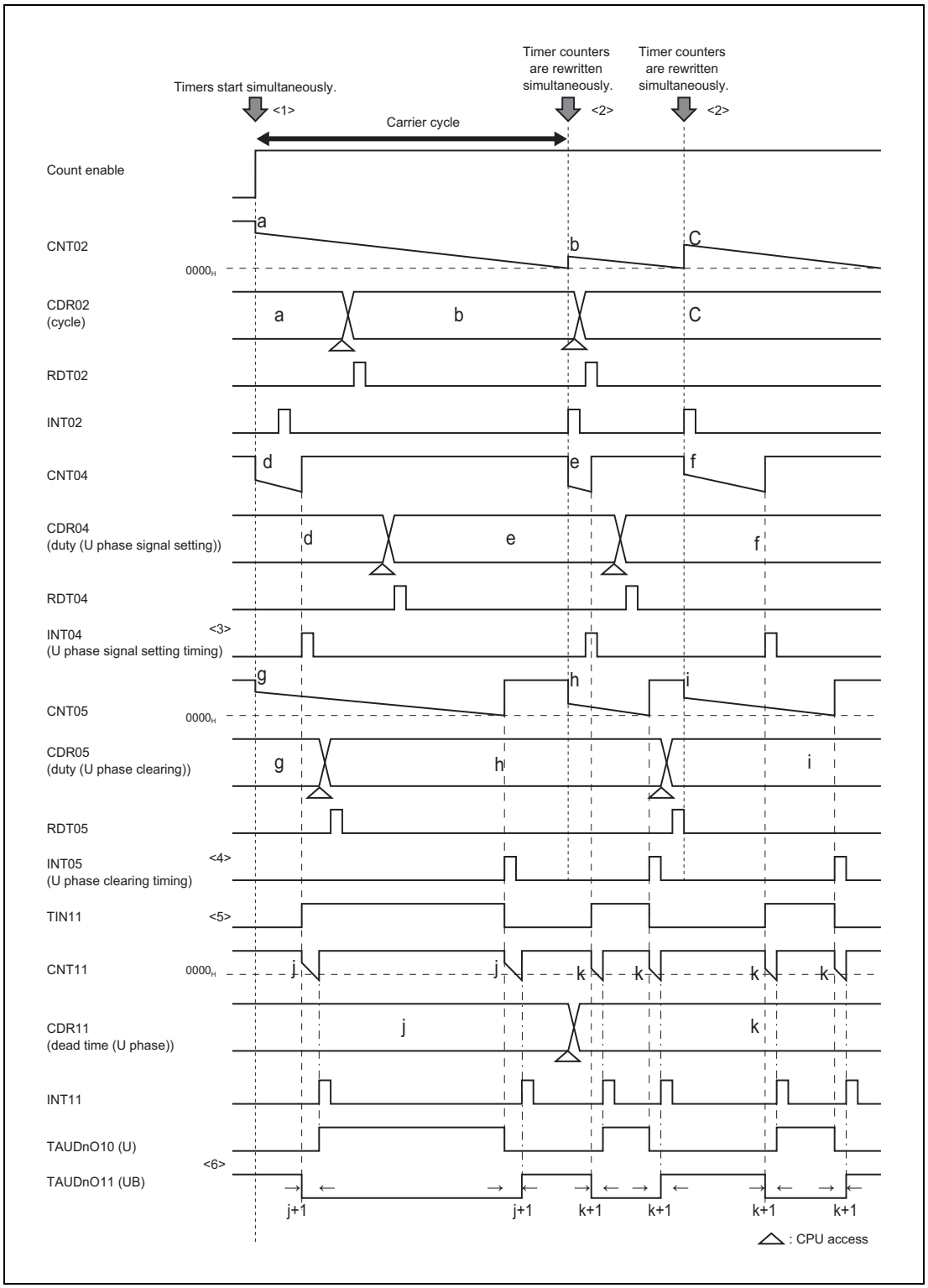


Figure 29.28 Example of One-Phase PWM (U phase, UB phase) Output with Dead Time

An operation example of the timer configuration for performing the U phase PWM output in **Figure 29.28** is provided below.

<1> By simultaneously starting timers, CH2 (the carrier cycle timer), CH4 (the U phase set signal output timing timer), and CH5 (the U phase clear signal output timing timer) are started simultaneously.
The CH11 is also timer-started, but until a TIN11 edge is detected, which is the count start timing, counting is not performed.

<2> For CH4 and CH5, when there is a CH2 underflow, the settings from CDR04 and CDR05 are reloaded to CNT04 and CNT05.

<3> When there is a CH4 underflow, the U phase set timing signal (INT04) is generated.

<4> When there is a CH5 underflow, the U phase clear timing signal (INT05) is generated.

<5> The peripheral interconnections supply the output of the SR flip-flop circuit that uses INT04 (the set timing signal) and INT05 (the clear timing signal) as input to the input TIN11 signal of one-phase PWM output.

<6> During one-phase PWM output, a PWM waveform with dead time is generated and output by detecting a TIN11 edge.

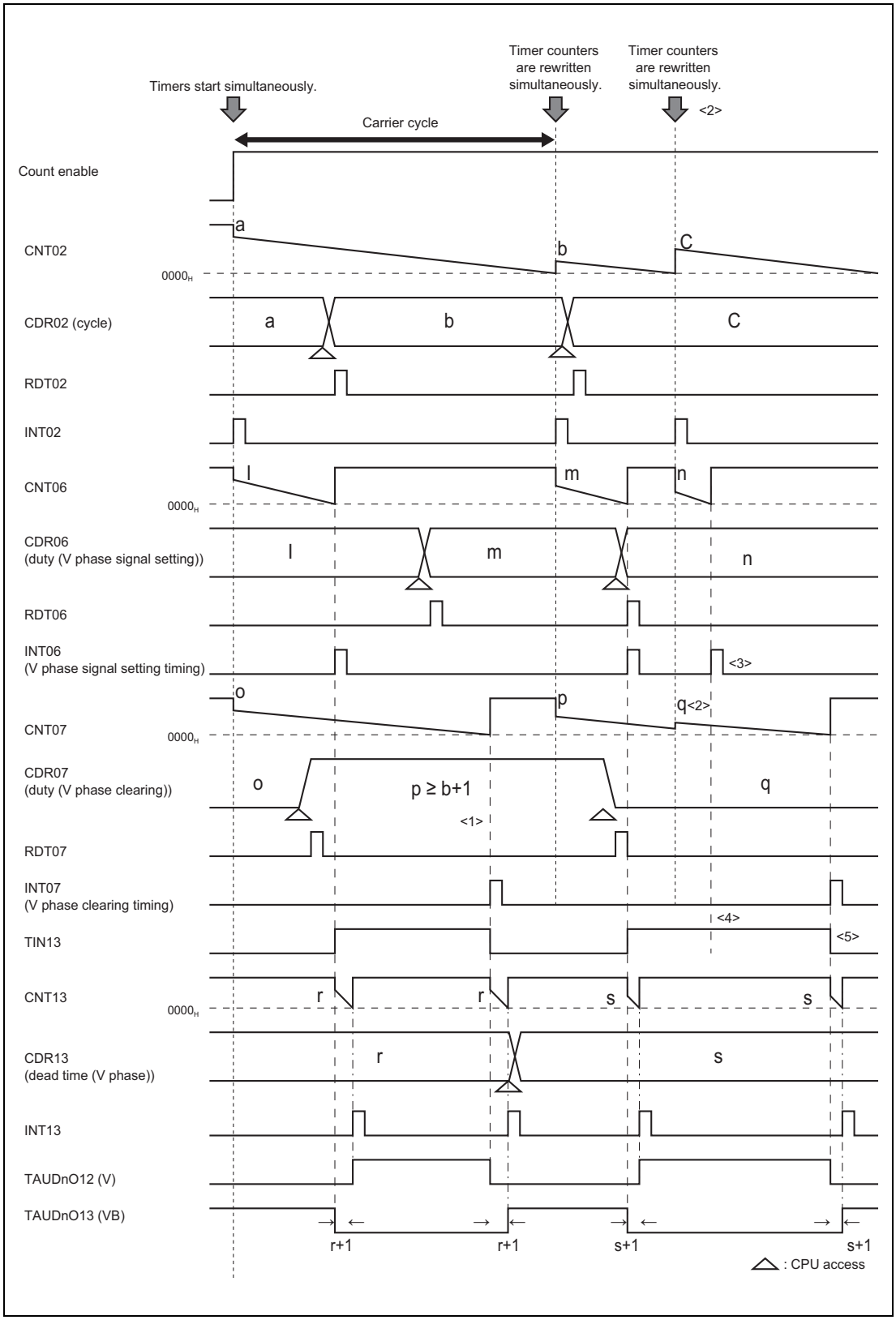


Figure 29.29 Example of One-Phase PWM (V phase, VB phase) Output with Dead Time

An operation example of the timer configuration for performing the V phase PWM output in **Figure 29.29** is provided below.

For details about the operations from when timers are simultaneously started until a one-phase PWM signal is output, see the U phase operation example.

<1> If the setting of CH7 (the V phase clear signal output timing timer), which generates the V phase clear timing signal (INT07), is greater than the CH2 (the carrier cycle timer) setting.

<2> Before a V phase clear timing signal (INT07) is generated by a CH7 underflow, a CH2 (carrier cycle timer) underflow occurs, and the CH7 setting is reloaded.

<3> It causes the V phase clear timing signal (INT07) not to be generated, resulting in consecutive generation of the V phase set timing signal (INT06).

<4> In this case, because the V phase set timing signal (INT06) is ignored by the SR flip-flop circuit, there is no effect on the PWM output waveform. Therefore, a PWM waveform that extends across carrier cycles is output.

<5> The PWM output is changed at the timing of the next V phase clear timing signal (INT07).

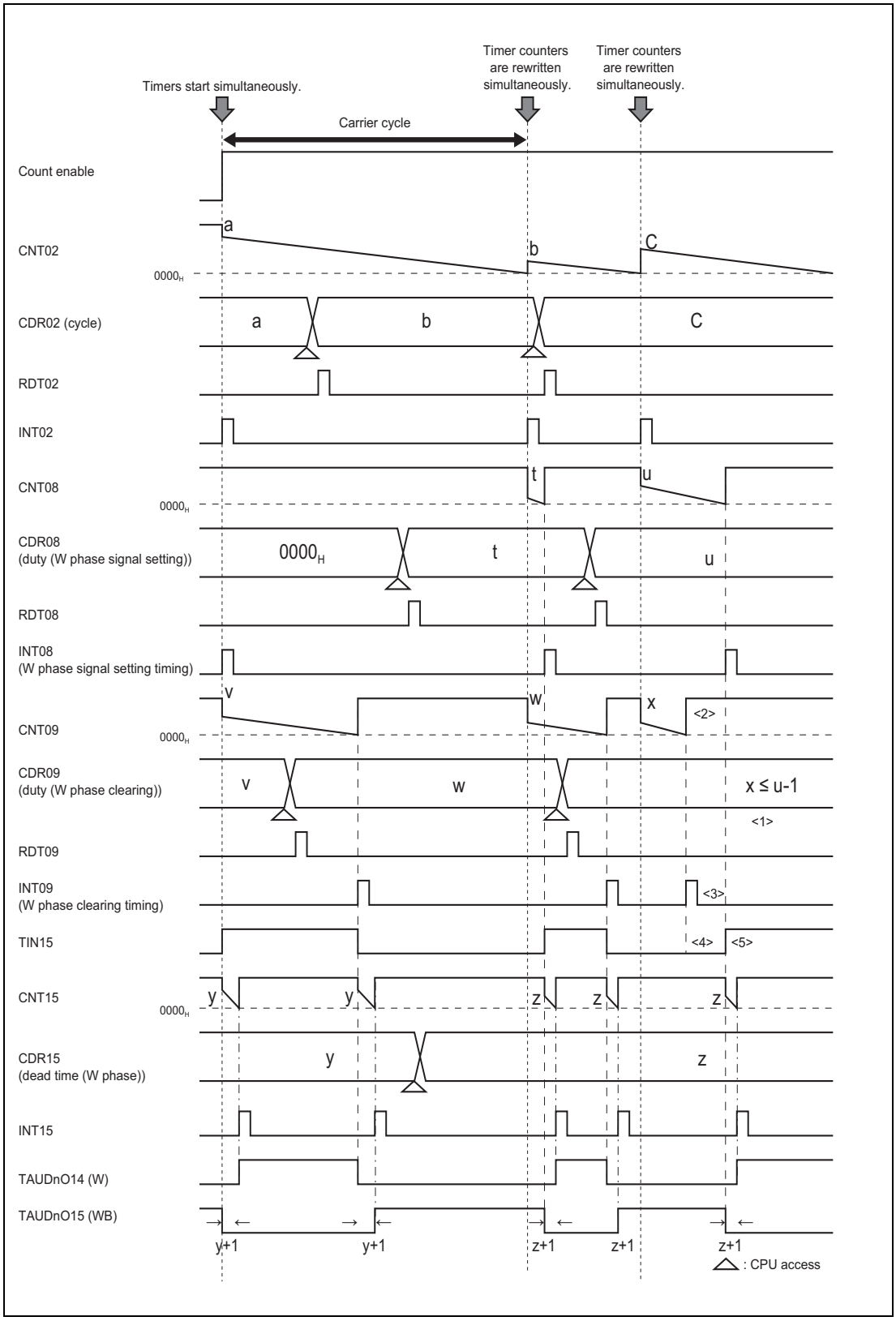


Figure 29.30 Example of One-Phase PWM (W phase, WB phase) Output with Dead Time

An operation example of the timer configuration for performing the W phase PWM output in **Figure 29.30** is provided below.

For details about the operations from when timers are simultaneously started until a one-phase PWM signal is output, see the U phase operation example.

<1> If the setting of CH9 (the W phase clear signal output timing timer), which generates the W phase clear timing signal (INT09), is less than the CH8 (the W phase set signal output timing timer) setting.

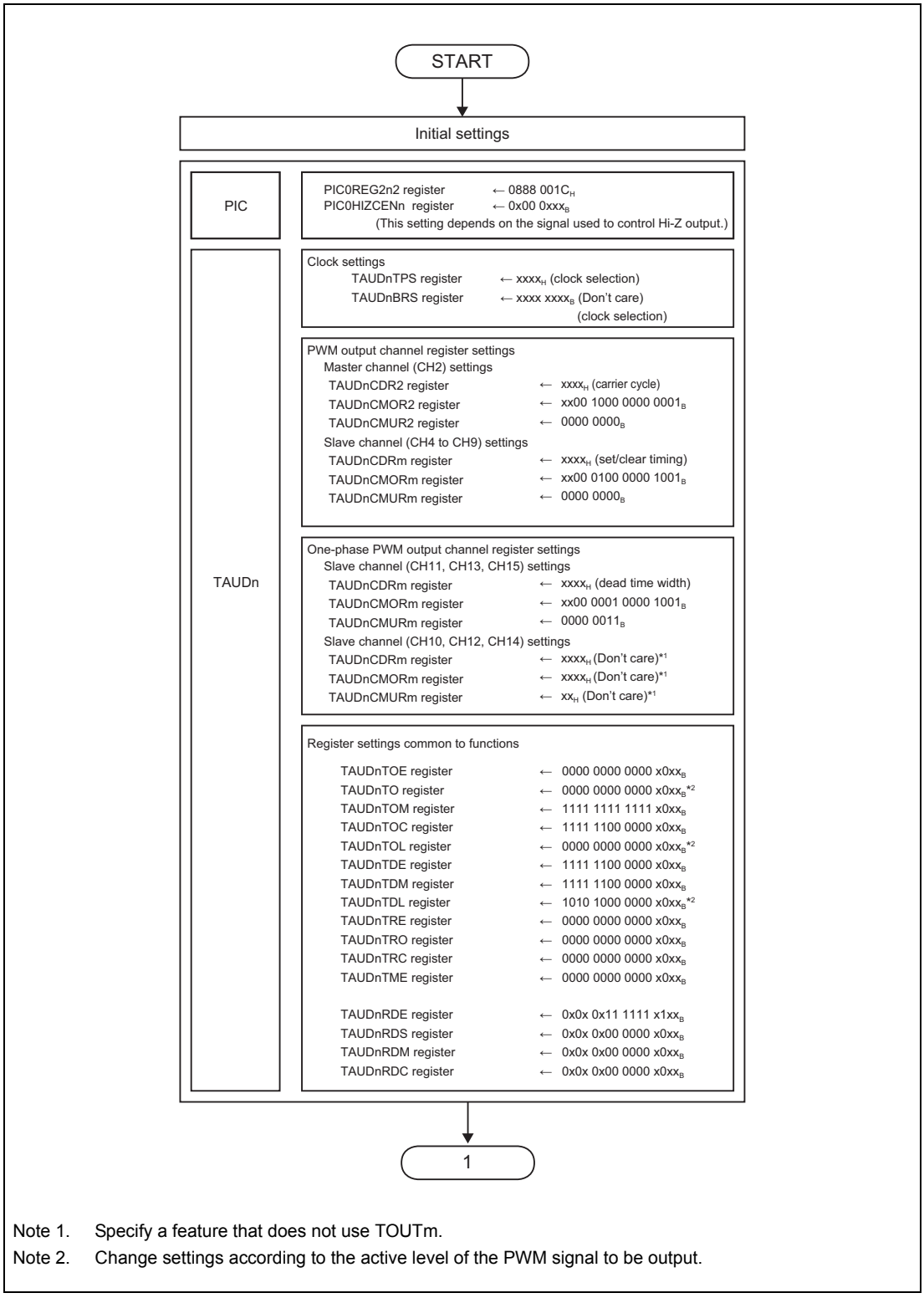
<2> Before a W phase set timing signal (INT08) is generated by a CH8 underflow, a CH9 (W phase clear signal output timing timer) underflow occurs, and the W phase clear timing signal (INT09) is generated.

<3> This results in consecutive W phase clear timing signals (INT09) being generated.

<4> In this case, because the consecutively generated W phase clear timing signals (INT09) are ignored by the SR flip-flop circuit, there is no effect on the PWM output waveform.

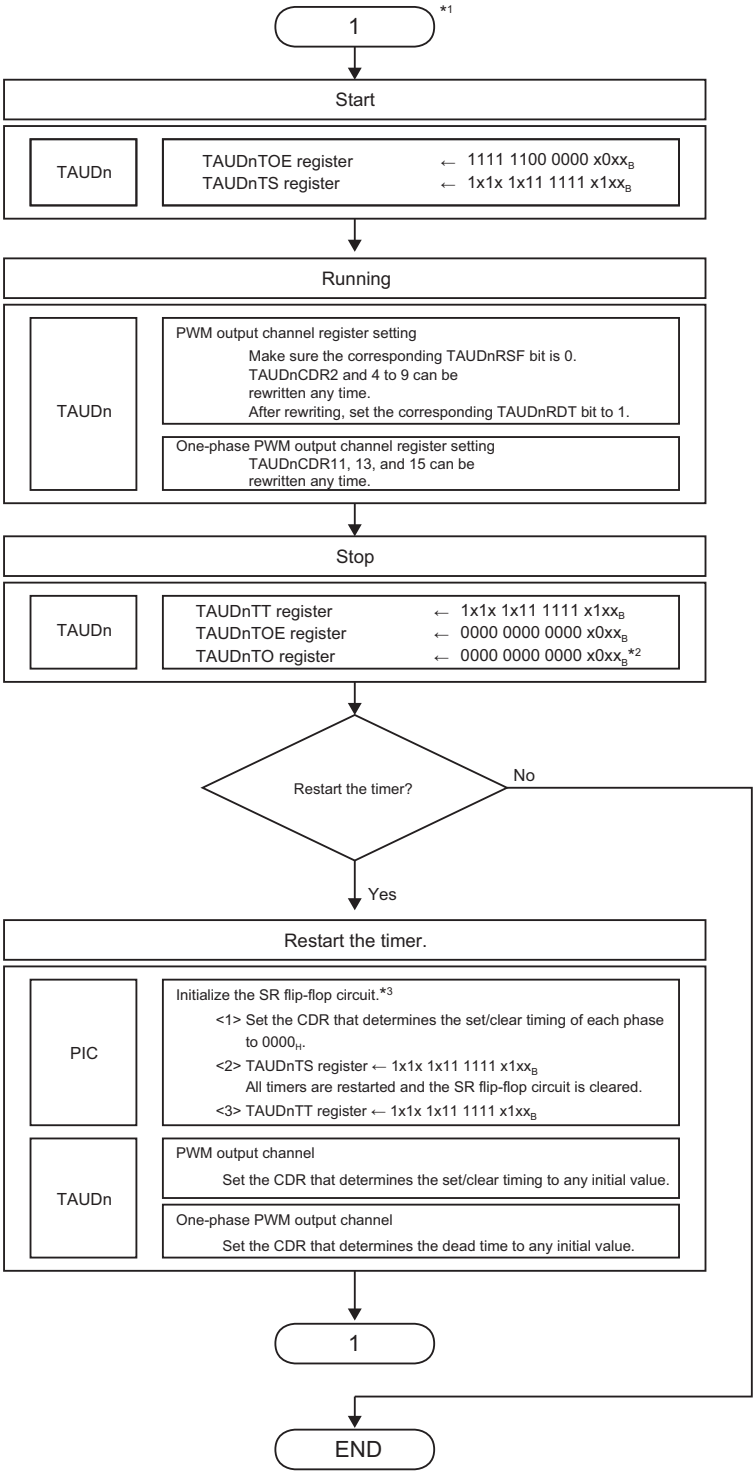
<5> The PWM output is changed at the timing of the next W phase set timing signal (INT08).

29.11.5 Setup Flow



Note 1. Specify a feature that does not use TOUTm.
Note 2. Change settings according to the active level of the PWM signal to be output.

Figure 29.31 Setup Flow (Active High Example)



Note 1. Specify the selection register and output port to use after specifying the initial settings for the peripheral interconnections and timers.

Note 2. Change settings according to the active level of the PWM signal to be output.

Note 3. If initialization processing is not performed, the SR flip-flop circuit enters the timer-stopped state, and the timer restart output pulse might be output at an unintended level.

Figure 29.32 Setup Flow (Active High Example) (continued)

29.11.6 Setting Examples for Operation Functions

This section provides example settings for each register.

29.11.6.1 TAUDn Settings (Active High Example)

Table 29.45 TAUDn: CH2-related (PWM Output Master Channel*1)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	
TAUDnCMUR2	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD PWM output. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be set for the master channel and slave channel.

Table 29.46 TAUDn: CH4 to CH9-related (PWM Output Slave Channel*1) (m = 4 to 9)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Any*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD PWM output. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be set for the master channel and slave channel.

NOTE

For TAUDnCMORm of the PWM output function, only TAUDnCKS[1:0] (operation clock select) can be set to any value, but other control bits are to be fixed values. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Table 29.47 TAUDn: CH11, CH13, and CH15-related (One-phase PWM Output) (m = 11, 13, or 15)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*1	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	001	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	11	Both rising and falling TINm edges are detected as effective. (High width)

Note 1. Specify the same operation clock settings as for the PWM output master channel (CH2).

NOTE

For the TAUDnCMORm register used during one-phase PWM output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. CH10, CH12, and CH14 can be used with any feature that does not use TOUTm output (such as A/D trigger output). For details, see **Section 25, Timer Array Unit D (TAUD)**.

Table 29.48 Common TAUDn Channel Settings (1/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOE	15 to 10	TAUDnTOE15 to TAUDnTOE10	0 1	Disables the timer. Enables the timer.
	9 to 4	TAUDnTOE09 to TAUDnTOE04	0	These are fixed to 0 because TOUT09 to TOUT04 are not used.
	3	TAUDnTOE03	Don't care	
	2	TAUDnTOE02	0	This is fixed to 0 because TOUT02 is not used.
	1, 0	TAUDnTOE01 TAUDnTOE00	Don't care	
TAUDnTO	15 to 10	TAUDnTO15 to TAUDnTO10	0*1	Outputs a low-level signal to TOUT15 to TOUT10.
	9 to 4	TAUDnTO09 to TAUDnTO04	0	Outputs a low-level signal to TOUT09 to TOUT04.
	3	TAUDnTO03	Don't care	
	2	TAUDnTO02	0	Outputs a low-level signal to TOUT02.
	1, 0	TAUDnTO01 TAUDnTO00	Don't care	
TAUDnTOM	15 to 4	TAUDnTOM15 to TAUDnTOM04	1	Synchronous operation mode
	3	TAUDnTOM03	Don't care	
	2	TAUDnTOM02	0	Independent operation mode
	1, 0	TAUDnTOM01 TAUDnTOM00	Don't care	

Table 29.48 Common TAUDn Channel Settings (2/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOC	15 to 10	TAUDnTOC15 to TAUDnTOC10	1	Synchronous operation mode 2
	9 to 4	TAUDnTOC09 to TAUDnTOC04	0	Synchronous operation mode 1
	3	TAUDnTOC03	Don't care	
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01 TAUDnTOC00	Don't care	
TAUDnTOL	15 to 4	TAUDnTOL15 to TAUDnTOL04	0*1	Positive logic output (active high)
	3	TAUDnTOL03	Don't care	
	2	TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01 TAUDnTOL00	Don't care	
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	1	Enables dead time control.*2
	9 to 4	TAUDnTDE09 to TAUDnTDE04	0	Disables dead time control.
	3	TAUDnTDE03	Don't care	
	2	TAUDnTDE02	0	Disables dead time control.
	1, 0	TAUDnTDE01 TAUDnTDE00	Don't care	
TAUDnTDM	15 to 10	TAUDnTDM15 to TAUDnTDM10	1	Outputs dead time upon detecting a TINm input edge at a lower odd channel.
	9 to 4	TAUDnTDM09 to TAUDnTDM04	0	Disabled because dead time control is disabled.
	3	TAUDnTDM03	Don't care	
	2	TAUDnTDM02	0	Disabled because dead time control is disabled.
	1, 0	TAUDnTDM01 TAUDnTDM00	Don't care	
TAUDnTDL	15	TAUDnTDL15	1*1	Operation as the negative segment of the W phase output
	14	TAUDnTDL14	0*1	Operation as the positive segment of the W phase output
	13	TAUDnTDL13	1*1	Operation as the negative segment of the V phase output
	12	TAUDnTDL12	0*1	Operation as the positive segment of the V phase output
	11	TAUDnTDL11	1*1	Operation as the negative segment of the U phase output
	10	TAUDnTDL10	0*1	Operation as the positive segment of the U phase output
	9 to 4	TAUDnTDL09 to TAUDnTDL04	0	Invalid because dead time control is disabled.
	3	TAUDnTDL03	Don't care	
	2	TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01 TAUDnTDL00	Don't care	

Table 29.48 Common TAUDn Channel Settings (3/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTRE	15 to 4	TAUDnTRE15 to TAUDnTRE04	0	Stops real-time output.
	3	TAUDnTRE03	Don't care	
	2	TAUDnTRE02	0	Stops real-time output.
	1, 0	TAUDnTRE01 TAUDnTRE00	Don't care	
TAUDnTRO	15 to 4	TAUDnTRO15 to TAUDnTRO04	0	Invalid because real-time output is disabled.
	3	TAUDnTRO03	Don’t care	
	2	TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01 TAUDnTRO00	Don’t care	
TAUDnTRC	15 to 4	TAUDnTRC15 to TAUDnTRC04	0	Disallow operation as real-time output trigger generation channel.
	3	TAUDnTRC03	Don’t care	
	2	TAUDnTRC02	0	Disallow operation as real-time output trigger generation channel.
	1, 0	TAUDnTRC01 TAUDnTRC00	Don’t care	
TAUDnTME	15 to 4	TAUDnTME15 to TAUDnTME04	0	Disables modulation output for timer output and real-time output.
	3	TAUDnTME03	Don’t care	
	2	TAUDnTME02	0	Disables modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don’t care	
TAUDnRDE	15	TAUDnRDE15	0	Disables simultaneous rewriting.
	14	TAUDnRDE14	Don’t care	
	13	TAUDnRDE13	0	Disables simultaneous rewriting.
	12	TAUDnRDE12	Don’t care	
	11	TAUDnRDE11	0	Disables simultaneous rewriting.
	10	TAUDnRDE10	Don’t care	
	9 to 4	TAUDnRDE09 to TAUDnRDE04	1	Enables simultaneous rewriting.
	3	TAUDnRDE03	Don’t care	
	2	TAUDnRDE02	1	Enables simultaneous rewriting.
	1, 0	TAUDnRDE01 TAUDnRDE00	Don’t care	

Table 29.48 Common TAUDn Channel Settings (4/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnRDS	15	TAUDnRDS15	0	Do not enable simultaneous rewriting by using another upper channel.
	14	TAUDnRDS14	Don't care	
	13	TAUDnRDS13	0	Do not enable simultaneous rewriting by using another upper channel.
	12	TAUDnRDS12	Don't care	
	11	TAUDnRDS11	0	Do not enable simultaneous rewriting by using another upper channel.
	10	TAUDnRDS10	Don't care	
	9 to 4	TAUDnRDS09 to TAUDnRDS04	0	Enables simultaneous rewriting by using a master channel.
	3	TAUDnRDS03	Don't care	
	2	TAUDnRDS02	0	Enables simultaneous rewriting by using a master channel.
	1, 0	TAUDnRDS01 TAUDnRDS00	Don't care	
TAUDnRDM	15	TAUDnRDM15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDM14	Don't care	
	13	TAUDnRDM13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDM12	Don't Care	
	11	TAUDnRDM11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDM10	Don't Care	
	9 to 4	TAUDnRDM09 to TAUDnRDM04	0	Loads the signal when the master channel starts counting.
	3	TAUDnRDM03	Don't Care	
	2	TAUDnRDM02	0	Loads the signal when the master channel starts counting.
	1, 0	TAUDnRDM01 TAUDnRDM00	Don't Care	
TAUDnRDC	15	TAUDnRDC15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDC14	Don't Care	
	13	TAUDnRDC13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDC12	Don't Care	
	11	TAUDnRDC11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDC10	Don't Care	
	9 to 4	TAUDnRDC09 to TAUDnRDC04	0	Do not use this channel to generate the simultaneous rewrite trigger.
	3	TAUDnRDC03	Don't Care	
	2	TAUDnRDC02	1	Do not use this channel to generate the simultaneous rewrite trigger.
	1, 0	TAUDnRDC01 TAUDnRDC00	Don't Care	

Note 1. Change the setting according to the used system.

Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 25, Timer Array Unit D (TAUD)**.

29.11.6.2 PIC Settings

Table 29.49 PIC Settings

Register	Bit Position	Bit Name	Setting	Remark
PIC0REG2n2	27	PIC0REG2n227	1	Selects the input selected by the PIC0REG2n204 bit.
	23	PIC0REG2n223	1	Selects the input selected by the PIC0REG2n203 bit.
	19	PIC0REG2n219	1	Selects the input selected by the PIC0REG2n202 bit.
	4	PIC0REG2n204	1	Selects the set/clear output according to INTTAUDnI8 and INTTAUDnI9.
	3	PIC0REG2n203	1	Selects the set/clear output according to INTTAUDnI6 and INTTAUDnI7.
	2	PIC0REG2n202	1	Selects the set/clear output according to INTTAUDnI4 and INTTAUDnI5.

29.12 High-accuracy Triangle PWM Output with Dead Time

29.12.1 Functional Overview

Compared to the triangle PWM output with dead time of TAUD, this feature makes it possible to control the variable dead time areas near duties of 100% and 0%. This makes more accurate triangle PWM output possible.

With the triangle PWM output with dead time feature of TAUD, it is not possible to output a UB-phase dead time pulse, such as when transitioning to U phase 0% triangular wave output. (See **Figure 29.33**)

For this feature, a pulse is generated in combination with the TAUD timer output, and a pseudo dead time pulse is added.

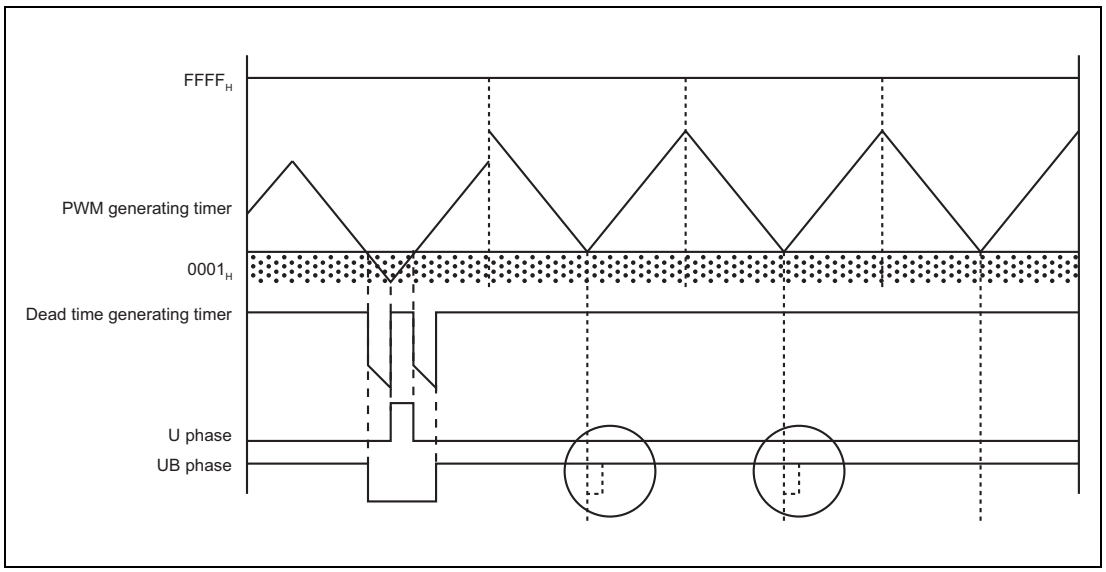


Figure 29.33 Timing of Dead Time Output by the TAUD Feature for Outputting a Triangle PWM Signal with Dead Time

29.12.2 Configuration

The unit and channel configuration for this feature is shown below. (n = 0)

Table 29.50 Configuration of Delay Pulse Output with Dead Time

Timer	Timer Motor Control Function
TAUD0 CH2, CH4 to CH15 (used channels fixed)	TAPA0

The signal names used in the descriptions below are abbreviations. The actual signal names corresponding to each abbreviation are as follows:

- INTm → INTTAUDnIm (TAUDn channel m interrupt)
- TINm → TAUDTTINm (TAUDn channel m input)
- TOUTm → TAUDTTOUTm (TAUDn channel m output)
- CDRm → TAUDnCDRm (TAUDn channel m data register)
- CNTm → TAUDnCNTm (TAUDn channel m counter register)

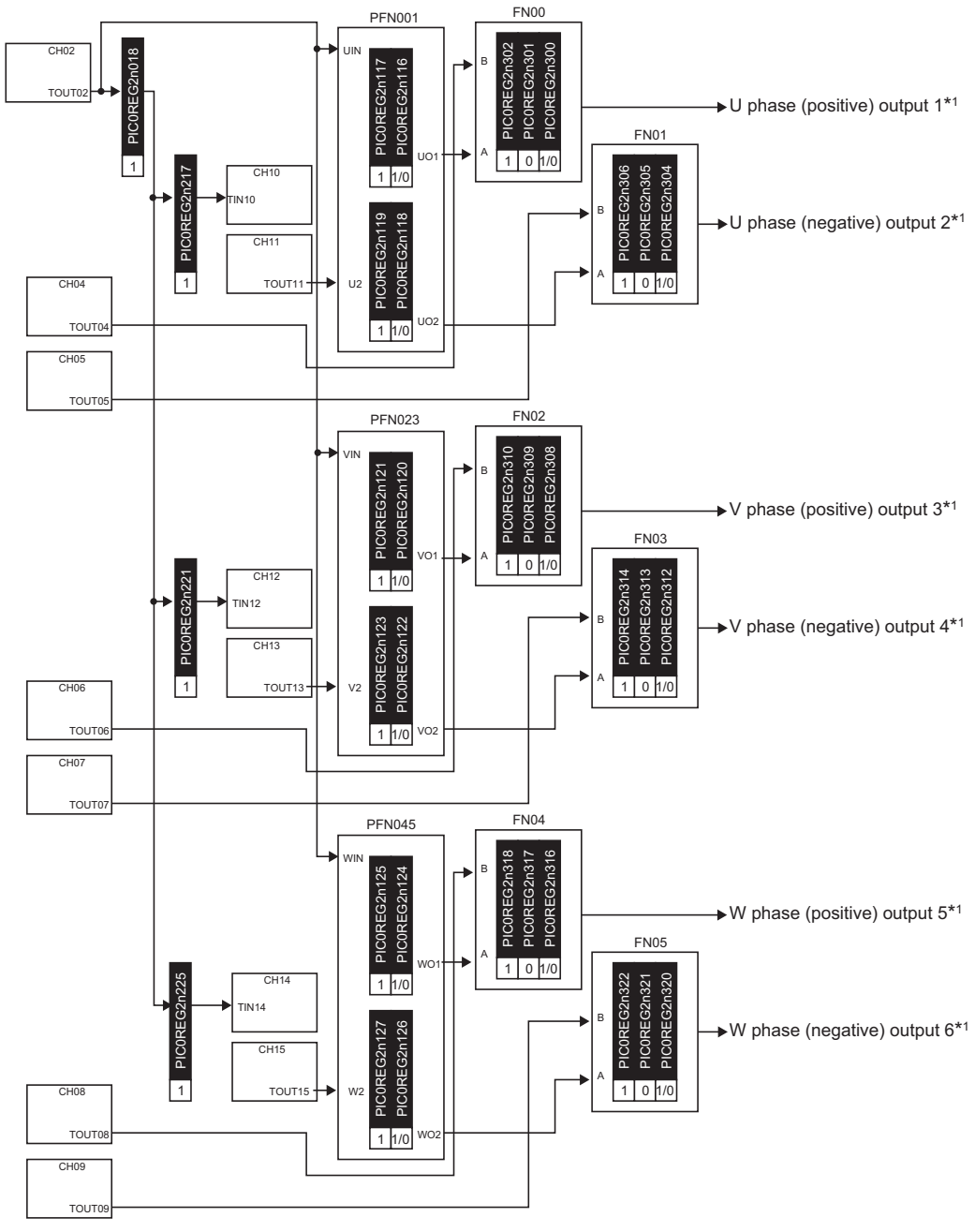
(1) TAUDn configuration

Table 29.51 TAUD Configuration

CH	Function Name	M/S*[1]	CDR Setting	Description
2	Triangle PWM output with dead time (CH2 is the master channel for CH4 to CH9.)	M	Cycle	
4		S	Duty (U phase)	
5		S	Dead time (U phase)	
6		S	Duty (V phase)	
7		S	Dead time (V phase)	
8		S	Duty (W phase)	
9		S	Dead time (W phase)	
10	One-shot pulse output	M	Delay	Generate the pulse to be inserted into the variable dead time area for U phase PWM.
11		S	Pulse width	
12	One-shot pulse output	M	Delay	Generate the pulse to be inserted into the variable dead time area for V phase PWM.
13		S	Pulse width	
14	One-shot pulse output	M	Delay	Generate the pulse to be inserted into the variable dead time area for W phase PWM.
15		S	Pulse width	

Note 1. M: Master channel, S: Slave channel

(2) Block diagram



Note 1. For the connection destination, see **Figure 29.25, Block Diagram: Motor Output Buffer Control**.

Figure 29.34 Block Diagram: High-Accuracy Triangle PWM Output with Dead Time

29.12.3 Registers

29.12.3.1 PIC0REG2n0 — Timer I/O Control Register 2n0 (n = 0)

This register selects TAUDn input.

Access: This register can be read or written in 32-bit units.

Address: PIC0REG200: FFDD 00C0$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	PIC0REG2n018	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.52 PIC0REG2n0 Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	*1
18	PIC0REG2n018	Select the TIN input signal to TAUDTTIN10, TAUDTTIN12, and TAUDTTIN14. 0: Setting prohibited 1: Select TAUDTTOUT2.
17 to 0	Reserved	*1

Note 1. Bits defined as 0 in the PIC0REG2n0 register may be defined by another timer connection function. For such bits, use the bit definition of the timer connection function.

29.12.3.2 PIC0REG2n1 — Timer I/O Control Register 2n1 (n = 0)

This register selects the logic of a combination circuit.

Access: This register can be read or written in 32-bit units.

Address: PIC0REG201: FFDD 00C4$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PIC0REG2n127	PIC0REG2n126	PIC0REG2n125	PIC0REG2n124	PIC0REG2n123	PIC0REG2n122	PIC0REG2n121	PIC0REG2n120	PIC0REG2n119	PIC0REG2n118	PIC0REG2n117	PIC0REG2n116
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.53 PIC0REG2n1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 28	Reserved	*1
27, 26	PIC0REG2n127 PIC0REG2n126	Select the FN05 A input signal according to the output logic specified for CH9 of TAUDn.

PIC0REG2n127	PIC0REG2n126	Input Signal
1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL09 = 0).)
1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL09 = 1).)
Other than the above		Setting prohibited

Bit Position	Bit Name	Function
25, 24	PIC0REG2n125 PIC0REG2n124	Select the FN04 A input signal according to the output logic specified for CH8 of TAUDn.

PIC0REG2n125	PIC0REG2n124	Input Signal
1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL08 = 0).)
1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL08 = 1).)
Other than the above		Setting prohibited

Bit Position	Bit Name	Function
23, 22	PIC0REG2n123 PIC0REG2n122	Select the FN03 A input signal according to the output logic specified for CH7 of TAUDn.

PIC0REG2n123	PIC0REG2n122	Input Signal
1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL07 = 0).)
1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL07 = 1).)
Other than the above		Setting prohibited

Table 29.53 PIC0REG2n1 Register Contents (2/2)

Bit Position	Bit Name	Function
21, 20	PIC0REG2n121 PIC0REG2n120	Select the FN02 A input signal according to the output logic specified for CH6 of TAUDn.
19, 18	PIC0REG2n119 PIC0REG2n118	Select the FN01 A input signal according to the output logic specified for CH5 of TAUDn.
17, 16	PIC0REG2n117 PIC0REG2n116	Select the FN00 A input signal according to the output logic specified for CH4 of TAUDn.
15 to 0	Reserved	*1

PIC0REG2n121	PIC0REG2n120	Input Signal
1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL06 = 0).)
1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL06 = 1).)
Other than the above		Setting prohibited

PIC0REG2n119	PIC0REG2n118	Input Signal
1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL05 = 0).)
1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL05 = 1).)
Other than the above		Setting prohibited

PIC0REG2n117	PIC0REG2n116	Input Signal
1	0	Combination circuit output (Select this when the active high setting is specified (TAUDnTOL04 = 0).)
1	1	Inverted combination circuit output (Select this when the active low setting is specified (TAUDnTOL04 = 1).)
Other than the above		Setting prohibited

Note 1. Bits defined as 0 in the PIC0REG2n1 register may be defined by another timer connection function. For such bits, use the bit definition of the timer connection function.

29.12.3.3 PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0)

Access: This register can be read or written in 32-bit units.

Address: PIC0REG202: FFDD 00C8$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	PIC0REG2n225	—	—	—	PIC0REG2n221	—	—	—	PIC0REG2n217	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.54 PIC0REG2n2 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	*1
25	PIC0REG2n225	Select the TIN input signal to TAUDTTIN14.

PIC0REG2n225	Input signal
1	Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)
Other than the above	Setting prohibited

Bit Position	Bit Name	Function
24 to 22	Reserved	*1
21	PIC0REG2n221	Select the TIN input signal to TAUDTTIN12.

PIC0REG2n221	Input signal
1	Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)
Other than the above	Setting prohibited

Bit Position	Bit Name	Function
20 to 18	Reserved	*1
17	PIC0REG2n217	Select the TIN input signal to TAUDTTIN10.

PIC0REG2n217	Input signal
1	Signal selected by the PIC0REG2n018 bit (TOUT of CH2 of TAUDn)
Other than the above	Setting prohibited

Bit Position	Bit Name	Function
16 to 0	Reserved	*1

Note 1. Bits defined as 0 in the PIC0REG2n2 register may be defined by another timer connection function. For such bits, use the bit definition of the timer connection function.

29.12.3.4 PIC0REG2n3 — Timer I/O Control Register 2n3 (n = 0)

This register selects the logic of a combination circuit.

Access: This register can be read or written in 32-bit units.

Address: PIC0REG203: FFDD 00CC$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	PIC0REG2n322	PIC0REG2n321	PIC0REG2n320	—	PIC0REG2n318	PIC0REG2n317	PIC0REG2n316
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	PIC0REG2n314	PIC0REG2n313	PIC0REG2n312	—	PIC0REG2n310	PIC0REG2n309	PIC0REG2n308	—	PIC0REG2n306	PIC0REG2n305	PIC0REG2n304	—	PIC0REG2n302	PIC0REG2n301	PIC0REG2n300
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 29.55 PIC0REG2n3 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 23	Reserved	*1
22 21 20	PIC0REG2n322 PIC0REG2n321 PIC0REG2n320	Select the logical operation to perform on input signals A and B according to the output logic specified for CH9 of TAUDn.

PIC0REG 2n322	PIC0REG 2n321	PIC0REG 2n320	Input signal
1	0	0	A and B (Select this when the active high setting is specified (TAUDnTOL09 = 0).)
1	0	1	A or B (Select this when the active low setting is specified (TAUDnTOL09 = 1).)
Other than the above			Setting prohibited

19	Reserved	*1
18 17 16	PIC0REG2n318 PIC0REG2n317 PIC0REG2n316	Select the logical operation to perform on input signals A and B according to the output logic specified for CH8 of TAUDn.

PIC0REG 2n318	PIC0REG 2n317	PIC0REG 2n316	Input signal
1	0	0	A and B (Select this when the active high setting is specified (TAUDnTOL08 = 0).)
1	0	1	A or B (Select this when the active low setting is specified (TAUDnTOL08 = 1).)
Other than the above			Setting prohibited

15	Reserved	*1

Table 29.55 PIC0REG2n3 Register Contents (2/2)

Bit Position	Bit Name	Function
14 13 12	PIC0REG2n314 PIC0REG2n313 PIC0REG2n312	Select the logical operation to perform on input signals A and B according to the output logic specified for CH7 of TAUDn.

PIC0REG 2n314	PIC0REG 2n313	PIC0REG 2n312	Input signal
1	0	0	A and B (Select this when the active high setting is specified (TAUDnTOL07 = 0).)
1	0	1	A or B (Select this when the active low setting is specified (TAUDnTOL07 = 1).)
Other than the above			Setting prohibited

Bit Position	Bit Name	Function
11	Reserved	*1
10 9 8	PIC0REG2n310 PIC0REG2n309 PIC0REG2n308	Select the logical operation to perform on input signals A and B according to the output logic specified for CH6 of TAUDn.

PIC0REG 2n310	PIC0REG 2n309	PIC0REG 2n308	Input signal
1	0	0	A and B (Select this when the active high setting is specified (TAUDnTOL06 = 0).)
1	0	1	A or B (Select this when the active low setting is specified (TAUDnTOL06 = 1).)
Other than the above			Setting prohibited

Bit Position	Bit Name	Function
7	Reserved	*1
6 5 4	PIC0REG2n306 PIC0REG2n305 PIC0REG2n304	Select the logical operation to perform on input signals A and B according to the output logic specified for CH5 of TAUDn.

PIC0REG 2n306	PIC0REG 2n305	PIC0REG 2n304	Input signal
1	0	0	A and B (Select this when the active high setting is specified (TAUDnTOL05 = 0).)
1	0	1	A or B (Select this when the active low setting is specified (TAUDnTOL05 = 1).)
Other than the above			Setting prohibited

Bit Position	Bit Name	Function
3	Reserved	*1
2 1 0	PIC0REG2n302 PIC0REG2n301 PIC0REG2n300	Select the logical operation to perform on input signals A and B according to the output logic specified for CH4 of TAUDn.

PIC0REG 2n302	PIC0REG 2n301	PIC0REG 2n300	Input signal
1	0	0	A and B (Select this when the active high setting is specified (TAUDnTOL04 = 0).)
1	0	1	A or B (Select this when the active low setting is specified (TAUDnTOL04 = 1).)
Other than the above			Setting prohibited

Note 1. Bits defined as 0 in the PIC0REG2n3 register may be defined by another timer connection function. For such bits, use the bit definition of the timer connection function.

29.12.3.5 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

This register selects the Hi-Z output control input signal of TAPAn.

Access: This register can be read or written in 8-bit units.

Address: PIC0HIZCEN0: FFDD 0080_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R	R/W

Table 29.56 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Select whether to enable or disable Hi-Z output control by the INTADCA0ERR interrupt signal. 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	Select whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Select whether to enable or disable Hi-Z output control by the TAPAnESO pin input. 0: Disable 1: Enable

29.12.4 Operation Example

This is achieved by combining the following TAUD features:

- Triangle PWM output with dead time
- One-shot pulse output

In addition, the function in PIC is also used because the pulse to be inserted into the variable dead time area is generated for the positive or negative phase:

- Combination circuit (PFN001, PFN023, and PFN045)

In addition, the following function in PIC is used to synthesize the dead time variable pulse to be inserted in the dead time variable area with the triangle PWM output waveform:

- Logical operation circuit (FN0i) (i = 0 to 5)

A high-accuracy triangle PWM signal with dead time is output by assigning the PWM output achieved using the above features to the U, V, and W phases. Therefore, the PWM output dead time can be freely specified for the PWM signal of each phase. Because the only difference among phases is the assigned channel, only one phase (the U phase) is described below.

29.12.4.1 Triangle PWM Output with Dead Time

A triangle PWM signal with dead time is output from TOUT04 and TOUT05 by using CH2, CH4, and CH5 in combination.

29.12.4.2 One-shot Pulse Output

A CDR11 pulse for which the width is delayed by the delay time (CDR10) from the valid edge of the TIN10 (TOUT02) signal of CH10 is output as TOUT11 by using CH10 and CH11 in combination.

This pulse is used as the variable dead time area pulse used near duties of 100% and 0%.

CAUTION

Specify each CDR setting for one-shot pulse output such that the following condition is satisfied: CDR05 ≥ (CDR10 + CDR11)

If a value that does not satisfy the above condition is specified, the output waveform might be affected. To minimize this effect, in addition to satisfying the above setting condition, leave CDR11 set to 0000_H until the variable dead time area pulse is required. Detect both rising and falling edges as the valid TIN10 (TOUT02) edge, and set TAUDnTOL11 to 1 (active low).
Specify the same operation clock for each TAUDn channel used for outputting a triangle PWM signal with dead time or a one-shot pulse.

For details about the TAUD functions, see **Section 25, Timer Array Unit D (TAUD)**.

29.12.4.3 U phase Combination Circuit (PFN001)

This circuit generates a variable dead time area pulse (FN00 A, FN01 A) for adding a generated one-shot pulse to a generated triangle PWM signal with dead time.

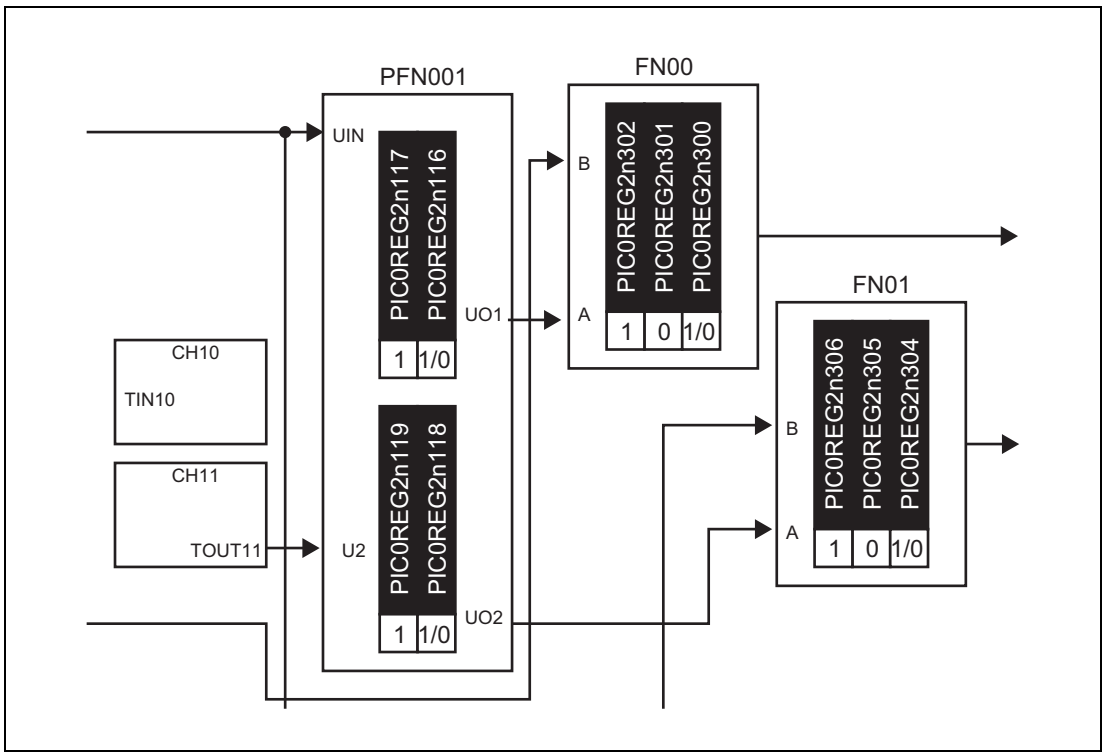


Figure 29.35 Block Diagram Excerpt (PFN001, FN00, and FN01)

The table below shows the relationships between combination circuit input (UIN, U2) and output (UO1, UO2).

Table 29.57 U and UB Phase Combination Circuit (PFN001) I/O Table

- **UO1 (U phase variable dead time area pulse) output**

UIN (TOUT02)	U2 (TOUT11)	UO1	
		PIC0REG2n117, 16 = 10_B U phase output active high (TAUDnTOL04 = 0)	PIC0REG2n117, 16 = 11_B U phase output active low (TAUDnTOL04 = 1)
0	0	1	0
0	1	1	0
1	0	0	1
1	1	1	0

- **UO2 (UB-phase variable dead time area pulse) output**

UIN (TOUT02)	U2 (TOUT11)	UO2	
		PIC0REG2n119, 18 = 10_B UB-phase output active high (TAUDnTOL05 = 0)	PIC0REG2n119, 18 = 11_B UB-phase output active low (TAUDnTOL05 = 1)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

NOTE

The PIC0REG2n116, PIC0REG2n117, PIC0REG2n118, and PIC0REG2n119 settings change depending on the active U phase and UB-phase levels of the generated triangle PWM signal with dead time.

29.12.4.4 Logical Operation Circuit (FN0i) (i = 0 or 1)

This circuit combines an output triangle PWM signal with dead time (TOUT04, TOUT05) with combination circuit output (UO1 and UO2 of PFN001) and generates a PWM signal to which a variable dead time area pulse has been added.

The combination logic for the logical operation circuit is switched according to the PIC0REG2n3 register setting. (Bits 0 to 2 are specified for U phase output, and bits 4 to 6 are specified for UB-phase output.)

Set up the logical operation circuit as shown in the table below. The combined signal is output from the TAPAnUP and TAPAnUM pins according to the specified combination logic.

Table 29.58 Logical Operation Circuit (FN0i) (i = 0 or 1) Settings and TAPAnUP and TAPAnUM Pin Output

- **U phase output (TOUT04)**

Active level	PIC0REG2n302 to 00	TAPAnUP pin output waveform
Active high (TAUDnTOL04 = 0)	100_B	AND of FN00 B (TOUT04) and FN00 A (UO1)
Active low (TAUDnTOL04 = 1)	101_B	OR of FN00 B (TOUT04) and FN00 A (UO1)

- **UB-phase output (TOUT05)**

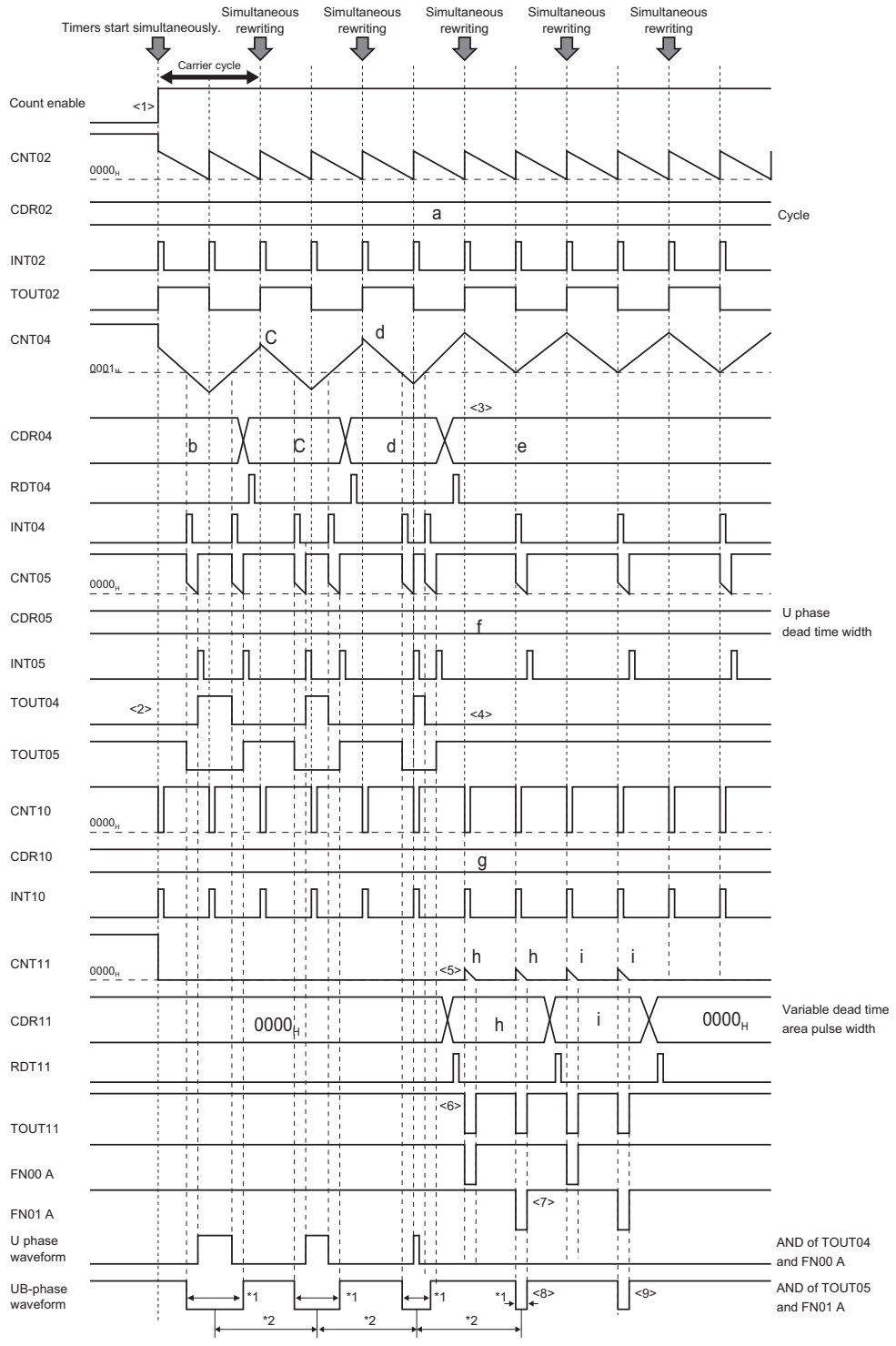
Active level	PIC0REG2n306 to 04	TAPAnUM pin output
Active high (TAUDnTOL05 = 0)	100_B	AND of FN01 B (TOUT05) and FN01 A (UO2)
Active low (TAUDnTOL05 = 1)	101_B	OR of FN01 B (TOUT05) and FN01 A (UO2)

Because the above makes variable dead time control possible to ensure output accuracy near duties of 0% and 100% even for TAUD, a more accurate triangle PWM signal can be output than that output using the TAUD feature for outputting a triangle PWM signal with dead time.

For the V/VB phase and W/WB phase, the used channels and register bits differ, but the settings are the same, as shown in **Figure 29.34, Block Diagram: High-Accuracy Triangle PWM Output with Dead Time.**

The peripheral interconnections provide a connection for adding the pulse generated during one-shot pulse output to the PWM signal generated during output of a triangle PWM signal with dead time by using the combination circuit and logical operation circuit of the peripheral interconnections.

The following figures show timing charts for outputting a high-accuracy triangle PWM signal with dead time.



Note 1. The variable dead time area pulse uses a sawtooth wave and is therefore expanded on one side, unlike a pulse that uses a triangle wave, which is expanded on both sides.

Note 2. Because the variable dead time area pulse is expanded on one side, the length of the one-phase PWM signal output cycle for the variable dead time area increases by 1/2 the added variable dead time area pulse width.

Figure 29.36 Example of a High-Accuracy PWM Signal Output with Dead Time (U phase: 0%, UB-Phase: 100%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))

An operation example in which the system transitions to a U phase of 0% and UB-phase of 100% in the timer configuration for performing the U phase PWM output shown in **Figure 29.36** is provided below. Output of a triangle PWM signal with dead time is active high.

<1> When timer operation is started, output of a triangle PWM with dead time is started by the CH2, CH4, and CH5 channels of TAUDn.

<2> A triangle PWM waveform with dead time is generated from TOUT04 and TOUT05.

<3> A U phase duty output value of 0% is specified for CDR04.

<4> Due to the setting in <3>, TOUT04 output is the inactive level, and TOUT05 output is the active level. However, no variable dead time area pulse is output during this operation.

<5> To create a variable dead time area pulse, the value to be used as the pulse width is specified for CDR11 when specifying the 0% U phase duty in <3>.
For this example, the CDR11 setting is fixed to 0000_H until the system enters the variable dead time area to prevent adverse effects on the output PWM signal.

<6> The variable dead time area pulse is output as a pulse that has the width specified for CDR11 after the delay time specified for CDR10 elapses, starting at the TOUT02 edge.

<7> The pulse output in <6> is converted to a variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A) by the combination circuit (PFN001).

<8> The pulse generated in <7> is combined with the TOUT04 and TOUT05 output waveforms by using the logical operation circuits (FN00, FN01), and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB-phase output).

<9> By later changing the CDR11 setting, which specifies the width of the variable dead time area pulse, the desired variable dead time area pulse can be added.

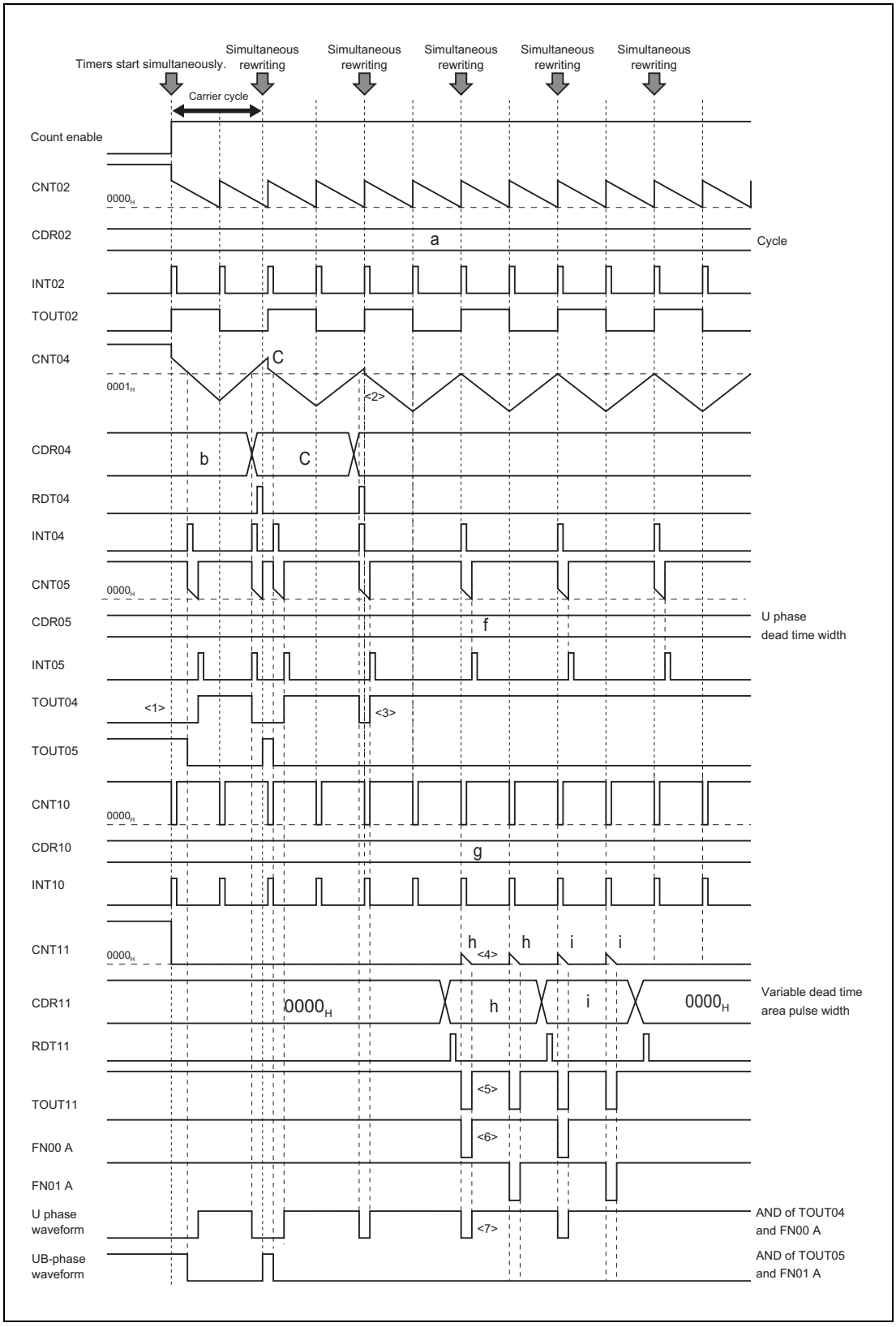


Figure 29.37 Example of a High-Accuracy PWM Signal Output with Dead Time (U phase: 100%, UB-Phase: 0%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))

An operation example in which the system transitions to a U phase of 100% and UB-phase of 0% in the timer configuration for performing the U phase PWM output shown in **Figure 29.37** is provided below. Output of a triangle PWM signal with dead time is active high.

<1> The timer operation from the start of timer operation until the output of a triangle PWM signal with dead time is the same.

<2> A U phase duty output value of 100% (CDR04 = 0000_H) is specified for CDR04.

<3> Due to the setting in <2>, TOUT04 output is the active level, and TOUT05 output is the inactive level. However, no variable dead time area pulse is output during this operation.

<4> To create a variable dead time area pulse, the value to be used as the pulse width is specified for CDR11 one cycle after specifying the 100% U phase duty setting in <2>.
For this example, the CDR11 setting is fixed to 0000_H until the system enters the variable dead time area to prevent adverse effects on the output PWM signal.

<5> The variable dead time area pulse is output as a pulse that has the width specified for CDR11 after the delay time specified for CDR10 elapses, starting at the TOUT02 edge.

<6> The pulse output in <5> is converted to a variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A) by the combination circuit (PFN001).

<7> The pulse generated in <6> is combined with the TOUT04 and TOUT05 output waveforms by using the logical operation circuits (FN00, FN01), and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB-phase output).

CAUTION

If the 100% U phase duty setting for CDR04 and the variable dead time area pulse width for CDR11 are specified at the same time, the variable dead time area pulse affects by the amount shown by <2> for the last PWM signal output from TOUT04 and shown by feature specification <1>, as shown in Figure 29.38.
To cancel this effect, the CDR11 setting is delayed one cycle.

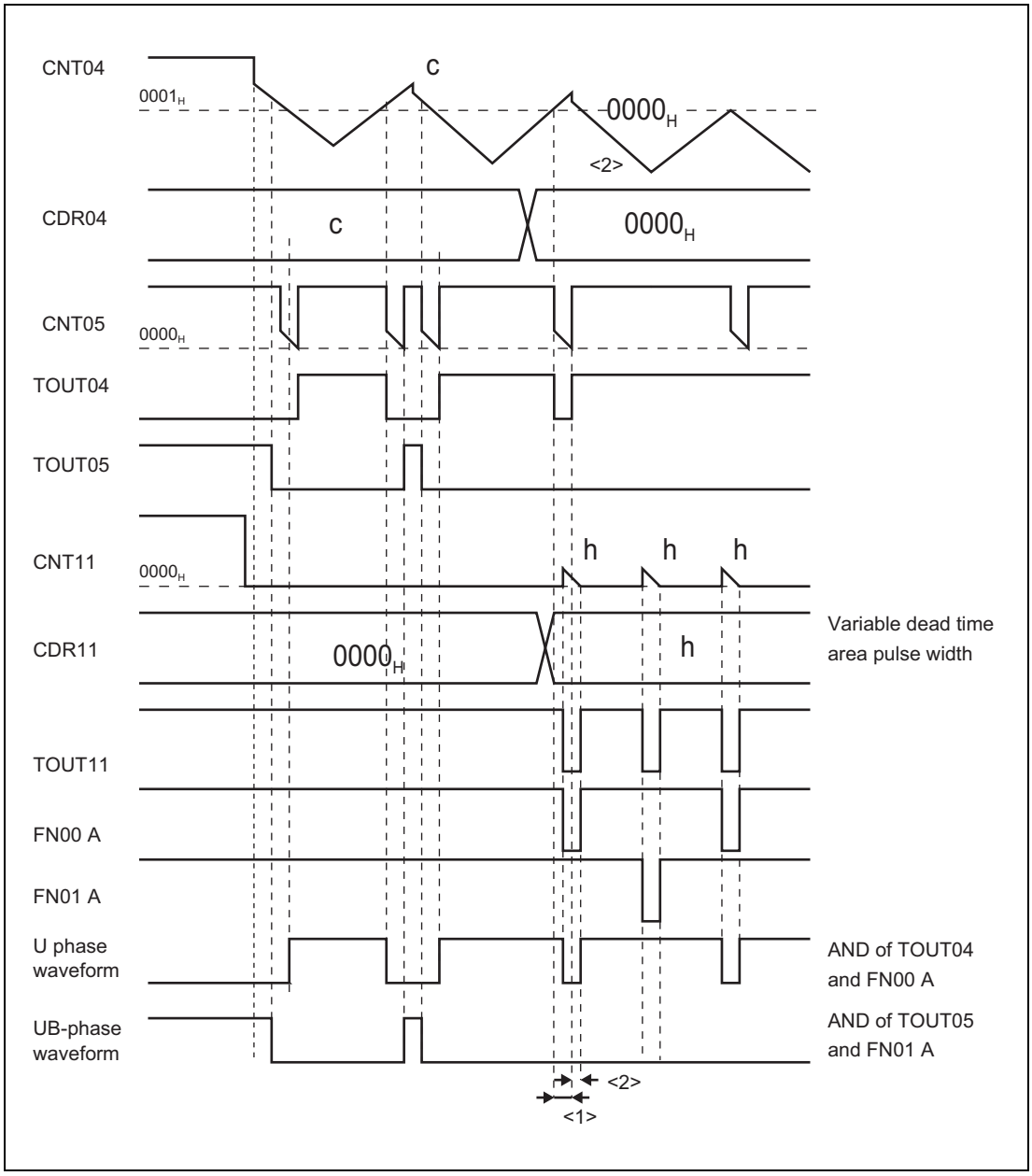


Figure 29.38 Effect on the Output Triangle PWM Wave with Dead Time by the Variable Dead Time Area Pulse

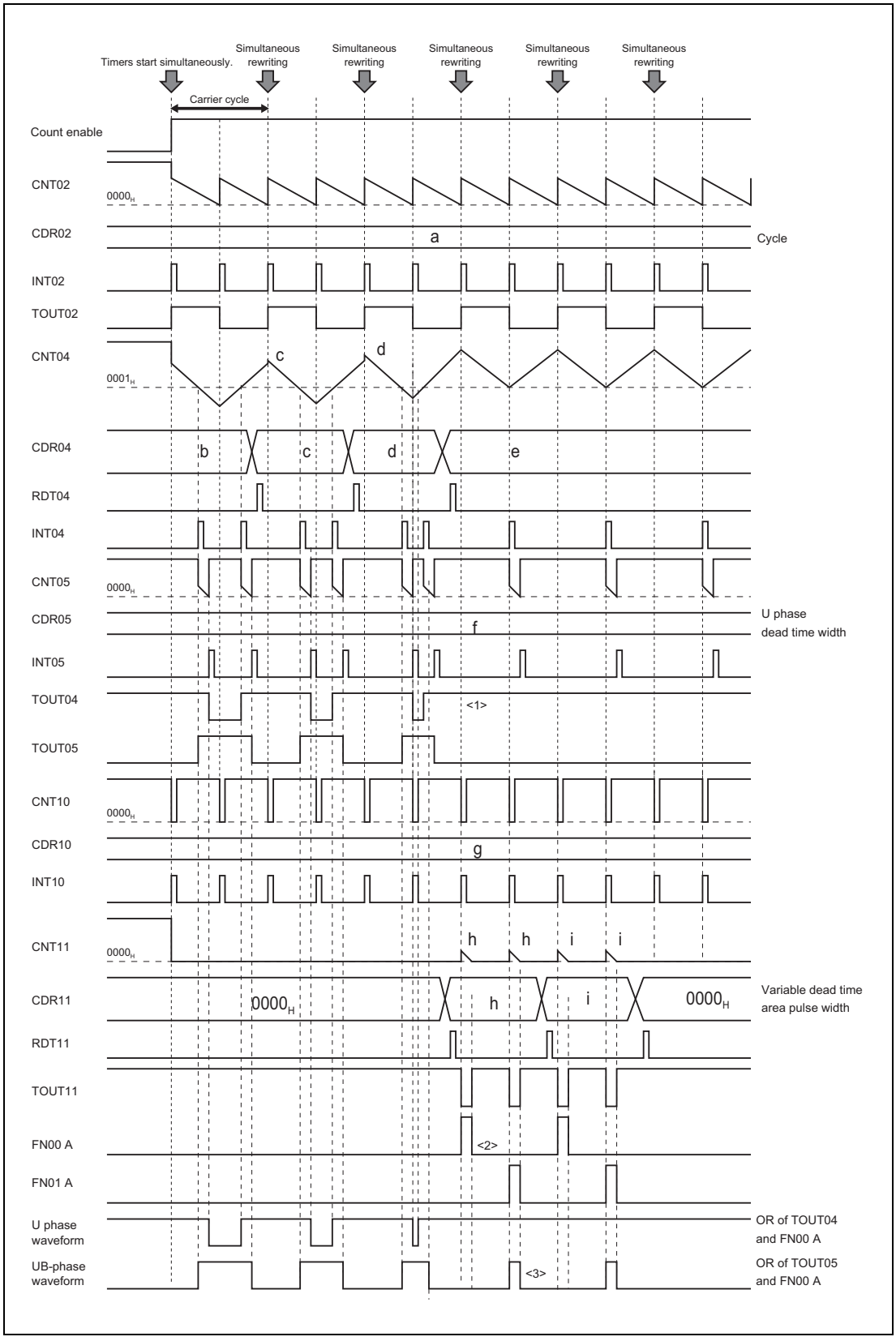


Figure 29.39 Example of a High-Accuracy PWM Signal Output with Dead Time (U phase: 100%, UB-Phase: 0%) (TAUDnTOL04 = 1 (Active Low), TAUDnTOL05 = 1 (Active Low))

An operation example in which the system transitions to a U phase of 100% and UB-phase of 0% in the timer configuration for performing the U phase PWM output shown in **Figure 29.39** is provided below. Output of a triangle PWM signal with dead time is active low.

<1> The timer operation from the start of timer operation until the output of a triangle PWM signal with dead time is the same as in **Figure 29.36, Example of a High-Accuracy PWM Signal Output with Dead Time (U phase: 0%, UB-Phase: 100%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))**. However, an active low PWM signal is output from TOUT04 and TOUT05.

<2> Therefore, active low output that corresponds with PWM output is specified as the combination circuit setting (PIC0REG2n116 and PIC0REG2n117, and PIC0REG2n118 and PIC0REG2n119). This results in the output of an active low variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A).

<3> In addition, active low output that corresponds with PWM output is also specified as the logical operation circuit setting (PIC0REG2n302 to PIC0REG2n300 and PIC0REG2n306 to PIC0REG2n304). The pulse generated in <2> is combined with the TOUT04 and TOUT05 output waveforms, and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB-phase output) as an active low PWM signal.

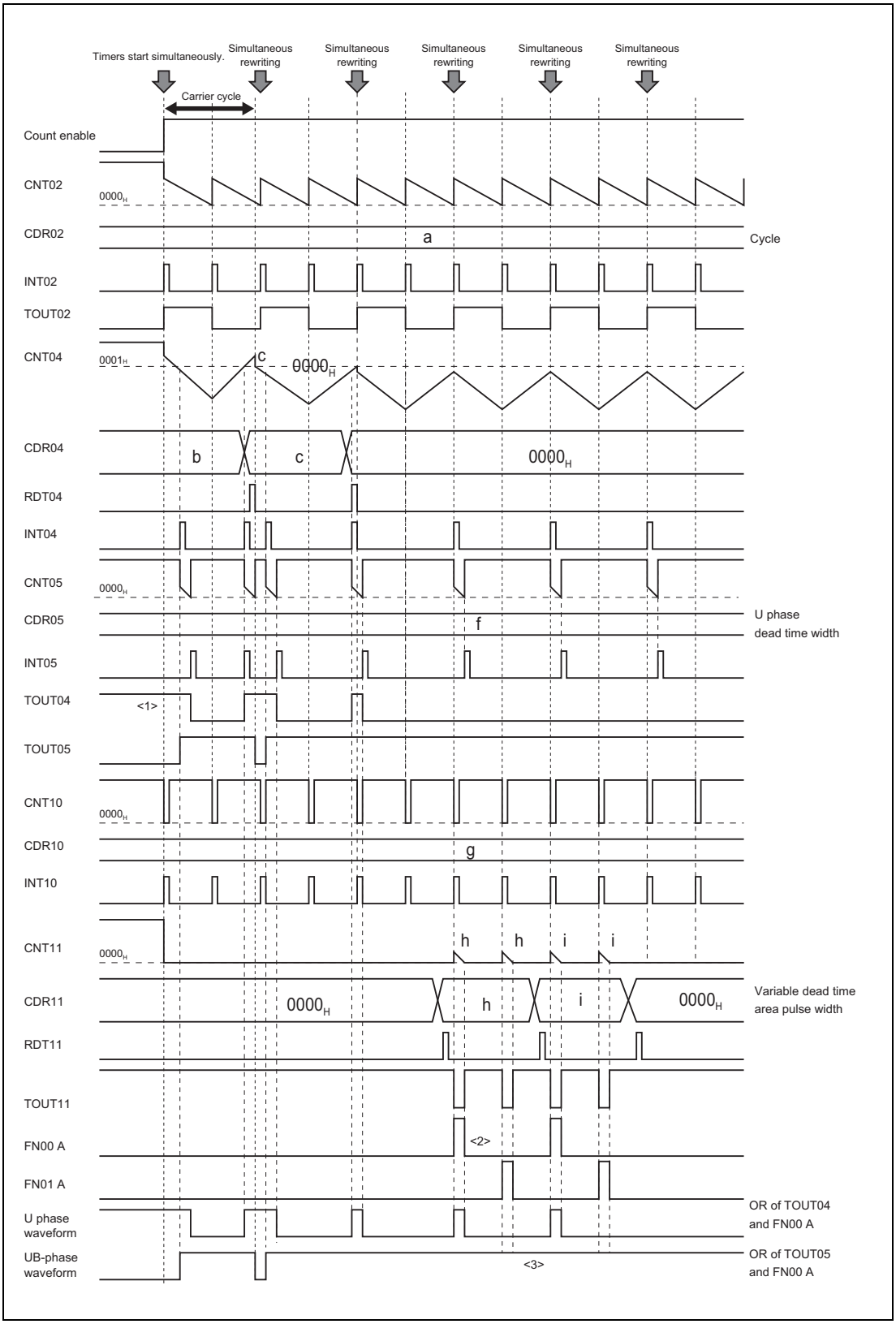


Figure 29.40 Example of a High-Accuracy PWM Signal Output with Dead Time (U phase: 0%, UB-Phase: 100%) (when TAUDnTOL04 = 0 (Active Low) and TAUDnTOL05 = 0 (Active Low))

An operation example in which the system transitions to a U phase of 0% and UB-phase of 100% in the timer configuration for performing the U phase PWM output shown in **Figure 29.40** is provided below. Output of a triangle PWM signal with dead time is active low.

<1> The timer operation from the start of timer operation until the output of a triangle PWM signal with dead time is the same as in **Figure 29.37, Example of a High-Accuracy PWM Signal Output with Dead Time (U phase: 100%, UB-Phase: 0%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))**. However, an active low PWM signal is output.

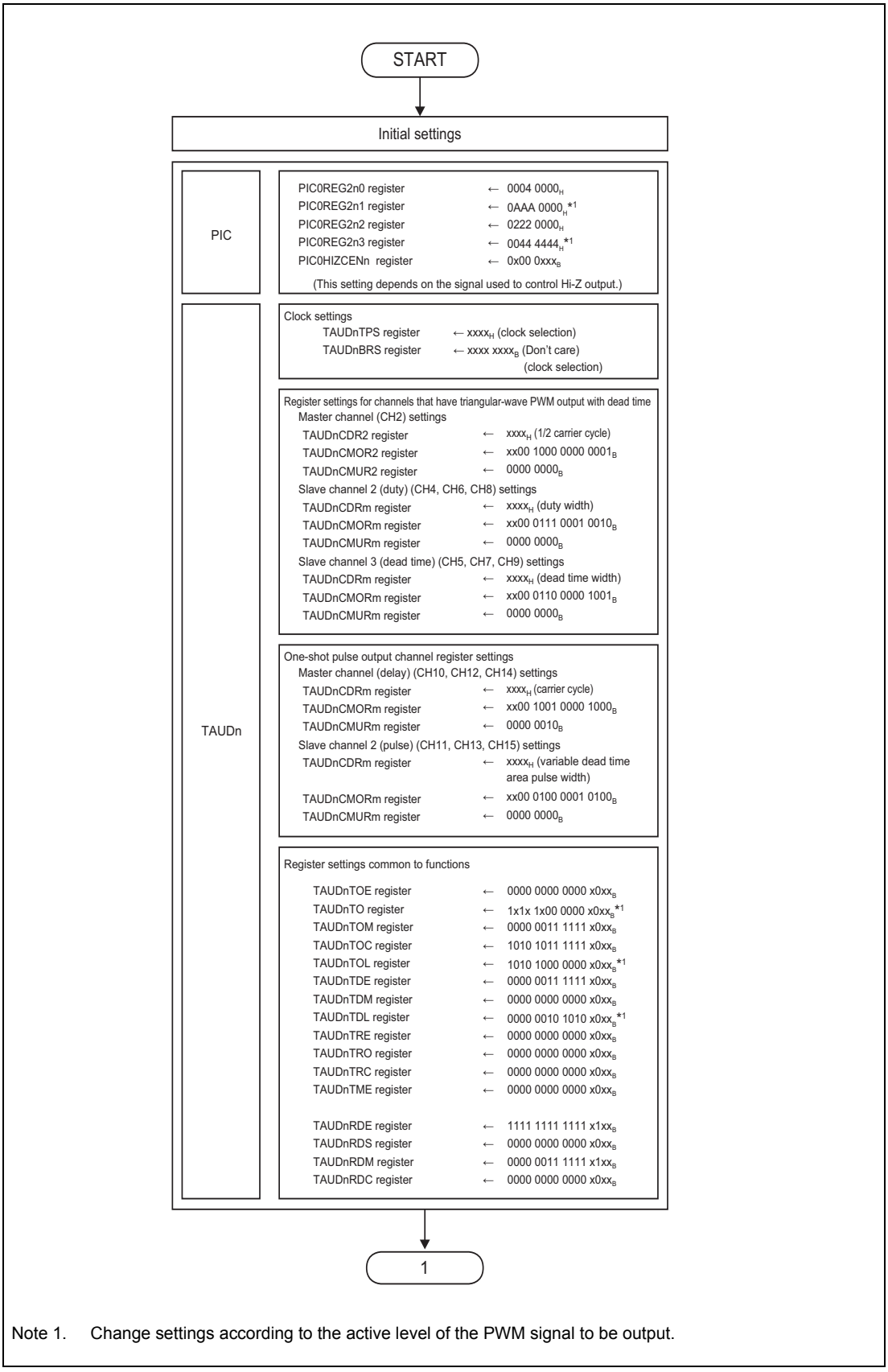
<2> Therefore, active low output that corresponds with PWM output is specified as the combination circuit setting (PIC0REG2n116 and PIC0REG2n117, and PIC0REG2n118 and PIC0REG2n119). This results in the output of an active low variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A).

<3> In addition, active low output that corresponds with PWM output is also specified as the logical operation circuit setting (PIC0REG2n302 to PIC0REG2n300 and PIC0REG2n306 to PIC0REG2n304). The pulse generated in <2> is combined with the TOUT04 and TOUT05 output waveforms, and the result is output from TAPAnUP (U phase output) and TAPAnUM (UB-phase output) as an active low PWM signal.

CAUTION

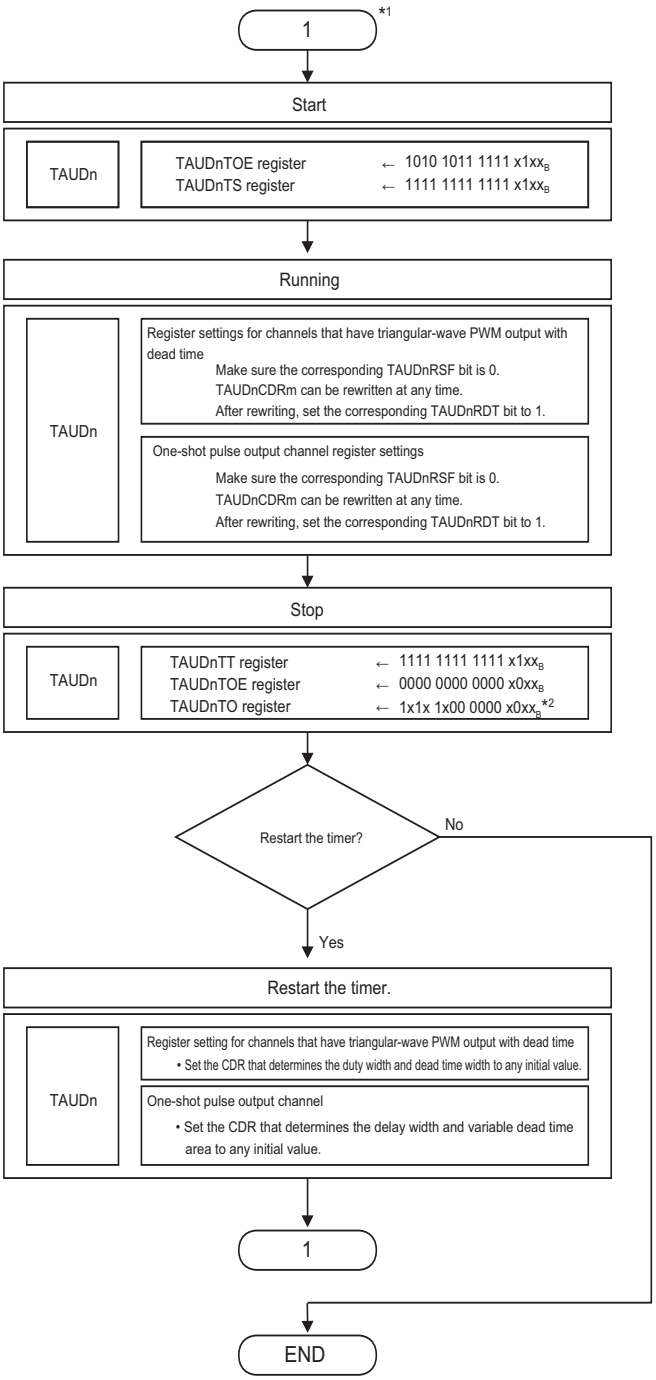
If the 100% U phase duty setting for CDR04 and the variable dead time area pulse width for CDR11 are specified at the same time, the last PWM signal output from TOUT04 is adversely affected due to the feature specifications.
To cancel this effect, the CDR11 setting is delayed one cycle.
For details, see Figure 29.38, Effect on the Output Triangle PWM Wave with Dead Time by the Variable Dead Time Area Pulse.

29.12.5 Setup Flow



Note 1. Change settings according to the active level of the PWM signal to be output.

Figure 29.41 Setup Flow (Active High Example)



Note 1. Specify the selection register and output port to use after specifying the initial settings for the peripheral interconnections and timers.

Note 2. Change settings according to the active level of the PWM signal to be output.

Figure 29.42 Setup Flow (Active High Example) (continued)

29.12.6 Setting Examples for Operation Functions

This section provides example settings for each register.

29.12.6.1 TAUDn settings (active high example)

Table 29.59 TAUDn: CH2-related (Master Channel Used To Output A Triangle PWM Signal with Dead Time*1)

Register	Bit position	Bit name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	At the start of operation, output INTm and toggle TOUTm.
TAUDnCMUR2	1, 0	TAUDnTIS[1:0]	00	Fixed

Note 1. The master channel and slave channel names are defined for TAUD triangle PWM output with dead time. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

NOTE

For the TAUDnCMORm register of the master channel used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) and TAUDnMD0 can be set to any value, but other control bits have fixed values. For details, see **Section 25, Timer Array Unit D (TAUD)**.
For this feature, set TAUDnMD0 to 1.

Table 29.60 TAUDn: CH4, CH6, and CH8-related (Slave Channel 2 used to Output a Triangle PWM Signal with Dead Time*1) (m = 4, 6, or 8)

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	111	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	1001	
	0	TAUDnMD0	0	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The same operation clock must be specified for the master channel and slave channel.
For the TAUDnCMORm register of slave channels 2 and 3, which is used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 29.61 TAUDn: CH5, CH7, and CH9-related (Slave Channel 3 used to Output a Triangle PWM Signal with Dead Time*1) (m = 5, 7, or 9)

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	110	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The same operation clock must be specified for the master channel and slave channel.
For the TAUDnCMORm register of slave channels 2 and 3, which is used when outputting a triangle PWM signal with dead time, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 29.62 TAUDn: CH10, CH12, and CH14-related (Master Channel used to Output a One-shot Pulse*1) (m = 10, 12, or 14)

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	001	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	0	Disables start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	10	Detects both rising and falling edges as valid.

Note 1. The master channel and slave channel names are defined for TAUD one-shot pulse output. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 29.63 TAUDn: CH11, CH13, and CH15-related (Slave Channel used to Output a One-Shot Pulse*1) (m = 11, 13, or 15)

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	
	4 to 1	TAUDnMD[4:1]	1010	
	0	TAUDnMD0	0	Disables start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD one-shot pulse output. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel. Specify the same clock setting as for the master channel (CH2) used to output a triangle PWM signal with dead time.

NOTE

For the TAUDnCMORm register used during one-shot pulse output, TAUDnCKS[1:0] (which selects the operation clock) and TAUDnMD0 can be set to any value, but other control bits have fixed values. For details, see **Section 25, Timer Array Unit D (TAUD)**.
For this feature clear TAUDnMD0 to 0.

Table 29.64 Common TAUDn Channel Settings (1/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTOE	15	TAUDnTOE15	0 1	Disables the timer. Enables the timer.
	14	TAUDnTOE14	0	
	13	TAUDnTOE13	0 1	Disables the timer. Enables the timer.
	12	TAUDnTOE12	0	
	11	TAUDnTOE11	0 1	Disables the timer. Enables the timer.
	10	TAUDnTOE10	0	
	9 to 4	TAUDnTOE09 to TAUDnTOE04	0 1	Disables the timer. Enables the timer.
	3	TAUDnTOE03	Don't care	
	2	TAUDnTOE02	0 1	Disables the timer. Enables the timer.
	1, 0	TAUDnTOE01 TAUDnTOE00	Don't care	

Table 29.64 Common TAUDn Channel Settings (2/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTO	15	TAUDnTO15	1*1	Outputs a high-level signal to TOUT15.
	14	TAUDnTO14	Don't Care	
	13	TAUDnTO13	1*1	Outputs a high-level signal to TOUT13.
	12	TAUDnTO12	Don't Care	
	11	TAUDnTO11	1*1	Outputs a high-level signal to TOUT11.
	10	TAUDnTO10	Don't Care	
	9 to 4	TAUDnTO09 to TAUDnTO04	0*1	Outputs a low-level signal to TOUT09 to TOUT04.
	3	TAUDnTO03	Don't Care	
	2	TAUDnTO02	0	Outputs a low-level signal to TOUT02.
	1, 0	TAUDnTO01 TAUDnTO00	Don't Care	
TAUDnTOM	15 to 10	TAUDnTOM15 to TAUDnTOM10	0	Independent operation mode
	9 to 4	TAUDnTOM09 to TAUDnTOM04	1	Synchronous operation mode
	3	TAUDnTOM03	Don't Care	
	2	TAUDnTOM02	0	Independent operation mode
	1, 0	TAUDnTOM01 TAUDnTOM00	Don't Care	
TAUDnTOC	15	TAUDnTOC15	1	Operation mode 2
	14	TAUDnTOC14	0	Operation mode 1
	13	TAUDnTOC13	1	Operation mode 2
	12	TAUDnTOC12	0	Operation mode 1
	11	TAUDnTOC11	1	Operation mode 2
	10	TAUDnTOC10	0	Operation mode 1
	9 to 4	TAUDnTOC09 to TAUDnTOC04	1	Operation mode 2
	3	TAUDnTOC03	Don't Care	
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01 TAUDnTOC00	Don't Care	
TAUDnTOL	15	TAUDnTOL15	1*1	Inverted logic output (active low)
	14	TAUDnTOL14	Don't Care	
	13	TAUDnTOL13	1*1	Inverted logic output (active low)
	12	TAUDnTOL12	Don't Care	
	11	TAUDnTOL11	1*1	Inverted logic output (active low)
	10	TAUDnTOL10	Don't Care	
	9 to 4	TAUDnTOL09 to TAUDnTOL04	0*1	Positive logic output (active high)
	3	TAUDnTOL03	Don't Care	
	2	TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01 TAUDnTOL00	Don't Care	

Table 29.64 Common TAUDn Channel Settings (3/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	0	Disables dead time control.
	9 to 4	TAUDnTDE09 to TAUDnTDE04	1	Enables dead time control.*2
	3	TAUDnTDE03	Don't Care	
	2	TAUDnTDE02	0	Disables dead time control.
	1, 0	TAUDnTDE01 TAUDnTDE00	Don't Care	
TAUDnTDM	15 to 9	TAUDnTDM15 to TAUDnTDM09	0	
	3	TAUDnTDM03	Don't Care	
	2	TAUDnTDM02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDM01 TAUDnTDM00	Don't Care	
TAUDnTDL	15 to 10	TAUDnTDL15 to TAUDnTDL10	0	Invalid because dead time control is disabled.
	9	TAUDnTDL09	1*1	Dead time is in the negative segment of the W phase output
	8	TAUDnTDL08	0*1	Dead time is in the positive segment of the W phase output
	7	TAUDnTDL07	1*1	Dead time is in the negative segment of the V phase output
	6	TAUDnTDL06	0*1	Dead time is in the positive segment of the V phase output
	5	TAUDnTDL05	1*1	Dead time is in the negative segment of the U phase output
	4	TAUDnTDL04	0*1	Dead time is in the positive segment of the U phase output
	3	TAUDnTDL03	Don't Care	
	2	TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01 TAUDnTDL00	Don't Care	
TAUDnTRE	15 to 4	TAUDnTRE15 to TAUDnTRE04	0	Disables real-time output.
	3	TAUDnTRE03	Don't Care	
	2	TAUDnTRE02	0	Disables real-time output.
	1, 0	TAUDnTRE01 TAUDnTRE00	Don't Care	
TAUDnTRO	15 to 4	TAUDnTRO15 to TAUDnTRO04	0	Invalid because real-time output is disabled.
	3	TAUDnTRO03	Don't Care	
	2	TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01 TAUDnTRO00	Don't Care	
TAUDnTRC	15 to 4	TAUDnTRC15 to TAUDnTRC04	0	Do not use this channel to generate the real-time output trigger.
	3	TAUDnTRC03	Don't Care	
	2	TAUDnTRC02	0	Do not use this channel to generate the real-time output trigger.
	1, 0	TAUDnTRC01 TAUDnTRC00	Don't Care	

Table 29.64 Common TAUDn Channel Settings (4/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTME	15 to 4	TAUDnTME15 to TAUDnTME04	0	Disables modulation output for timer output and real-time output.
	3	TAUDnTME03	Don't Care	
	2	TAUDnTME02	0	Disables modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don't Care	
TAUDnRDE	15 to 4	TAUDnRDE15 to TAUDnRDE04	1	Enables simultaneous rewriting.
	3	TAUDnRDE03	Don't Care	
	2	TAUDnRDE02	1	Enables simultaneous rewriting.
	1, 0	TAUDnRDE01 TAUDnRDE00	Don't Care	
TAUDnRDS	15 to 4	TAUDnRDS15 to TAUDnRDS04	0	Do not enable simultaneous rewriting by using another upper channel.
	3	TAUDnRDS03	Don't Care	
	2	TAUDnRDS02	0	Do not enable simultaneous rewriting by using another upper channel.
	1, 0	TAUDnRDS01 TAUDnRDS00	Don't Care	
TAUDnRDM	15 to 10	TAUDnRDM15 to TAUDnRDM10	0	Performs simultaneous rewriting when the master channel starts counting.
	9 to 4	TAUDnRDM09 to TAUDnRDM04	1	Performs simultaneous rewriting after the master channel starts counting when there is a peak in the triangle wave on the corresponding slave channel.
	3	TAUDnRDM03	Don't Care	
	2	TAUDnRDM02	1	Performs simultaneous rewriting after the master channel starts counting when there is a peak in the triangle wave on the corresponding slave channel.
	1, 0	TAUDnRDM01 TAUDnRDM00	Don't Care	
TAUDnRDC	15 to 4	TAUDnRDC15 to TAUDnRDC04	0	Do not use this channel to generate the simultaneous rewrite trigger.
	3	TAUDnRDC03	Don't Care	
	2	TAUDnRDC02	0	Do not use this channel to generate the simultaneous rewrite trigger.
	1, 0	TAUDnRDC01 TAUDnRDC00	Don't Care	

Note 1. Change the setting according to the used system.

Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 25, Timer Array Unit D (TAUD)**.

29.12.6.2 PIC Settings (Active High Example)

Table 29.65 PIC Settings

Register	Bit position	Bit name	Setting	Remark
PIC0REG2n0	18	PIC0REG2n018	1	Selects the TOUT signal of CH2 of TAUDn.
PIC0REG2n1	27, 26	PIC0REG2n127 PIC0REG2n126	1 0	Negative W phase active high combination circuit output
	25, 24	PIC0REG2n125 PIC0REG2n124	1 0	Positive W phase active high combination circuit output
	23, 22	PIC0REG2n123 PIC0REG2n122	1 0	Negative V phase active high combination circuit output
	21, 20	PIC0REG2n121 PIC0REG2n120	1 0	Positive V phase active high combination circuit output
	19, 18	PIC0REG2n119 PIC0REG2n118	1 0	Negative U phase active high combination circuit output
	17, 16	PIC0REG2n117 PIC0REG2n116	1 0	Positive U phase active high combination circuit output
PIC0REG2n2	25	PIC0REG2n225	1	Selects the input selected by the PIC0REG2n018 bit.
	21	PIC0REG2n221	1	Selects the input selected by the PIC0REG2n018 bit.
	17	PIC0REG2n217	1	Selects the input selected by the PIC0REG2n018 bit.
PIC0REG2n3	22, 21, 20	PIC0REG2n322 PIC0REG2n321 PIC0REG2n320	1 0 0	Negative W phase active high logical operation circuit output
	18, 17, 16	PIC0REG2n318 PIC0REG2n317 PIC0REG2n316	1 0 0	Positive W phase active high logical operation circuit output
	14, 13, 12	PIC0REG2n314 PIC0REG2n313 PIC0REG2n312	1 0 0	Negative V phase active high logical operation circuit output
	10, 9, 8	PIC0REG2n310 PIC0REG2n309 PIC0REG2n308	1 0 0	Positive V phase active high logical operation circuit output
	6, 5, 4	PIC0REG2n306 PIC0REG2n305 PIC0REG2n304	1 0 0	Negative U phase active high logical operation circuit output
	2, 1, 0	PIC0REG2n302 PIC0REG2n301 PIC0REG2n300	1 0 0	Positive U phase active high logical operation circuit output

29.13 Delay Pulse Output with Dead Time

29.13.1 Functional Overview

This function outputs a three-phase PWM signal with dead time delayed by a specified delay time from the cycle timing.

Unlike the function of three-phase PWM output with dead time, a PWM signal that has a reset in the next cycle can be output.

29.13.2 Configuration

The unit and channel configuration for this feature is shown below. (n = 0, m = 0 to 15)

Table 29.66 Configuration of Delay Pulse Output with Dead Time

Timer	Timer motor control function
TAUD0 CH2 to CH15 (used channels fixed)	TAPA0

The signal names used in the descriptions below are abbreviations. The actual signal names corresponding to each abbreviation are as follows:

- INTm → INTTAUDnIm (TAUDn channel m interrupt)
- TINm → TAUDTTINm (TAUDn channel m input)
- TOUTm → TAUDTTOUTm (TAUDn channel m output)
- CDRm → TAUDnCDRm (TAUDn channel m data register)
- CNTm → TAUDnCNTm (TAUDn channel m counter register)

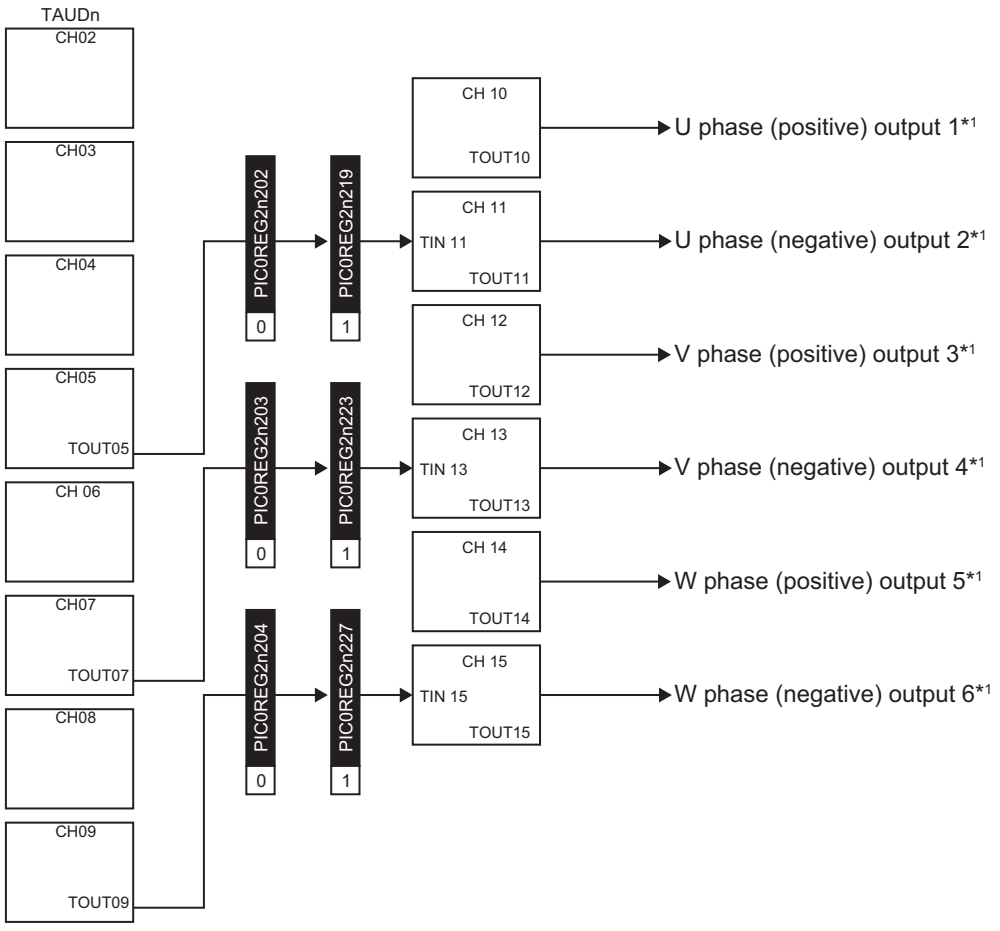
29.13.2.1 TAUDn configuration

Because the CDRm value of CH3 does not affect TOUT0 to TOUT15, the INTm signal of CH3 can also be used for other purposes such as A/D conversion trigger generation.

Table 29.67 TAUDn configuration

CH	Function name	M/S*[1]	CDR setting	Description
2	Delay pulse output function (CH2 is the master channel for CH3 to CH9.)	M	Cycle	
3		S		Reserved
4		S	Delay (U phase)	
5		S	Pulse width (U phase)	
6		S	Delay (V phase)	
7		S	Pulse width (V phase)	
8		S	Delay (W phase)	
9		S	Pulse width (W phase)	
10	Any feature that does not use TOUTm	S		TOUT: U phase output
11	One-phase PWM output	S	Dead time (U phase)	TOUT: UB-phase output
12	Any feature that does not use TOUTm	S		TOUT: V phase output
13	One-phase PWM output	S	Dead time (V phase)	TOUT: VB-phase output
14	Any feature that does not use TOUTm	S		TOUT: W phase output
15	One-phase PWM output	S	Dead time (W phase)	TOUT: WB-phase output

Note 1. M: Master channel, S: Slave channel



Note 1. For the connection destination, see **Figure 29.25, Block Diagram: Motor Output Buffer Control**.

Figure 29.43 Block Diagram: Delay Pulse Output with Dead Time

29.13.3 Registers

29.13.3.1 PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0)

This register selects the TAUDn CHm input signals. This section describes bits to be used in the delay pulse output with dead time.

Access: This register can be read or written in 32-bit units.

Address: PIC0REG202: FFDD 00C8$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	PIC0REG2n227	—	—	—	PIC0REG2n223	—	—	—	PIC0REG2n219	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PIC0REG2n204	PIC0REG2n203	PIC0REG2n202	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 29.68 PIC0REG2n2 Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	*1
27	PIC0REG2n227	Select the TIN input signal to TAUDTTIN15. **PIC0REG2n227** / **Input signal** 1 / Signal selected by the PIC0REG2n204 bit Other than the above / Setting prohibited
26 to 24	Reserved	*1
23	PIC0REG2n223	Select the TIN input signal to TAUDTTIN13. **PIC0REG2n223** / **Input signal** 1 / Signal selected by the PIC0REG2n203 bit Other than the above / Setting prohibited
22 to 20	Reserved	*1
19	PIC0REG2n219	Select the TIN input signal to TAUDTTIN11. **PIC0REG2n219** / **Input signal** 1 / Signal selected by the PIC0REG2n202 bit Other than the above / Setting prohibited
18 to 5	Reserved	*1
4	PIC0REG2n204	Select the signal supplied to TAUDTTIN15. 0: Select TAUDTTOUT9. 1: Setting prohibited
3	PIC0REG2n203	Select the signal supplied to TAUDTTIN13. 0: Select TAUDTTOUT7. 1: Setting prohibited
2	PIC0REG2n202	Select the signal supplied to TAUDTTIN11. 0: Select TAUDTTOUT5. 1: Setting prohibited
1, 0	Reserved	*1

Note 1. Bits defined as 0 in the PIC0REG2n2 register may be defined by another timer connection function. For such bits, use the bit definition of the timer connection function.

29.13.3.2 PIC0HIZCENn — Hi-Z Output Control Register n (n = 0)

This register selects the Hi-Z output control input signal of TAPAn.

Access: This register can be read or written in 8-bit units.

Address: PIC0HIZCEN0: FFDD 0080_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	PIC0HIZCENn6	—	—	—	PIC0HIZCENn2	—	PIC0HIZCENn0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R	R/W

Table 29.69 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	PIC0HIZCENn6	Select whether to enable or disable Hi-Z output control by the INTADCA0ERR interrupt signal. 0: Disable 1: Enable
5 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2	PIC0HIZCENn2	Select whether to enable or disable Hi-Z output control by the WDTA1NMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PIC0HIZCENn0	Select whether to enable or disable Hi-Z output control by the TAPAnESO pin input. 0: Disable 1: Enable

29.13.4 Operation Example

This is achieved by combining the following TAUD features:

- Delay pulse output function
- One-phase PWM output

The delay pulse output feature generates a PWM signal that is later than the cycle timing by an amount equal to the delay amount. Next, a one-phase PWM signal to which dead time has been added is output for the delayed PWM signal by one-phase PWM output feature.

A delay pulse with dead time is output by assigning the PWM output achieved using the above features to the U, V, and W phases. Therefore, the PWM output dead time can be freely specified for the PWM signal of each phase. Because the only difference between phases is the assigned channel, only one phase (the U phase) is described below.

29.13.4.1 Delay pulse output function

By using a combination of CH2, CH4, and CH5, a basic PWM signal for one-phase PWM is output from TOUT05 delayed by the amount specified by CH4 with respect to the cycle specified by CH2.

Note that CH3 is a reserved timer for achieving this feature, so do not use it for other features.

CAUTION

Do not specify a delay amount that exceeds the cycle.

29.13.4.2 One-phase PWM output

One-phase PWM output is generated from TOUT10 and TOUT11 by using a combination of CH10 and CH11.

By specifying the dead time value for CDR11, a one-phase PWM signal with dead time is output for the TIN11 input.

Similarly, the V phase uses CH12 and CH13 to output a one-phase PWM signal with dead time, while the W phase uses CH14 and CH15.

CAUTION

Specify the same clock for each TAUDn channel that uses the delay pulse output and one-phase PWM output features.

For details about the TAUD functions, see **Section 25, Timer Array Unit D (TAUD)**.

The differences between the delay pulse output with dead time and the three-phase PWM output with dead time are described below.

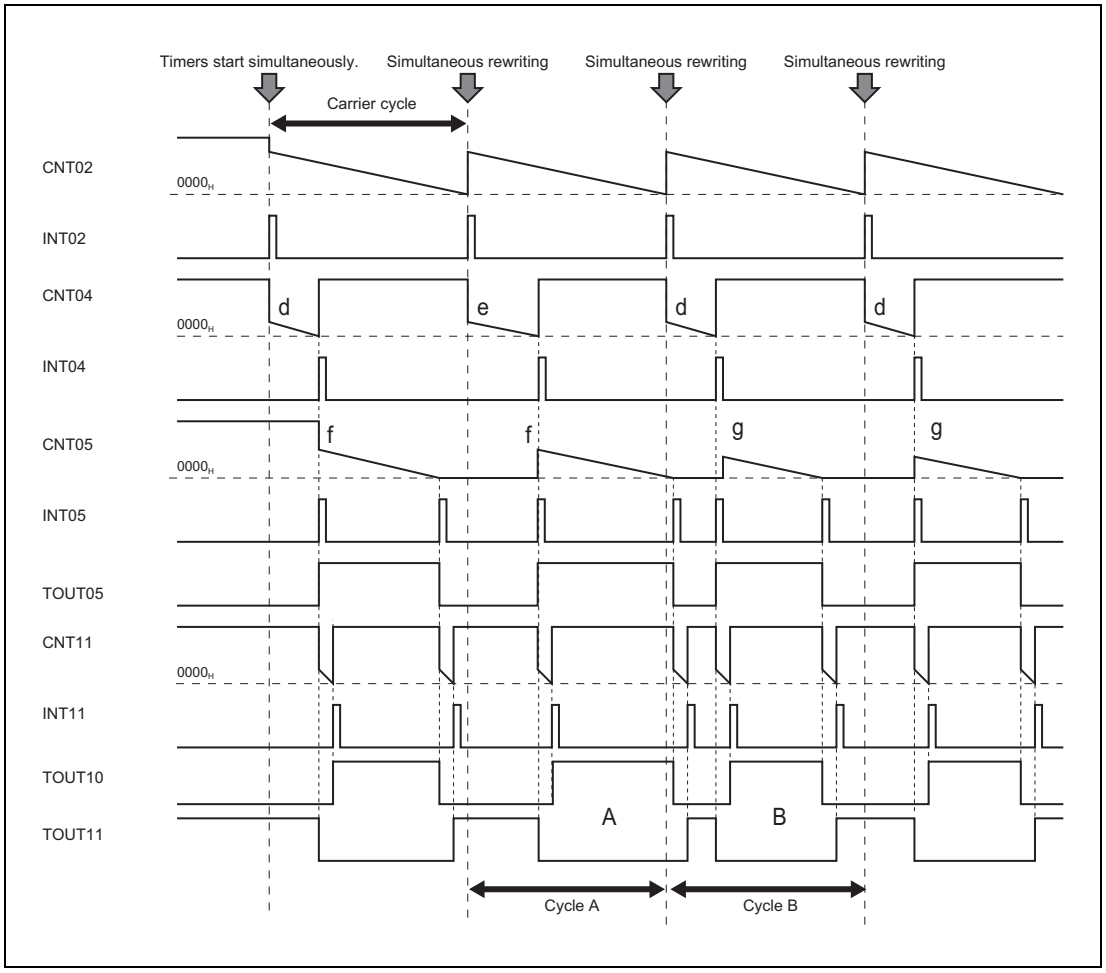


Figure 29.44 PWM Output by Outputting a Delay Pulse with Dead Time

In **Figure 29.44**, PWM waveform A is supposed to be output before cycle A ends, but because the delay timing is set longer, the PWM clear position is set over the end of cycle A. Next, PWM waveform B, which is for cycle B, is output.

The timing chart below shows the operation shown in **Figure 29.44, PWM Output by Outputting a Delay Pulse with Dead Time** implemented by the three-phase PWM output with dead time function.

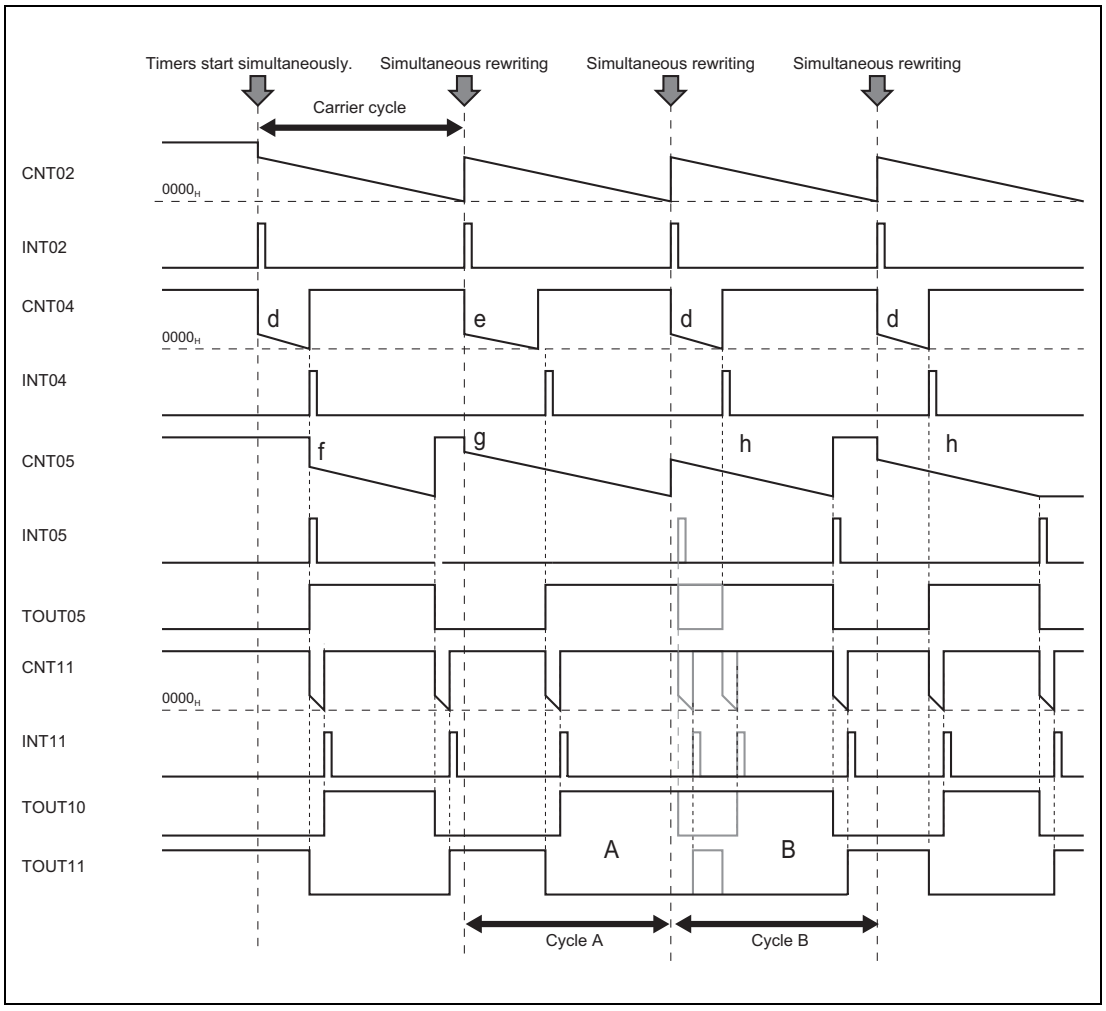


Figure 29.45 Output of a Three-Phase PWM Signal with Dead Time (1)

Figure 29.45 shows an example in which the output PWM signal does not end before carrier cycle A because the set timing for outputting a three-phase PWM signal with dead time is delayed and the clear timing is after the end of the carrier cycle.

For cycle A, the set timing of PWM waveform A is the same as that in the figure on the previous page, but because the clear timing is after the end of cycle A, a reload operation occurs in cycle A before PWM waveform A is cleared, and the clear timing for PWM waveform A does not occur.

In addition, the set timing of PWM waveform B for cycle B is ignored because a PWM waveform is already set. The result is that there is no PWM waveform change until the clear timing of cycle B, and a waveform with which PWM waveforms A and B are combined is output.

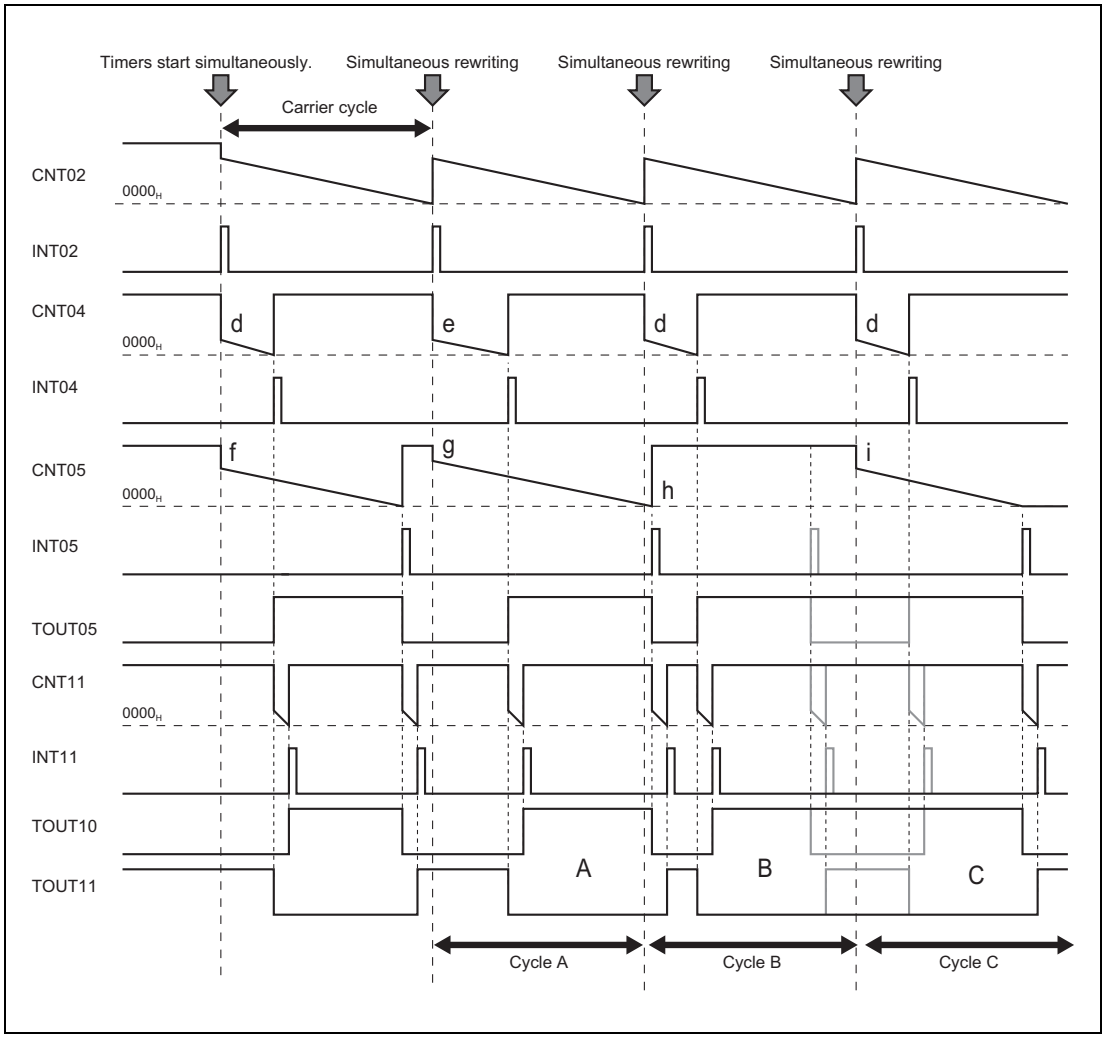


Figure 29.46 Output of a Three-Phase PWM Signal with Dead Time (2)

Figure 29.46 shows an example of outputting a three-phase PWM signal with dead time in which counter operation for which the clear timing is longer than cycle A is continued in cycle B, and PWM output A is cleared at the beginning of cycle B.

The output of PWM waveform A for cycle A is the same as the output of a delay pulse with dead time, but because the clear timing is used at the beginning of cycle B, the clear timing of PWM output B, which is supposed to be output during cycle B, does not occur.

In addition, the set timing of PWM waveform C for cycle C is ignored because a PWM waveform is already set. The result is that there is no PWM waveform change until the clear timing of cycle C, and a waveform with which PWM waveforms B and C are combined is output.

In this way, it is possible to achieve freer PWM output timing when outputting a delay pulse with dead time than when outputting a three-phase PWM signal with dead time.

The peripheral interconnections provide a connection for using the PWM output timing of delay pulse output as input for one-phase PWM output.

Figure 29.47 shows a timing chart for outputting a delay pulse with dead time.

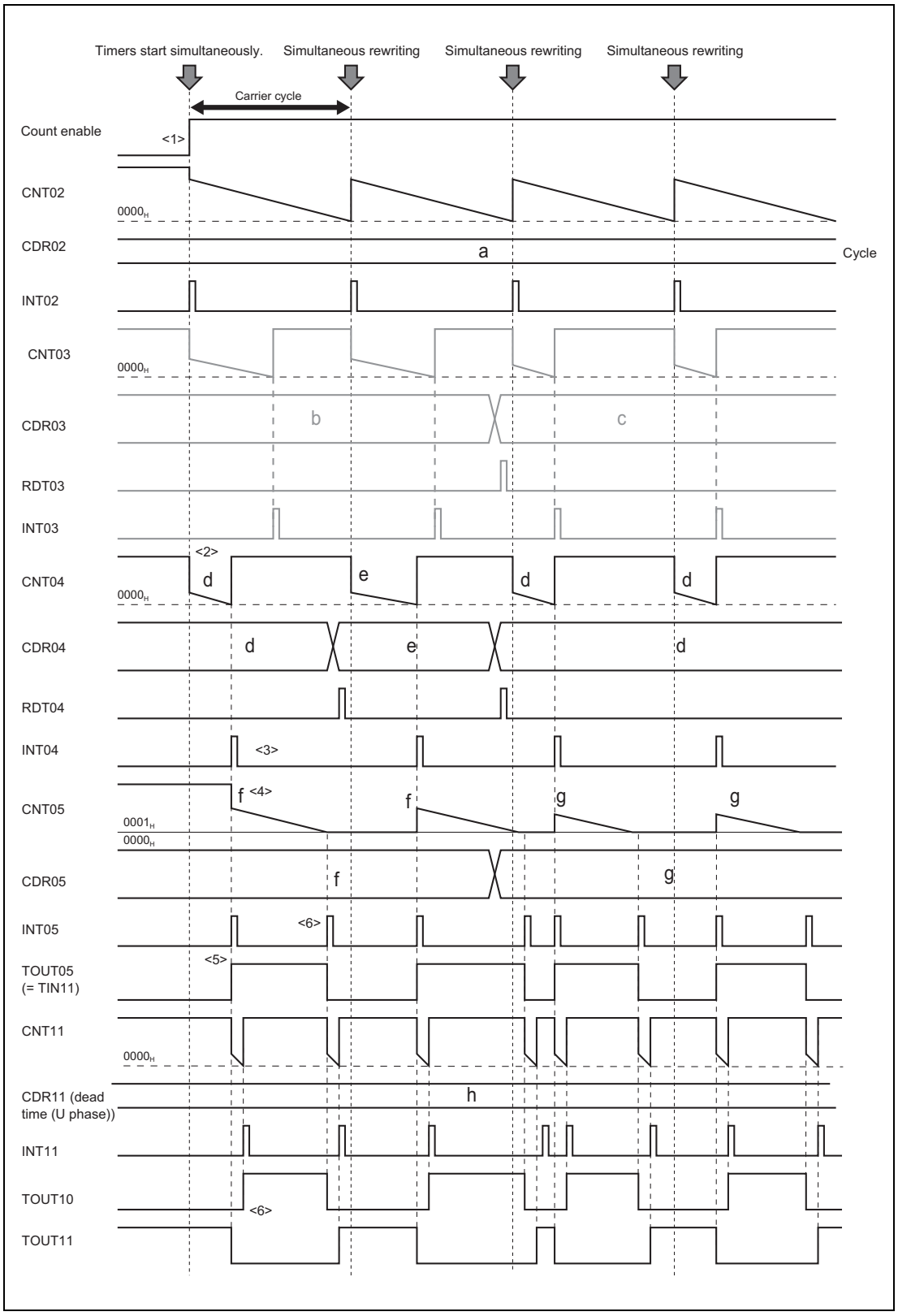


Figure 29.47 Output of a Delay Pulse with Dead Time

The output of a delay pulse with dead time shown in **Figure 29.47** is described below.

<1> CH2 (the carrier cycle timer) and CH4 (the delay timing timer) are started simultaneously by starting timers simultaneously.
CH5 (the PWM duty timer) and CH11 (the dead time timer) are also enabled, but no counting operations are performed until the edges of INT04, which indicates the count start timing for CH5, and TIN11, which indicates the count start timing for CH11, are detected.
Because CH3 does not affect PWM output for this function, the channel is not described.

<2> For CH4, when there is a CH2 underflow, the settings from CDR04 are reloaded to CNT04.

<3> The CH4 underflow generates the delay timing signal (INT04).

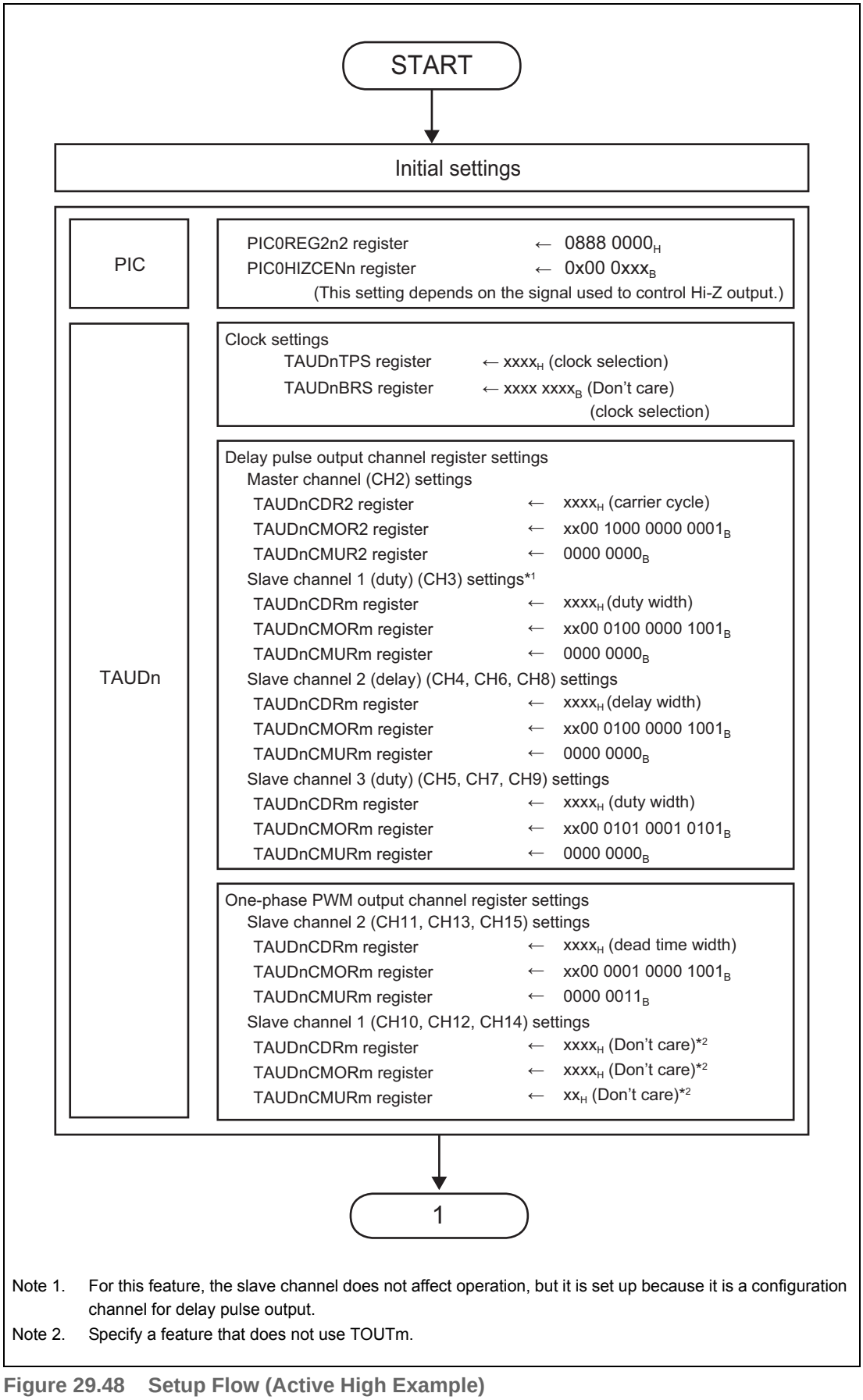
<4> When INT04 is generated, the settings from CDR05 are reloaded to CNT05, and then the CH5 (the PWM duty timer) operation starts.

<5> At this time, INT05 is generated and the TOUT05 output level changes to the active level.

<6> Due to the CH5 underflow, INT05 is generated again, and TOUT05 changes to the inactive level. TOUT05, which is changed by the CH4 and CH5 underflow, is supplied to the TIN11 input of one-phase PWM output.

<7> During one-phase PWM output, a PWM waveform with dead time is generated and output by detecting a TIN11 edge.

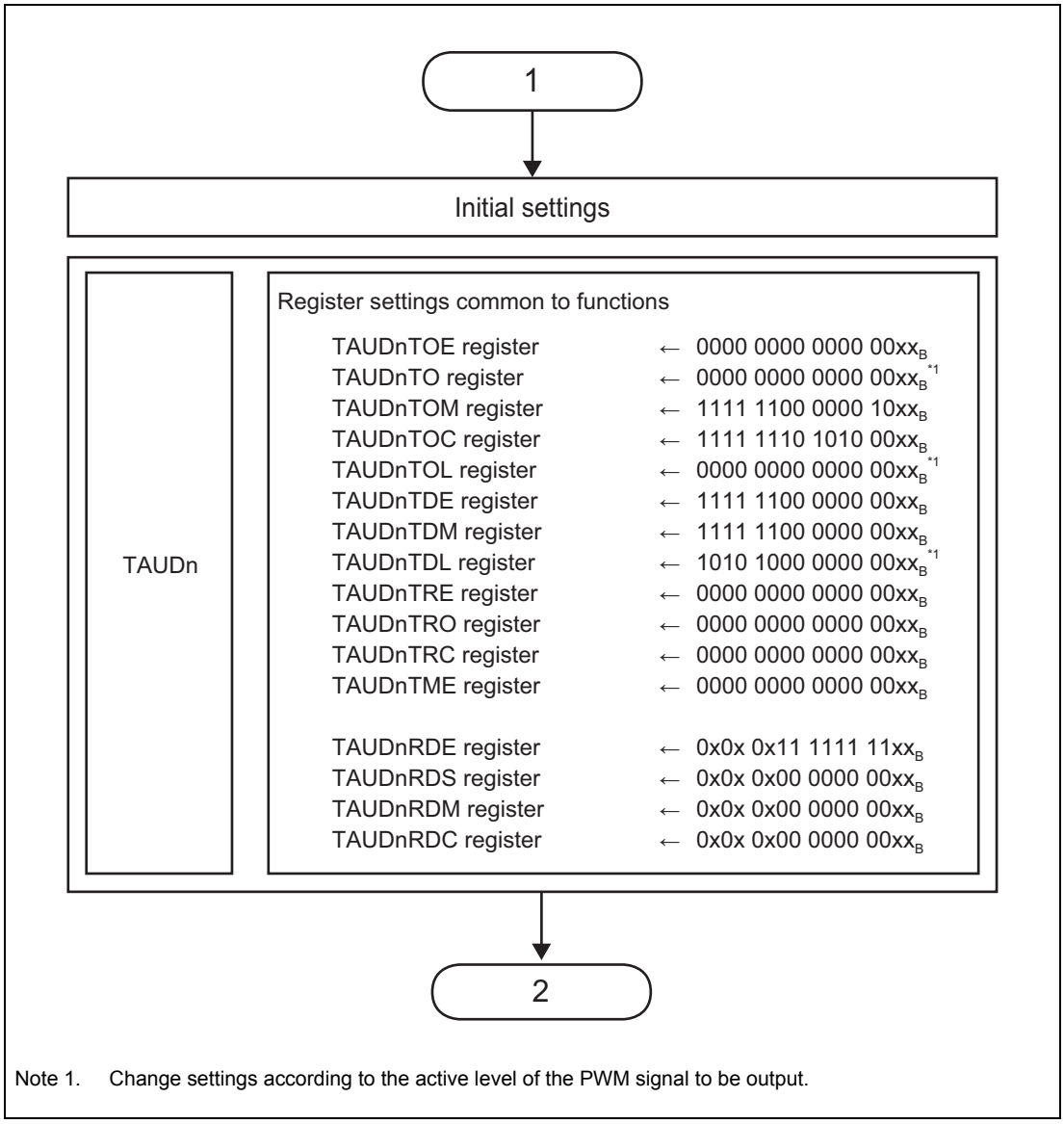
29.13.5 Setup Flow



Note 1. For this feature, the slave channel does not affect operation, but it is set up because it is a configuration channel for delay pulse output.

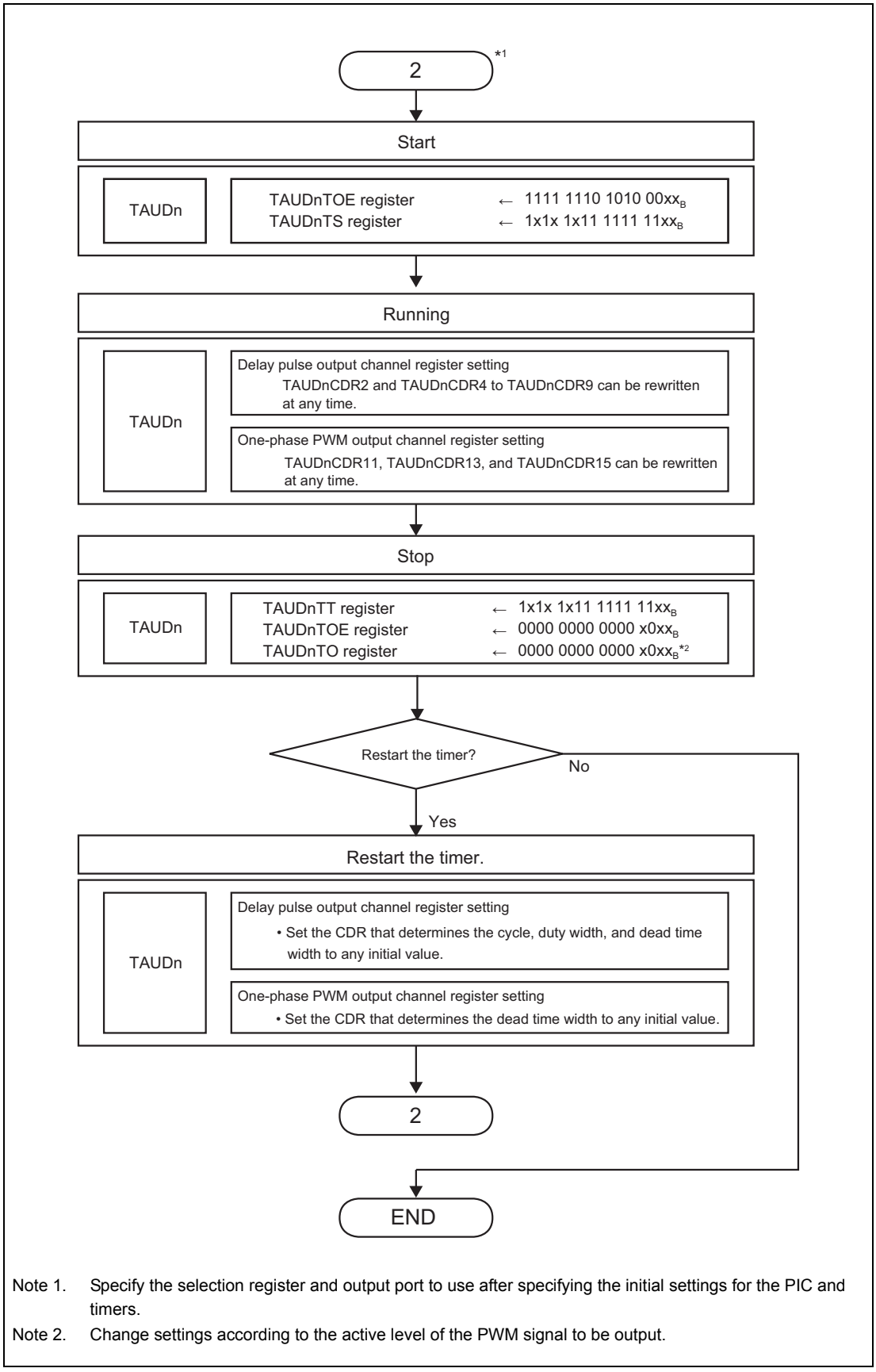
Note 2. Specify a feature that does not use TOUTm.

Figure 29.48 Setup Flow (Active High Example)



Note 1. Change settings according to the active level of the PWM signal to be output.

Figure 29.49 Setup Flow (Active High Example) (continued)



Note 1. Specify the selection register and output port to use after specifying the initial settings for the PIC and timers.

Note 2. Change settings according to the active level of the PWM signal to be output.

Figure 29.50 Setup Flow (Active High Example) (continued)

29.13.6 Setting Examples for Operation Functions

This section provides example settings for each register.

29.13.6.1 TAUDn Settings

Table 29.70 TAUDn: CH2-related (Master Channel used to Output a Delay Pulse*1)

Register	Bit position	Bit name	Setting	Remark
TAUDnCMOR2	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	1	
	10 to 8	TAUDnSTS[2:0]	000	
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0000	
	0	TAUDnMD0	1	Output INTm at the start of operation.
TAUDnCMUR2	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 29.71 TAUDn: CH3-related (Slave Channel used to Output a Delay Pulse*1*2)

Register	Bit position	Bit name	Setting	Remark
TAUDnCMOR3	15, 14	TAUDnCKS[1:0]	Don't care*3	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	Start trigger: INTm detection on the master channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enables start triggers during counting.
TAUDnCMUR3	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Note 3. For this feature, the channel does not affect operation, but it is set up because it is a configuration channel for delay pulse output.

NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Table 29.72 TAUDn: CH4, CH6, and CH8-related (Slave Channel 2 used to Output a Delay Pulse*1) (m = 4, 6, or 8)

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	100	Start trigger: INTm detection on the master channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enables start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the slave channel and master channel.

NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Table 29.73 TAUDn: CH5, CH7, and CH9-related (Slave Channel 3 used to Output a Delay Pulse*1) (m = 5, 7, or 9)

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*2	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	101	Start trigger: INTm detection on an upper channel
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	1010	
	0	TAUDnMD0	1	Enables start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	00	

Note 1. The master channel and slave channel names are defined for TAUD delay pulse output. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the slave channel and master channel.

NOTE

For the TAUDnCMORm register used during delay pulse output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 25, Timer Array Unit D (TAUD)**.

Table 29.74 TAUDn: CH11, CH13, and CH15-related (One-Phase PWM Output) (m = 11, 13, or 15)

Register	Bit position	Bit name	Setting	Remark
TAUDnCMORm	15, 14	TAUDnCKS[1:0]	Don't care*1	Operation clock setting
	13, 12	TAUDnCCS[1:0]	00	
	11	TAUDnMAS	0	
	10 to 8	TAUDnSTS[2:0]	001	Start trigger: Detection of a TINm-input valid edge
	7, 6	TAUDnCOS[1:0]	00	
	5		0	Fixed to 0
	4 to 1	TAUDnMD[4:1]	0100	
	0	TAUDnMD0	1	Enables start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	11	Both rising and falling TINm edges are detected as valid. (High width)

Note 1. Specify the same operation clock settings as for the PWM output master channel (CH2).

NOTE

For the TAUDnCMORm register used during one-phase PWM output, TAUDnCKS[1:0] (which selects the operation clock) can be set to any value, but other control bits have fixed values. For details, see **Section 25, Timer Array Unit D (TAUD)**.
CH10, CH12, and CH14 can be used with any feature that does not use TOUTm output (such as A/D trigger output).

Table 29.75 Common TAUDn Channel Settings (1/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTOE	15 to 10	TAUDnTOE15 to TAUDnTOE10	0 1	Disables the timer. Enables the timer.
	9	TAUDnTOE09	0 1	Disables the timer. Enables the timer.
	8	TAUDnTOE08	0	This is fixed to 0 because TOUT08 is not used.
	7	TAUDnTOE07	0 1	Disables the timer. Enables the timer.
	6	TAUDnTOE06	0	This is fixed to 0 because TOUT06 is not used
	5	TAUDnTOE05	0 1	Disables the timer. Enables the timer.
	4	TAUDnTOE04	0	This is fixed to 0 because TOUT04 is not used
	3	TAUDnTOE03	0	This is fixed to 0 because TOUT03 is not used
	2	TAUDnTOE02	0	This is fixed to 0 because TOUT02 is not used.
	1, 0	TAUDnTOE01 TAUDnTOE00	Don't care	
TAUDnTO	15 to 10	TAUDnTO15 to TAUDnTO10	0*1	Outputs a low-level signal to TOUT15 to TOUT10.
	9 to 2	TAUDnTO09 to TAUDnTO02	0	Outputs a low-level signal to TOUT09 to TOUT02.
	1, 0	TAUDnTO01 TAUDnTO00	Don't care	

Table 29.75 Common TAUDn Channel Settings (2/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTOM	15 to 10	TAUDnTOM15 to TAUDnTOM10	1	Synchronous operation mode
	9 to 4	TAUDnTOM09 to TAUDnTOM04	0	Independent operation mode
	3	TAUDnTOM03	1	Synchronous operation mode
	2	TAUDnTOM02	0	Independent operation mode
	1, 0	TAUDnTOM01 TAUDnTOM00	Don't Care	
TAUDnTOC	15 to 10	TAUDnTOC15 to TAUDnTOC10	1	Synchronous operation mode 2
	9 to 4	TAUDnTOC09 to TAUDnTOC04	1, 0, 1, 0, 1, 0	CH5, CH7, CH9: Operation mode 2 CH4, CH6, CH8: Operation mode 1
	3	TAUDnTOC03	0	Operation mode 1
	2	TAUDnTOC02	0	Operation mode 1
	1, 0	TAUDnTOC01 TAUDnTOC00	Don't Care	
TAUDnTOL	15 to 10	TAUDnTOL15 to TAUDnTOL10	0*1	Positive logic output (active high)
	9 to 2	TAUDnTOL09 to TAUDnTOL02	0	Positive logic output (active high)
	1, 0	TAUDnTOL01 TAUDnTOL00	Don't Care	
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	1	Enables dead time control.*2
	9 to 2	TAUDnTDE09 to TAUDnTDE02	0	Disables dead time control.
	1, 0	TAUDnTDE01 TAUDnTDE00	Don't Care	
TAUDnTDM	15 to 10	TAUDnTDM15 to TAUDnTDM10	1	Outputs dead time upon detecting a TINm input edge at a lower odd channel.
	9 to 2	TAUDnTDM09 to TAUDnTDM02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDM01 TAUDnTDM00	Don't Care	
TAUDnTDL	15	TAUDnTDL15	1*1	Dead time is in the negative segment of the W phase output
	14	TAUDnTDL14	0*1	Dead time is in the negative segment of the W phase output
	13	TAUDnTDL13	1*1	Dead time is in the negative segment of the V phase output
	12	TAUDnTDL12	0*1	Dead time is in the negative segment of the V phase output
	11	TAUDnTDL11	1*1	Dead time is in the negative segment of the U phase output
	10	TAUDnTDL10	0*1	Dead time is in the negative segment of the U phase output
	9 to 2	TAUDnTDL09 to TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01 TAUDnTDL00	Don't Care	

Table 29.75 Common TAUDn Channel Settings (3/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTRE	15 to 2	TAUDnTRE15 to TAUDnTRE02	0	Disables real-time output.
	1, 0	TAUDnTRE01 TAUDnTRE00	Don't Care	
TAUDnTRO	15 to 2	TAUDnTRO15 to TAUDnTRO02	0	Invalid because real-time output is disabled.
	1, 0	TAUDnTRO01 TAUDnTRO00	Don't Care	
TAUDnTRC	15 to 2	TAUDnTRC15 to TAUDnTRC02	0	Do not use this channel to generate the real-time output trigger.
	1, 0	TAUDnTRC01 TAUDnTRC00	Don't Care	
TAUDnTME	15 to 2	TAUDnTME15 to TAUDnTME02	0	Disables modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don't Care	
TAUDnRDE	15	TAUDnRDE15	0	Disables simultaneous rewriting.
	14	TAUDnRDE14	Don't Care	
	13	TAUDnRDE13	0	Disables simultaneous rewriting.
	12	TAUDnRDE12	Don't Care	
	11	TAUDnRDE11	0	Disables simultaneous rewriting.
	10	TAUDnRDE10	Don't Care	
	9 to 2	TAUDnRDE09 to TAUDnRDE02	1	Enables simultaneous rewriting.
	1, 0	TAUDnRDE01 TAUDnRDE00	Don't Care	
TAUDnRDS	15	TAUDnRDS15	0	Do not enable simultaneous rewriting by using another upper channel.
	14	TAUDnRDS14	Don't Care	
	13	TAUDnRDS13	0	Do not enable simultaneous rewriting by using another upper channel.
	12	TAUDnRDS12	Don't Care	
	11	TAUDnRDS11	0	Do not enable simultaneous rewriting by using another upper channel.
	10	TAUDnRDS10	Don't Care	
	9 to 2	TAUDnRDS09 to TAUDnRDS02	0	Enables simultaneous rewriting by using a master channel.
	1, 0	TAUDnRDS01 TAUDnRDS00	Don't Care	

Table 29.75 Common TAUDn Channel Settings (4/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnRDM	15	TAUDnRDM15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDM14	Don't Care	
	13	TAUDnRDM13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDM12	Don't Care	
	11	TAUDnRDM11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDM10	Don't Care	
	9 to 2	TAUDnRDM09 to TAUDnRDM02	0	Loads the signal when the master channel starts counting.
	1, 0	TAUDnRDM01 TAUDnRDM00	Don't Care	
TAUDnRDC	15	TAUDnRDC15	0	Invalid because simultaneous rewriting is not enabled.
	14	TAUDnRDC14	Don't Care	
	13	TAUDnRDC13	0	Invalid because simultaneous rewriting is not enabled.
	12	TAUDnRDC12	Don't Care	
	11	TAUDnRDC11	0	Invalid because simultaneous rewriting is not enabled.
	10	TAUDnRDC10	Don't Care	
	9 to 2	TAUDnRDC09 to TAUDnRDC02	0	Do not use this channel to generate the simultaneous rewrite trigger.
	1, 0	TAUDnRDC01 TAUDnRDC00	Don't Care	

Note 1. Change the setting according to the used system.

Note 2. These are used to control positive/negative phase waveform output for which even channels are paired with odd channels to perform dead time control. For details, see **Section 25, Timer Array Unit D (TAUD)**.

29.13.6.2 Peripheral Interconnections Settings

Table 29.76 Peripheral Interconnections Settings

Register	Bit position	Bit name	Setting	Remark
PIC0REG2n2	27	PIC0REG2n227	1	Selects the input selected by the PIC0REG2n204 bit.
	23	PIC0REG2n223	1	Selects the input selected by the PIC0REG2n203 bit.
	19	PIC0REG2n219	1	Selects the input selected by the PIC0REG2n202 bit.
	4	PIC0REG2n204	0	Selects TAUDTTOUT9.
	3	PIC0REG2n203	0	Selects TAUDTTOUT7.
	2	PIC0REG2n202	0	Selects TAUDTTOUT5.

Section 30 PWM Output/Diagnostic (PWM-Diag)

This section contains a generic description of the PWM output/diagnostic function (PWM-Diag).

The first part of this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the units constituting PWM-Diag.

30.1 Features of RH850/F1M PWM-Diag

30.1.1 Number of Units and Channels

The PWM-Diag unit consists of a PWBA block for generating clock signals, PWGA blocks that generate PWM signals, and a PWSA block for generating triggers for A/D conversion. The numbers of individual units are listed below.

Each PWGA unit has one PWM channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 30.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
PWBA			
Number of Units	1		
Name	PWBAn (n = 0)		
PWGA			
Number of Units	64	72	80
Name	PWGAn (n = 0 to 63)	PWGAn (n = 0 to 71)	PWGAn (n = 0 to 79)
PWSA			
Number of Units	1		
Name	PWSAn (n = 0)		

Table 30.2 Indices

Index	Description
n	Throughout this section, individual units constituting the PWM-Diag function are identified by the index “n”; for example, PWBAnTE indicates the PWBAn status register.
m	The PWBA generation clock is identified by the index “m” (m = 0 to 3); for example, PWBAnBRSm indicates the PWMCLKm clock cycle configuration register.
x, y	An A/D converter configuration register number corresponding to a PWM-Diag channel is identified by the index “x, y”; for example, PWSAnPVCRx_y (x_y = 00_01, 02_03, ..., 78_79).
j	Registers storing trigger channel numbers (encoded value) from PWGAn are identified by the index “j”; for example, the PWSAnQUEj register (j = 0 to 7).
k	Sets of registers where each has the same function are identified by the index “k”; for example, the SLPWGAk register (k = 0 to 2).

The following table shows values indicated by the indices of each product.

Table 30.3 Indices of Products

Indices of each product		
144 pins	**176 pins**	**233 pins**
x = 00, 02, ,,, 62 y = 01, 03, ,,, 63	x = 00, 02, ,,, 70 y = 01, 03, ,,, 71	x = 00, 02, ,,, 78 y = 01, 03, ,,, 79
j = 0 to 7	j = 0 to 7	j = 0 to 7
k = 0 to 2	k = 0 to 2	k = 0 to 2

30.1.2 Register Base Address

PWM-Diag base addresses are listed in the following table.

PWM-Diag register addresses are given as offsets from the base addresses.

Table 30.4 Register Base Addresses

Base Address Name	Base Address
<PWBAn_base>	FFE7 2800_H
<PWGAn_base>	FFE7 1000_H + 40_H × n
<PWSAn_base>	FFE7 0000_H

30.1.3 Clock Supply

The PWM-Diag clock supply is shown in the following table.

Table 30.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
PWBAn	PCLK	CKSCLK_IPERI2
PWGAn	PCLK	CKSCLK_IPERI2
PWSAn	PCLK	CKSCLK_IPERI2

30.1.4 Interrupt Requests

PWM-Diag interrupt requests are listed in the following table.

Table 30.6 Interrupt Requests (1/2)

Unit Interrupt Signal	Signal Outline	Interrupt Number	DMA Trigger Number
PWGA_INT0	PWGA0 interrupt	92	—
PWGA_INT1	PWGA1 interrupt	93	—
PWGA_INT2	PWGA2 interrupt	94	—
PWGA_INT3	PWGA3 interrupt	95	—
PWGA_INT4	PWGA4 interrupt	85	—
PWGA_INT5	PWGA5 interrupt	86	—
PWGA_INT6	PWGA6 interrupt	87	—
PWGA_INT7	PWGA7 interrupt	88	—
PWGA_INT8	PWGA8 interrupt	96	—
PWGA_INT9	PWGA9 interrupt	97	—
PWGA_INT10	PWGA10 interrupt	98	—
PWGA_INT11	PWGA11 interrupt	99	—
PWGA_INT12	PWGA12 interrupt	100	—
PWGA_INT13	PWGA13 interrupt	101	—
PWGA_INT14	PWGA14 interrupt	102	—
PWGA_INT15	PWGA15 interrupt	103	—
PWGA_INT16	PWGA16 interrupt	145	—
PWGA_INT17	PWGA17 interrupt	147	—
PWGA_INT18	PWGA18 interrupt	149	—
PWGA_INT19	PWGA19 interrupt	151	—
PWGA_INT20	PWGA20 interrupt	124	—
PWGA_INT21	PWGA21 interrupt	125	—
PWGA_INT22	PWGA22 interrupt	126	—
PWGA_INT23	PWGA23 interrupt	127	—
PWGA_INT24	PWGA24 interrupt	184	—
PWGA_INT25	PWGA25 interrupt	185	—
PWGA_INT26	PWGA26 interrupt	153	—
PWGA_INT27	PWGA27 interrupt	186	—
PWGA_INT28	PWGA28 interrupt	187	—
PWGA_INT29	PWGA29 interrupt	188	—
PWGA_INT30	PWGA30 interrupt	155	—
PWGA_INT31	PWGA31 interrupt	157	—
PWGA_INT32	PWGA32 interrupt	189	—
PWGA_INT33	PWGA33 interrupt	190	—
PWGA_INT34	PWGA34 interrupt	191	—
PWGA_INT35	PWGA35 interrupt	192	—
PWGA_INT36	PWGA36 interrupt	193	—
PWGA_INT37	PWGA37 interrupt	194	—
PWGA_INT38	PWGA38 interrupt	195	—
PWGA_INT39	PWGA39 interrupt	196	—

Table 30.6 Interrupt Requests (2/2)

Unit Interrupt Signal	Signal Outline	Interrupt Number	DMA Trigger Number
PWGA_INT40	PWGA40 interrupt	197	—
PWGA_INT41	PWGA41 interrupt	198	—
PWGA_INT42	PWGA42 interrupt	199	—
PWGA_INT43	PWGA43 interrupt	200	—
PWGA_INT44	PWGA44 interrupt	201	—
PWGA_INT45	PWGA45 interrupt	202	—
PWGA_INT46	PWGA46 interrupt	203	—
PWGA_INT47	PWGA47 interrupt	204	—
PWGA_INT48	PWGA48 interrupt	240	—
PWGA_INT49	PWGA49 interrupt	241	—
PWGA_INT50	PWGA50 interrupt	242	—
PWGA_INT51	PWGA51 interrupt	243	—
PWGA_INT52	PWGA52 interrupt	244	—
PWGA_INT53	PWGA53 interrupt	245	—
PWGA_INT54	PWGA54 interrupt	246	—
PWGA_INT55	PWGA55 interrupt	247	—
PWGA_INT56	PWGA56 interrupt	248	—
PWGA_INT57	PWGA57 interrupt	249	—
PWGA_INT58	PWGA58 interrupt	250	—
PWGA_INT59	PWGA59 interrupt	251	—
PWGA_INT60	PWGA60 interrupt	252	—
PWGA_INT61	PWGA61 interrupt	253	—
PWGA_INT62	PWGA62 interrupt	254	—
PWGA_INT63	PWGA63 interrupt	255	—
PWGA_INT64	PWGA64 interrupt	277	—
PWGA_INT65	PWGA65 interrupt	278	—
PWGA_INT66	PWGA66 interrupt	279	—
PWGA_INT67	PWGA67 interrupt	280	—
PWGA_INT68	PWGA68 interrupt	281	—
PWGA_INT69	PWGA69 interrupt	282	—
PWGA_INT70	PWGA70 interrupt	283	—
PWGA_INT71	PWGA71 interrupt	284	—
PWGA_INT72	PWGA72 interrupt	290	—
PWGA_INT73	PWGA73 interrupt	291	—
PWGA_INT74	PWGA74 interrupt	292	—
PWGA_INT75	PWGA75 interrupt	293	—
PWGA_INT76	PWGA76 interrupt	294	—
PWGA_INT77	PWGA77 interrupt	295	—
PWGA_INT78	PWGA78 interrupt	296	—
PWGA_INT79	PWGA79 interrupt	297	—
PWSA_INT_QFULL	PWSA queue full interrupt	91	—

30.1.5 Reset Sources

PWM-Diag reset sources are listed in the following table. The individual PWM-Diag units are initialized by these reset sources.

Table 30.7 Reset Sources

Unit Name	Reset Source
PWBAn PWGAn PWSAn	All reset sources (ISORES)

30.1.6 External Input/Output Signals

External input/output signals of the PWM-Diag are listed below.

Table 30.8 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
PWGA_TOUTn (Unit: PWGA)	PWGA unit n output	PWGAnO

CAUTION

When port P0_0 is used as PWGA10O, the P0_0 pin outputs a low-level $\overline{\text{RESETOUT}}$ signal while a reset is asserted and continues to output a low level after the reset is deasserted.
For details, see **Section 2.11.1.1, P0_0: $\overline{\text{RESETOUT}}$**.

30.1.7 Internal Signals

The I/O signals for connecting two PWM-Diag channels or a PWM-Diag and another function are listed below.

Table 30.9 Internal Output Signals

Unit Signal Name	Description	Connected to
PWBA0		
PWMCLK0	PWGA count clock 0	PWGAn
PWMCLK1	PWGA count clock 1	PWGAn
PWMCLK2	PWGA count clock 2	PWGAn
PWMCLK3	PWGA count clock 3	PWGAn
PWGAn		
PWGA_TRGOUT	PWGAn trigger	PWSA0
PWSA0		
PWSA_ADTRG[1:0]	A/D converter unit select signal	ADCA0, ADCA1
PWSA_PVCR_VALUE[11:0]	A/D converter control signal	ADCA0, ADCA1
ADCAn		
ADC_CONV_ENDn	A/D conversion completion signal	PWSA0

30.1.8 Functional Overview

This function is comprised of four types of units: clock divider (PWBA), PWM generator (PWGA), A/D conversion trigger select function (PWSA), and A/D converter (ADCA).

PWBA

- Clock divider
  PWBA generates a PWMCLKm count clock signal by frequency division of PCLK and supplies it to the PWM generator PWGA.
  The cycle of the PWMCLKm count clock signal can be calculated from the setting of the PWBAnBRSm register by the equation below.

  PWMCLKm count clock cycle = (PWBAnBRSm value × 2) × PCLK cycle

  In addition, PWBA can control operation when the on-chip debugger is in use by using the PWBAnEMU register.

PWGA

PWGA outputs PWM waveforms and A/D conversion trigger to PWSA by using the clock PWMCLKm input from PWBA.

- PWM waveform output PWGA_TOUTn
  This generator outputs PWM waveforms from the PWGA_TOUTn pin. The PWM cycle is the full count cycle of the PWGAnCNT register (12-bit free-running counter). Set the high-level period of PWM output in the PWGAnCSDR and PWGAnCRDR registers.

  The PWM waveform cycle and duty can be calculated by the equations below.

  PWM waveform cycle = PWGAnCNT (12-bit full count: FFF_H + 1)
  × Count clock cycle
  = 4096 × PWMCLKm count clock cycle

  When PWGAnCRDR[11:0] > PWGAnCSDR[11:0],

  High-level period of PWM waveform =
  (PWGAnCRDR register value – PWGAnCSDR register value)
  × PWMCLKm count clock cycle

  PWM waveform duty (%) = High-level period of PWM waveform/PWM waveform cycle × 100 = (PWGAnCRDR register value – PWGAnCSDR register value)/4096 × 100

  Note that the PWM output is fixed to the low level when the PWGAnCRDR register value is equal to the PWGAnCSDR register value.

  When $1xxx_H$ is set in the PWGAnCRDR register (i.e. bit 12 is set to 1), the PWM output is fixed to the high level.

- A/D conversion trigger output PWGA_TRGOUTn
  The A/D conversion trigger signal PWGA_TRGOUTn for PWSA is generated when the PWGAnCTDR register value and the PWGAnCNT register value match while the PWM output PWGA_TOUTn is at the high level.
  The timing can be calculated by the equation below.

  A/D conversion trigger signal generation timing = PWGAnCTDR register value
  × PWMCLKm count clock cycle

- PWGA interrupt request signal PWGA_INTn
  PWGA generates the interrupt request signal PWGA_INTn at the falling edge of the PWM output PWGA_TOUTn.
  When the PWM output is fixed to the low level, PWGA_INTn is generated when the PWGAnCRDR register value and the PWGAnCNT register value (free-running counter value) match; when the PWM output is fixed to the high level, PWGA_INTn is generated at the timing of overflow of the PWGAnCNT register.

PWSA

PWSA transmits the required setting information to the A/D converter and outputs the A/D conversion start trigger, based on the A/D conversion trigger signal PWGA_TRGOUTn from the PWM generator (PWGA).

- A/D conversion control by PWSA
  PWSA outputs the information required for the A/D conversion, which is set in the corresponding PWSAnPVCRx_y register for the channel number of the trigger input from PWGAn, (i.e., information on ADC physical channel, external MPX control, and error detection level selection) to the A/D converter.
  At the same timing, A/D conversion trigger (PWSA_ADTRG) is output to ADCA0 or ADCA1.
  (A maximum of eight input trigger signal PWGA_TRGOUTn data received during A/D conversion are stored and kept in PWSAnQUE.)
  The setting information to be output to the A/D converter is kept until the next trigger is generated.
  When the A/D conversion triggered by the PWM-Diag function is completed in the A/D converter, PWSA triggers the next A/D conversion based on the data stored in the PWSAnQUE register.
- Queuing of A/D conversion triggers from PWGA
  The A/D conversion trigger signal (PWGA_TRGOUTn) input from PWGAn is stored in the PWSAnQUEj register as a channel number. The PWSAnQUEj register stores a maximum of eight channel numbers of the A/D conversion trigger signal PWGA_TRGOUTn received during A/D conversion in a queue structure.
  A PWSA queue full interrupt occurs in the following states, when the queue of the PWSAnQUEj register becomes full:
  - A trigger number is written to PWSAnQUE7
  - A trigger number has already been written to PWSAnQUE7 and cannot be written when PWGA_TRGOUTn is input.

ADCA

A/D conversion is executed upon receipt of information required for A/D conversion and A/D conversion trigger from PWSA.

A/D conversion is executed using the PWM-Diag-dedicated scan group; on completion of the A/D conversion, it is reported to the PWSA.

For the basic operation of the A/D converter, see **Section 31, A/D Converter (ADCA)**.

For the A/D converter operation with the PWM-Diag function, see **Section 31.4.7.1, A/D Conversion with PWM-Diag Enabled**.

30.1.9 Block Diagram

The following figure shows an example of connecting the LED control circuit combining the PWM-Diag and the A/D converter.

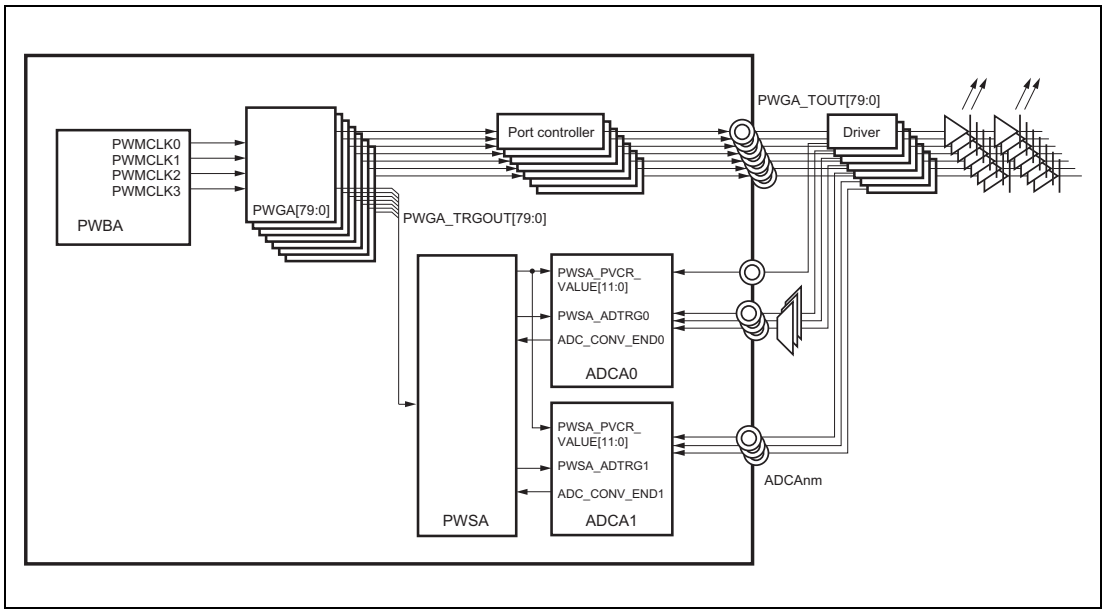


Figure 30.1 Example of Connecting the LED Control Circuit using the PWM-Diag and the A/D Converter

30.2 Registers

30.2.1 List of Registers

PWM Output/Diagnostic registers are listed in the following table.

<PWBAn_base>, <PWSAn_base>, and <PWGAn_base> are defined in **Section 30.1.2, Register Base Address**.

Table 30.10 List of PWM Output/Diagnostic Registers

Module name	Register name	Symbol	Address
PWBAn	PWMCLKm cycle configuration register	PWBAnBRSm	<PWBAn_base> + 0004_H × m
	PWMCLKm enable status register	PWBAnTE	<PWBAn_base> + 0010_H
	PWMCLKm start trigger register	PWBAnTS	<PWBAn_base> + 0014_H
	PWMCLKm stop trigger register	PWBAnTT	<PWBAn_base> + 0018_H
	PWBA emulation register	PWBAnEMU	<PWBAn_base> + $001C_H$
PWGAn	PWM output set condition register	PWGAnCSDR	<PWGAn_base> + 0000_H
	PWM output reset condition register	PWGAnCRDR	<PWGAn_base> + 0004_H
	PWGA_TRGOUTn generation condition register	PWGAnCTDR	<PWGAn_base> + 0008_H
	Buffer register reload trigger register	PWGAnRDT	<PWGAn_base> + $000C_H$
	Buffer register reload status register	PWGAnRSF	<PWGAn_base> + 0010_H
	PWM cycle count register	PWGAnCNT	<PWGAn_base> + 0014_H
	PWGA control register	PWGAnCTL	<PWGAn_base> + 0020_H
	PWGAnCSDR buffer register	PWGAnCSBR	<PWGAn_base> + 0024_H
	PWGAnCRDR buffer register	PWGAnCRBR	<PWGAn_base> + 0028_H
	PWGAnCTDR buffer register	PWGAnCTBR	<PWGAn_base> + $002C_H$
—	PWGA synchronous trigger register	SLPWGAk	FFE7 3000_H + k × 4_H
PWSAn	PWSA control register	PWSAnCTL	<PWSAn_base> + 0000_H
	Trigger queue status register	PWSAnSTR	<PWSAn_base> + 0004_H
	Trigger queue status clear register	PWSAnSTC	<PWSAn_base> + 0008_H
	Trigger queue register	PWSAnQUEj	<PWSAn_base> + 0020_H + j × 4_H
	PWM-Diag mode A/D setting register	PWSAnPVCRx_y	<PWSAn_base> + 0040_H + x × 2_H
	PWSA emulation control register	PWSAnEMU	<PWSAn_base> + $000C_H$

30.2.1.1 PWBAnBRSm Register

This register sets the clock cycle of PWMCLKm.

Access: This register can be read or written in 16-bit units.

Address: <PWBAn_base> + 0004_H × m

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	PWBAnBRSm[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.11 PWBAnBRSm Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
10 to 0	PWBAnBRSm [10:0]	Register for setting the clock cycle of PWMCLKm. – PWBAnBRSm = 0: PWMCLKm = PCLK – PWBAnBRSm = 1: PWMCLKm = PCLK / 2 × 1 – PWBAnBRSm = 2: PWMCLKm = PCLK / 2 × 2 ... – PWBAnBRSm = n: PWMCLKm = PCLK / 2 × n (n = 1 to 2047) These bits can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0).

30.2.1.2 PWBAnTE Register

This is a status register that indicates the output status of PWMCLKm (m = 0 to 3).

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWBAn_base> + 0010_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTE3	PWBAnTE2	PWBAnTE1	PWBAnTE0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 30.12 PWBAnTE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3	PWBAnTE3	A status flag indicating the operation status of PWMCLK3 0: Not operating 1: Operating
2	PWBAnTE2	A status flag indicating the operation status of PWMCLK2 0: Not operating 1: Operating
1	PWBAnTE1	A status flag indicating the operation status of PWMCLK1 0: Not operating 1: Operating
0	PWBAnTE0	A status flag indicating the operation status of PWMCLK0 0: Not operating 1: Operating

30.2.1.3 PWBAnTS Register

This register is a start trigger register for PWMCLKm (m = 0 to 3).

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWBAn_base> + 0014_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTS3	PWBAnTS2	PWBAnTS1	PWBAnTS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 30.13 PWBAnTS Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3	PWBAnTS3	Start Trigger for PWMCLK3 0: Writing 0 has no effect. 1: Starts the output of PWMCLK3.
2	PWBAnTS2	Start Trigger for PWMCLK2 0: Writing 0 has no effect. 1: Starts the output of PWMCLK2.
1	PWBAnTS1	Start Trigger for PWMCLK1 0: Writing 0 has no effect. 1: Starts the output of PWMCLK1.
0	PWBAnTS0	Start Trigger for PWMCLK0 0: Writing 0 has no effect. 1: Starts the output of PWMCLK0.

30.2.1.4 PWBAnTT Register

This register is a stop trigger register for PWMCLKm (m = 0 to 3).

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWBAn_base> + 0018_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTT3	PWBAnTT2	PWBAnTT1	PWBAnTT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 30.14 PWBAnTT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing, write the value after reset.
3	PWBAnTT3	Stop Trigger for PWMCLK3 0: Writing 0 has no effect. 1: Stops the output of PWMCLK3.
2	PWBAnTT2	Stop Trigger for PWMCLK2 0: Writing 0 has no effect. 1: Stops the output of PWMCLK2.
1	PWBAnTT1	Stop Trigger for PWMCLK1 0: Writing 0 has no effect. 1: Stops the output of PWMCLK1.
0	PWBAnTT0	Stop Trigger for PWMCLK0 0: Writing 0 has no effect. 1: Stops the output of PWMCLK0.

30.2.1.5 PWBAnEMU Register

This register sets the operation during emulation.

Access: This register can be read or written in 8-bit units.

Address: <PWBAn_base> + $001C_H$

Value after reset: 00_H

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

Table 30.15 PWBAnEMU Register Contents

Bit Position	Bit Name	Function
7	PWBAnSVSDIS	(When the EPC.SVSTOP bit = 0) The count clock is provided continuously when the debugger is controlling the microcontroller (by using break points, etc.), regardless of the value of this bit (1 or 0). (When the EPC.SVSTOP bit = 1) 0: The count clock is stopped when the debugger is controlling the microcontroller (by using break points, etc.). 1: The count clock is provided continuously when the debugger is controlling the microcontroller (by using break points, etc.). This bit can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0).
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

30.2.1.6 PWGAnCTL — PWGA Control Register

PWGAnCTL is used to select the count clock from PWBA.

Access: This register can be read or written in 8-bit units.

Address: <PWGAn_base> + 0020_H

Value after reset: 00_H

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

Table 30.16 PWGAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	PWGAnCKS [1:0]	Count Clock Enable Input PWMCLK3 to PWMCLK0 Select 00: Uses PWMCLK0 as count clock 01: Uses PWMCLK1 as count clock 10: Uses PWMCLK2 as count clock 11: Uses PWMCLK3 as count clock These bits can only be rewritten when the PWGAn operation is stopped (SLPWGAk.SLPWGA[31:0] = 0).

30.2.1.7 PWGAnCNT — PWM Cycle Count Register

This is a count register.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGAn_base> + 0014_H

Value after reset: $0FFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCNT[11:0]											
Value after reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.17 PWGAnCNT Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCNT [11:0]	12-bit counter value

30.2.1.8 PWGAnCSDR — PWM Output Set Condition Register

This register sets the setting condition for PWGA_TOUTn output.

Access: This register can be read or written in 16-bit units.

Address: <PWGAn_base> + 0000_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCSDR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.18 PWGAnCSDR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	PWGAnCSDR [11:0]	These bits set the setting condition for PWM output. The set value is reflected to the PWGAnCSBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

30.2.1.9 PWGAnCRDR — PWM Output Reset Condition Register

This register sets the reset condition for PWGA_TOUTn output.

Access: This register can be read or written in 16-bit units.

Address: <PWGAn_base> + 0004_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PWGAnCRDR[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.19 PWGAnCRDR Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12 to 0	PWGAnCRDR [12:0]	These bits set the reset condition for PWM output. The set value is reflected to the PWGAnCRBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

30.2.1.10 PWGAnCTDR — PWGA_TRGOUTn Generation Condition Register

This register sets the generation condition for PWGA_TRGOUTn.

Access: This register can be read or written in 16-bit units.

Address: <PWGAn_base> + 0008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCTDR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.20 PWGAnCTDR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
11 to 0	PWGAnCTDR [11:0]	These bits set the A/D conversion trigger generation condition for PWSAn. The set value is reflected to the PWGAnCTBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

30.2.1.11 PWGAnCSBR — PWGAnCSDR Buffer Register

This is a buffer register for the PWGAnCSDR register.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGAn_base> + 0024_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCSBR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.21 PWGAnCSBR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCSBR [11:0]	The set value is reflected to the PWGAnCSDR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, the pin output is driven high.

30.2.1.12 PWGAnCRBR — PWGAnCRDR Buffer Register

This is a buffer register for the PWGA_TOUTn reset condition.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGAn_base> + 0028_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PWGAnCRBR[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.22 PWGAnCRBR Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12 to 0	PWGAnCRBR [12:0]	The set value is reflected to the PWGAnCRDR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, the pin output is driven low.

30.2.1.13 PWGAnCTBR — PWGAnCTDR Buffer Register

This is a buffer register for the PWGA_TRGOUTn generation condition.

Access: This register is a read-only register that can be read in 16-bit units.

Address: <PWGAn_base> + $002C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCTBR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.23 PWGAnCTBR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCTBR [11:0]	The set value is reflected to the PWGAnCTDR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, a trigger is transmitted to PWSAn.

30.2.1.14 PWGAnRSF — Buffer Register Reload Status Register

This register is a status register for simultaneous rewrite control.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWGAn_base> + 0010_H

Value after reset: 00_H

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

Table 30.24 PWGAnRSF Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	PWGAnRSF	Simultaneous Rewrite Control Status 0: Simultaneous rewrite is enabled. This value indicates the completion of simultaneous rewrite after the generation of a simultaneous rewrite trigger signal. 1: Simultaneous rewrite is in progress. This value indicates the waiting state for completion.

30.2.1.15 PWGAnRDT — Buffer Register Reload Trigger Register

This is a simultaneous rewrite request trigger register.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWGAn_base> + $000C_H$

Value after reset: 00_H

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

Table 30.25 PWGAnRDT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	PWGAnRDT	Simultaneous Rewrite Request Trigger 0: Writing 0 has no effect. 1: Triggers the simultaneous rewrite request for the compare registers (PWGAnCSDR, PWGAnCRDR, and PWGAnCTDR), and sets PWGAnRSF.PWGAnRSF to 1.

30.2.1.16 SLPWGAk — PWGA Synchronous Trigger Register (k = 0 to 2)

This register triggers start and stop for multiple channels simultaneously.

Access: This register can be read or written in 32-bit units.

Address: FFE7 3000_H + k × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SLPWGA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SLPWGA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.26 SLPWGAk Register Contents

Bit Position	Bit Name	Function
31 to 0	SLPWGA [31:0]	Trigger start and stop to multiple channels simultaneously. 0: Stops the corresponding channels. 1: Starts the corresponding channels. The bits correspond to the following channels. SLPWGA0.SLPWGA[31:0]: PWGA31 - PWGA0 SLPWGA1.SLPWGA[31:0]: PWGA63 - PWGA32 SLPWGA2.SLPWGA[15:0]: PWGA79 - PWGA64

30.2.1.17 PWSAnCTL Register

This register is used to control operations of PWSA.

Access: This register can be read or written in 8-bit units.

Address: <PWSAn_base> + 0000_H

Value after reset: 00_H

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

Table 30.27 PWSAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWSAnENBL	Operation Permission Control 0: Operation is prohibited (initial state). Writing 0 initializes PWSAnSTR and PWSAnQUEj. 1: Operation is enabled.

30.2.1.18 PWSAnSTR Register

This is a status register that indicates whether the number of a channel for which an A/D conversion trigger has been generated is stored in a PWSAnQUEj register.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWSAn_base> + 0004_H

Value after reset: 00_H

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

Table 30.28 PWSAnSTR Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	PWSAnQFL	Indicates the queuing state of the A/D conversion trigger. 0: Some PWSAnQUEj registers do not store a channel number. 1: All of the PWSAnQUEj registers store a channel number.
0	PWSAnQNE	Bit indicating that there is a trigger in the trigger queue 0: A channel number is not stored in a PWSAnQUEj register, or A/D conversion is in progress while only PWSAnQUE0 stores a channel number. 1: The number of the channel waiting for conversion is stored in j = 1 and subsequent PWSAnQUEj registers.

30.2.1.19 PWSAnSTC Register

This register clears the status of the PWSAnSTR register.

Access: This register is a write-only register that can be written in 8-bit units.

Address: <PWSAn_base> + 0008_H

Value after reset: 00_H

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

Table 30.29 PWSAnSTC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing, write the value after reset.
1	PWSAnCLFL	PWSAnQFL Clear 0: PWSAnQFL retains the status (Writing 0 has no effect). 1: PWSAnQFL is cleared to 0.
0	PWSAnCLNE	PWSAnQNE Clear 0: PWSAnQNE retains the status (Writing 0 has no effect). 1: PWSAnQNE is cleared to 0.

30.2.1.20 PWSAnQUEj (j = 0 to 7) Register

This register stores the channel number that received the trigger from PWGAn.

Access: This register is a read-only register that can be read in 8-bit units.

Address: <PWSAn_base> + $0020_H + j \times 4_H$

Value after reset: $7F_H$

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

Table 30.30 PWSAnQUEj Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	PWSAnQUEj [6:0]	These bits hold the channel number (0 to 79) of the PWGA for which a trigger was generated in order from PWSAnQUE0 to PWSAnQUE7. After the A/D conversion of PWSAnQUE0 is completed, the values in PWSAnQUE1 to PWSAnQUE7 shift to PWSAnQUE0 to PWSAnQUE6.

NOTE

If a trigger occurs simultaneously for multiple channels, the trigger with the smaller channel number has priority.

30.2.1.21 PWSAnPVCRx_y (x = 00, 02, 04 ... 78, y = 01, 03, 05 ... 79) Register

This register is used to set the corresponding A/D converter for each channel.

Two consecutive channels are set such as PWSA0PVCR02_03, and the 16 higher-order bits of each register correspond to an odd-numbered channel while the 16 lower-order bits correspond to an even-numbered channel.

At the generation of a trigger, the set value is transmitted to the ADCAnPWDVCR register of the A/D converter.

For the ADCAnPWDVCR register, see **Section 31, A/D Converter (ADCA)**.

Access: This register can be read or written in 32-bit units.

Address: <PWSAn_base> + 0040_H + x × 2_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	PWSAnSLADy	PWSAnVRDTy[27]	PWSAnVRDTy[26:24]			PWSAnVRDTy[23:22]		PWSAnVRDTy[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PWSAnSLADx	PWSAnVRDTx[11]	PWSAnVRDTx[10:8]			PWSAnVRDTx[7:6]		PWSAnVRDTx[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.31 PWSAnPVCRx_y Register Contents

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
28	PWSAnSLADy	ADCA Select (odd-numbered channel) 0: Output to ADCA0. 1: Output to ADCA1.
27	PWSAnVRDTy [27]	This bit indicates the set value of the ADCAnPWDVCR.MPXE bit (odd-numbered channel).
26 to 24	PWSAnVRDTy [26:24]	These bits indicate the set value of the ADCAnPWDVCR.MPXV[2:0] bits (odd-numbered channel).
23, 22	PWSAnVRDTy [23:22]	These bits indicate the set value of the ADCAnPWDVCR.ULS[1:0] bits (odd-numbered channel).
21 to 16	PWSAnVRDTy [21:16]	These bits indicate the set value of the ADCAnPWDVCR.GCTRL[5:0] bits (odd-numbered channel).
15 to 13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	PWSAnSLADx	ADCA Select (even-numbered channel) 0: Output to ADCA0. 1: Output to ADCA1.
11	PWSAnVRDTx [11]	This bit indicates the set value of the ADCAnPWDVCR.MPXE bit. (even-numbered channel)
10 to 8	PWSAnVRDTx [10:8]	These bits indicate the set value of the ADCAnPWDVCR.MPXV[2:0] bits. (even-numbered channel)
7, 6	PWSAnVRDTx [7:6]	These bits indicate the set value of the ADCAnPWDVCR.ULS[1:0] bits. (even-numbered channel)
5 to 0	PWSAnVRDTx [5:0]	These bits indicate the set value of the ADCAnPWDVCR.GCTRL[5:0] bits. (even-numbered channel)

30.2.1.22 PWSAnEMU — Emulation Control Register

This register is used to set the operation for emulation.

Access: This register can be read or written in 8-bit units.

Address: <PWSAn_base> + $000C_H$

Value after reset: 00_H

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

Table 30.32 PWSAnEMU Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWSAnSVSDIS	(When the EPC.SVSTOP bit = 0) The operation continues when the debugger is controlling the microcontroller (by using break points, etc.), regardless of the value of this bit (1 or 0). (When the EPC.SVSTOP bit = 1) 0: When the debugger is controlling the microcontroller (by using break points, etc.); – The output state to A/D is retained, the ADC_CONV_ENDn input at a break point is internally retained, and PWSAnQUEj is updated after break release. – The PWGA_TRGOUT input is accepted even at a break, and PWSA_INT_QFULL is also output. – Reading and writing to the register is possible. 1: The operation continues when the debugger is controlling the microcontroller (by using break points, etc.). The above bit can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0), the operation of all channels PWGAn has stopped (SLPWGAk.SLPWGA), and no trigger has been generated from any of the channels PWGAn (PWSAnQUE0 is the value after reset).

30.3 Operating Procedure

Procedures for setting when starting and stopping operation of PWM-Diag are illustrated below.

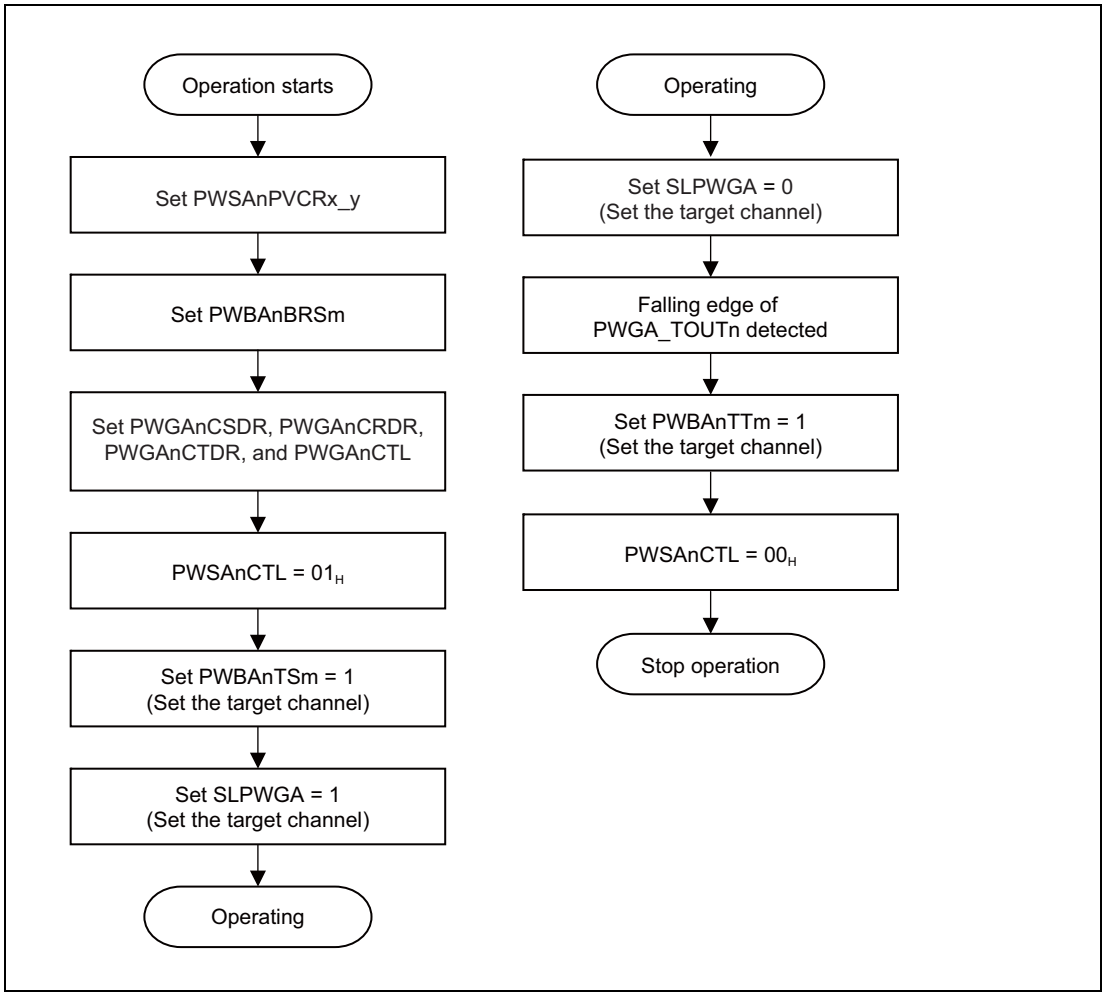
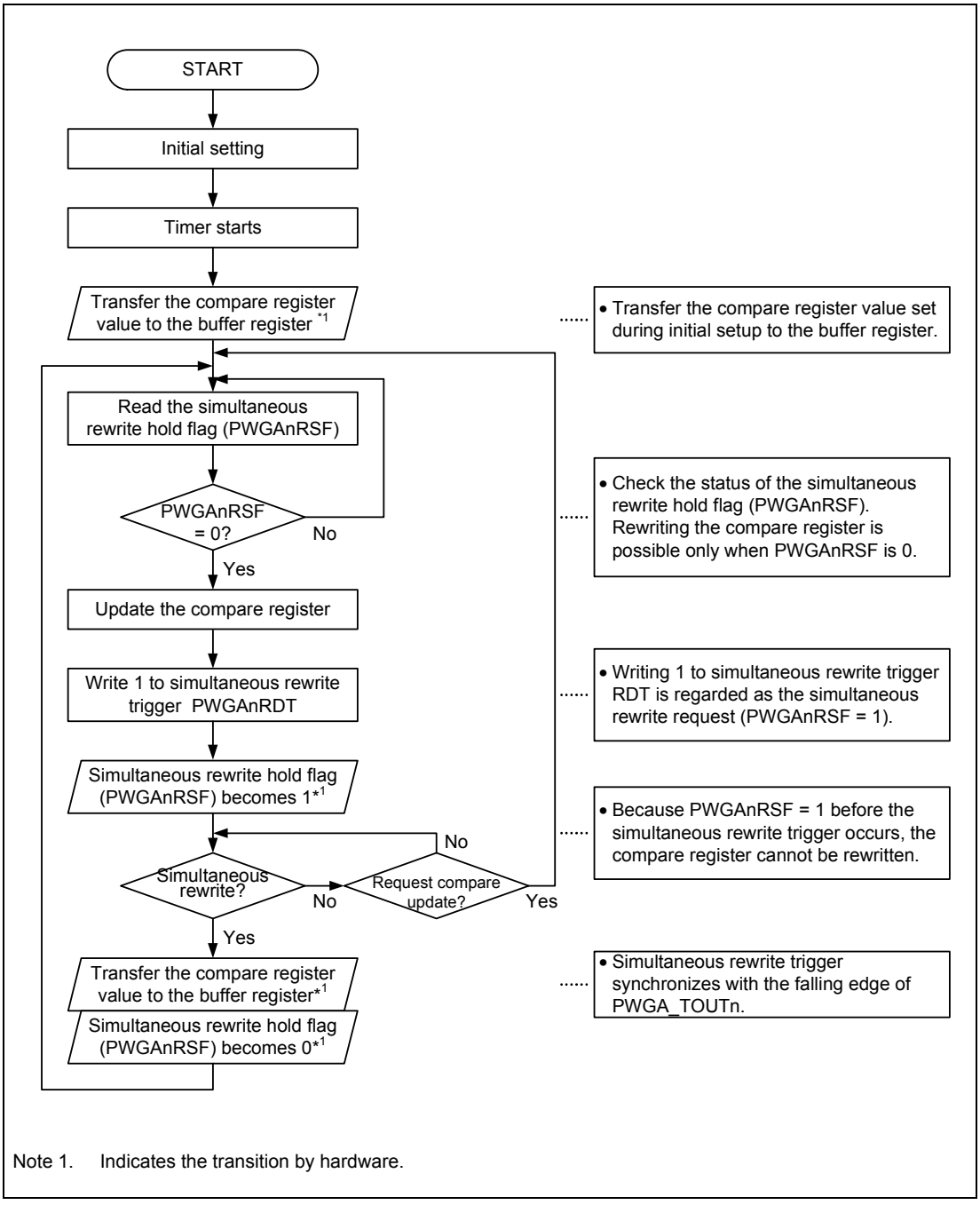


Figure 30.2 PWM-Diag Operating Procedure

Procedures for simultaneous rewrite of PWGA are illustrated below.

The described term “compare register” indicates PWGAnCSDR, PWGAnCRDR, or PWGAnCTDR.

In addition, the described term “buffer register” indicates PWGAnCSBR, PWGAnCRBR, or PWGAnCTBR.



Note 1. Indicates the transition by hardware.

Figure 30.3 Simultaneous Rewrite Procedure

30.4 Operation Waveform of PWM-Diag

30.4.1 PWM Waveform Output by PWGA and Operation Waveform for A/D Conversion Trigger Output

30.4.1.1 Basic Operation Waveform of PWGA

The basic operation waveforms of PWGA are illustrated below.

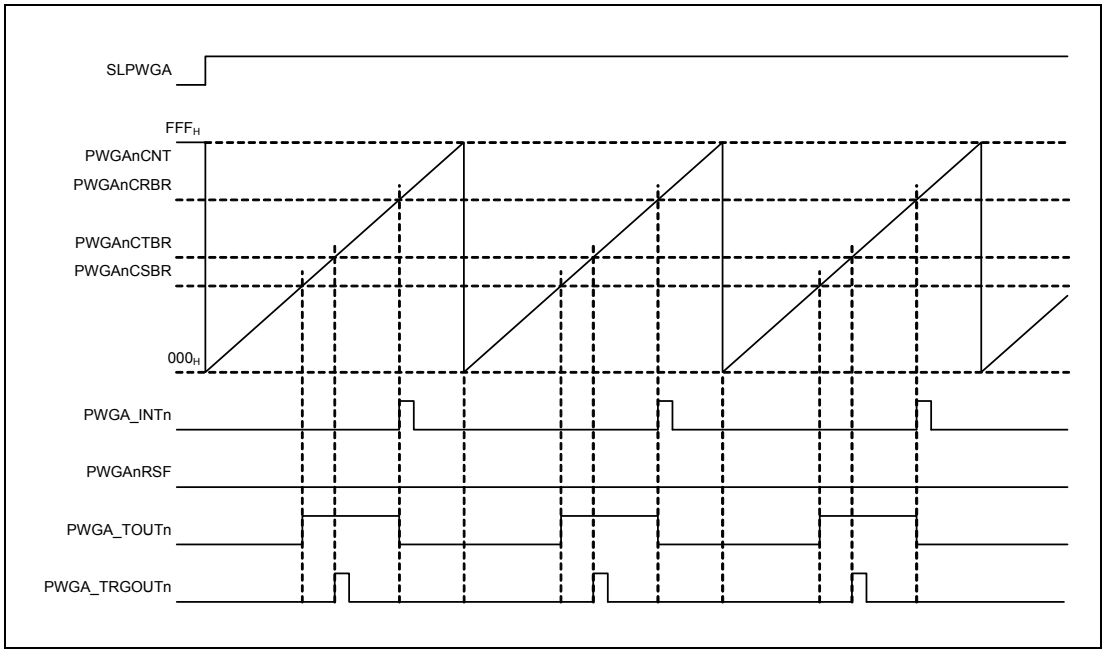


Figure 30.4 Basic Waveform

30.4.1.2 Operation Waveform when Simultaneous Rewrite for PWGA is Executed

The following figure illustrates the operation waveforms when simultaneous rewrite for PWGA is executed.

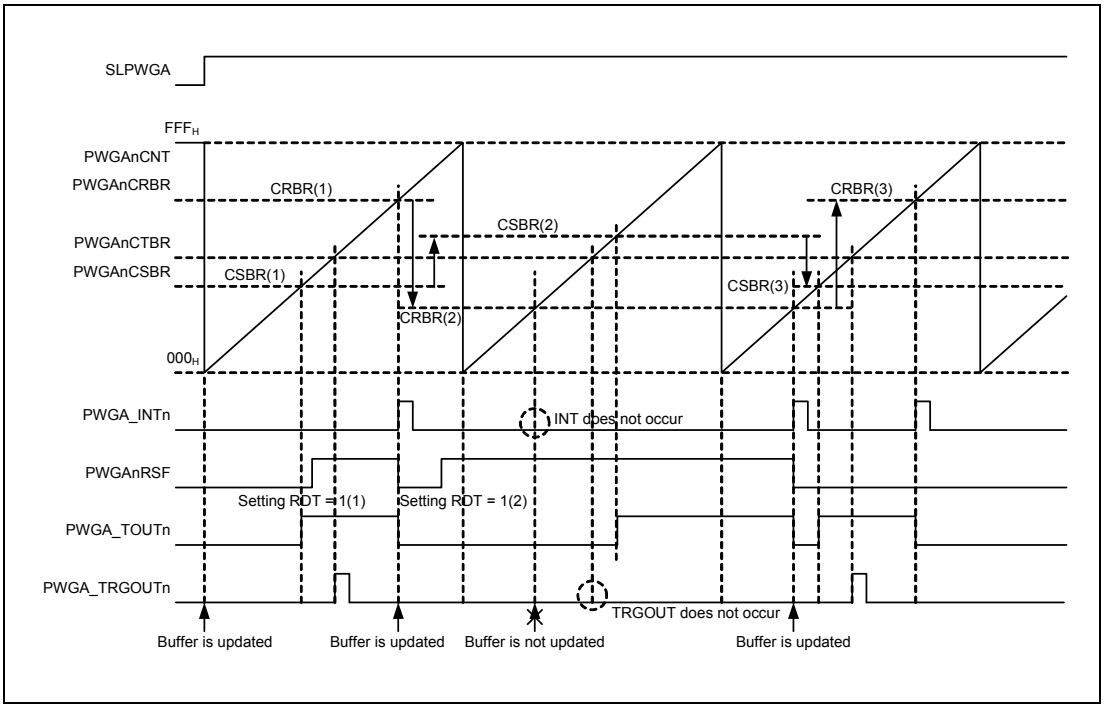


Figure 30.5 Waveform when Simultaneous Rewrite is Executed

Simultaneous rewrite is executed by re-setting the PWGAnCSDR and PWGAnCRDR registers, then setting either the PWGAnRDT or SLPWGAk register.

Moreover, if the relationship between set values in one interval is PWGAnCSDR > PWGAnCRDR, a falling edge in that interval is meaningless, and the falling edge in the next interval is valid.

Moreover, PWGA_TRGOUTn does not become valid unless PWGA_TOUTn is at the high level.

30.4.1.3 Operation Waveform when Stopping and Resuming PWGA Operation

The following figure illustrates the operation waveforms when stopping and resuming PWGA operation.

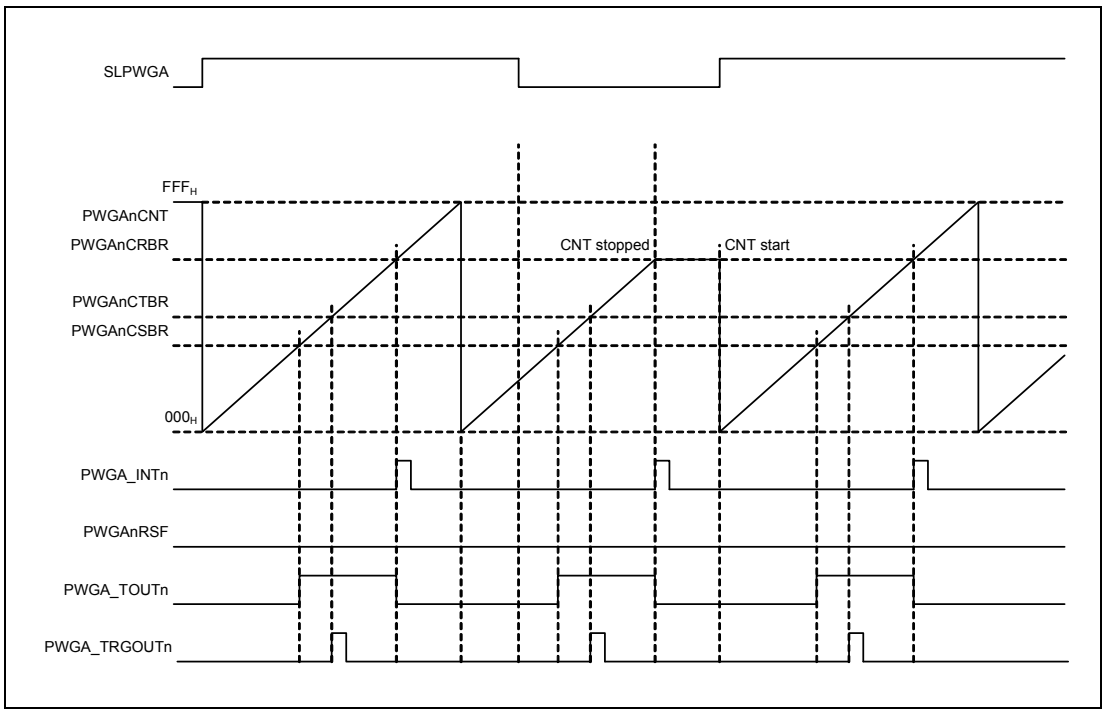


Figure 30.6 Stopping and Resuming Operation (1)

After the setting of SLPWGA has been changed from 1 to 0, PWGAnCNT stops operation because PWGA_INTn is generated.

After PWGA_INTn has been generated, by changing the setting of SLPWGA from 0 to 1, PWGAnCNT resumes counting from 000_H.

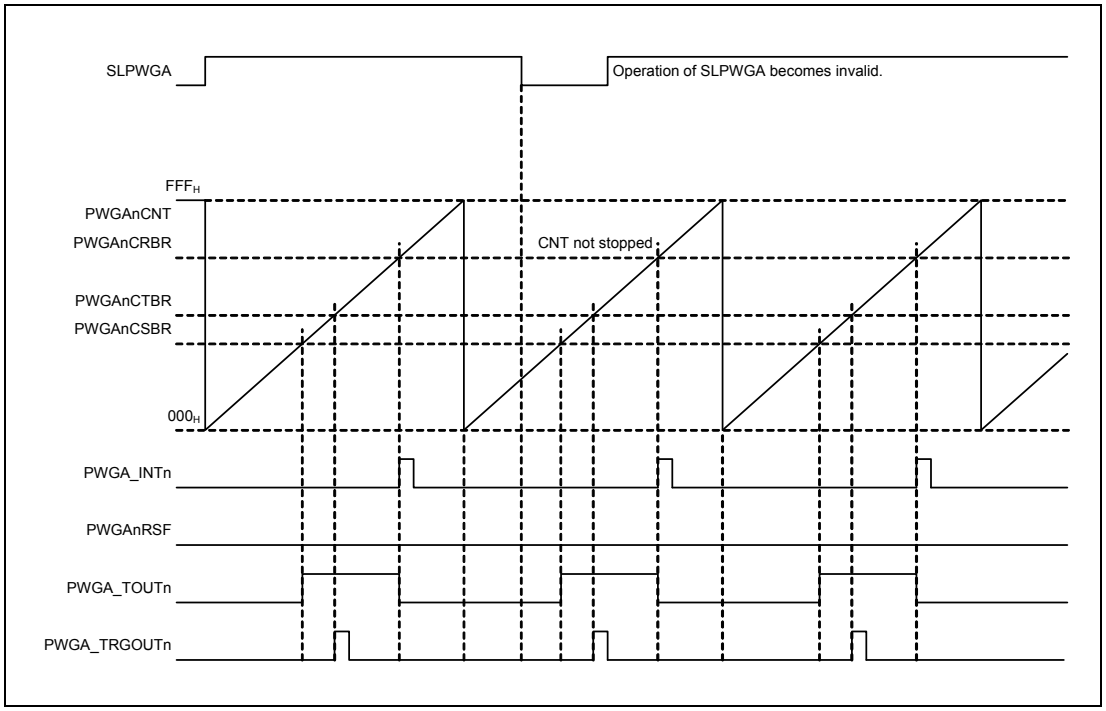


Figure 30.7 Stopping and Resuming Operation (2)

After the setting of SLPWGA has been changed from 1 to 0, if the setting of SLPWGA is changed from 0 to 1 before PWGA_INTn is generated, operations of SLPWGA become invalid, and PWGAnCNT continues counting.

30.4.1.4 Waveforms of PWGA Operation with Specific Settings

The following figures illustrate the waveforms of PWGA operation with specific settings.

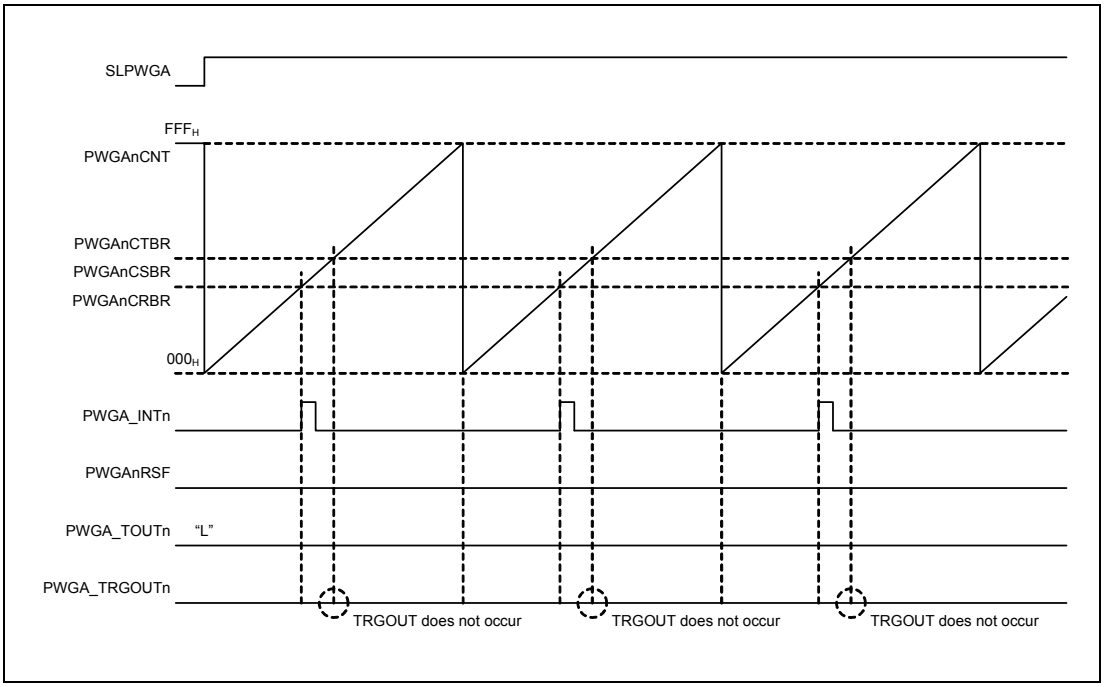


Figure 30.8 PWGA_TOUTn = 0% Output Waveform

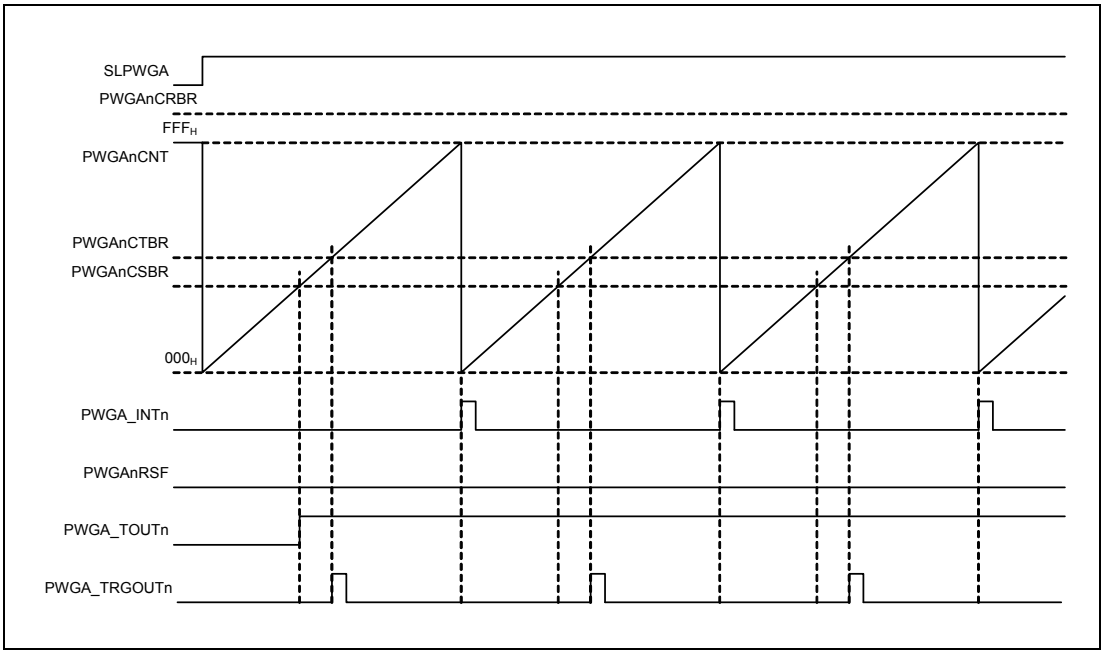
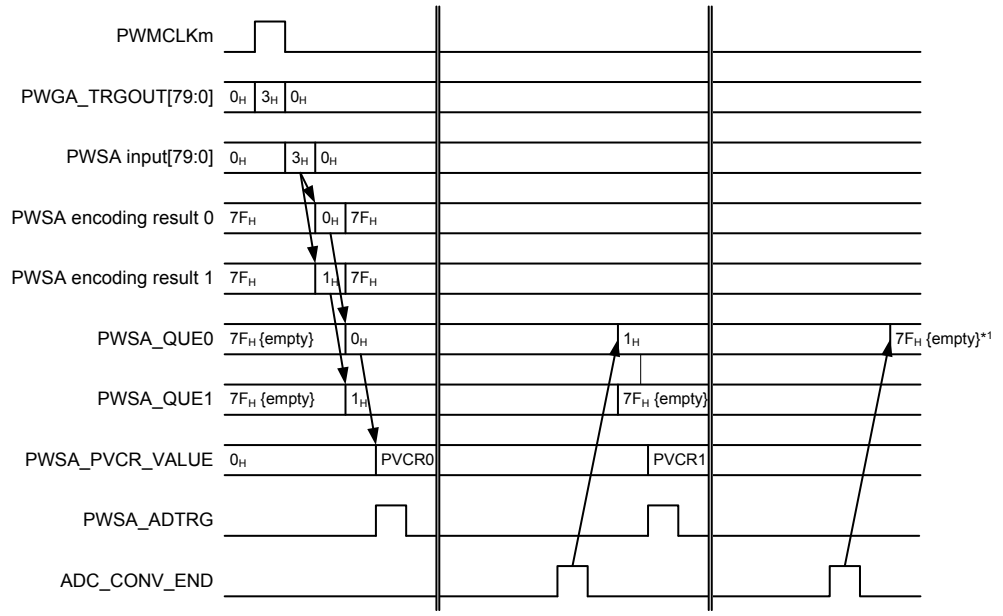


Figure 30.9 PWGA_TOUTn = 100% Output Waveform

30.4.2 Operation Waveform when A/D Conversion Trigger Occurs in PWSA

An example of the PWSA operation is shown below.



Note 1. QUE0 indicates the PWGA_TRG number while the A/D conversion is executed. The initial value (while the A/D conversion is not executed) is $7F_H$.

Figure 30.10 Example of PWSA Operation

(1) Triggers occur simultaneously in channels 0 and 1 of PWGA. Channel 0 with the smaller channel number is stored in PWSAnQUE0, and channel 1 with the larger channel number is stored in PWSAnQUE1. The lower 16-bit data of PWSAnPVCR00_01 corresponding to the value stored in PWSAnQUE0 is transmitted to the A/D converter and a trigger is output to the A/D converter. At this time, as the A/D conversion for channel 1 is in the waiting state, the PWSAnSTR.PWSAnQNE bit is set.

(2) On completion of A/D conversion executed in step (1), the channel number of PWSAnQUE1 shifts to PWSAnQUE0 and PWSAnQUE1 enters the empty state.
After that, as similar to step (1), the upper 16-bit data of PWSAnPVCR00_01 corresponding to the value stored in PWSAnQUE0 is transmitted to the A/D converter and a trigger is output to the A/D converter.

(3) On completion of A/D conversion executed in step (2), PWSAnQUE0 enters the empty state.

30.5 PWM-Diag Related Functions in A/D Converter (ADCA)

This section describes the A/D converter used for the PWM-Diag function.

30.5.1 ADCA registers when the PWM-Diag function is used

- Before starting PWSA operation, the A/D converter must be set using the following register.
  - PWM-Diag scan group control register (ADCAnPWDSGCR)
- When the PWM-Diag is running, the PWSAnPVCRx_y value corresponding to the channel under conversion is set in the following register of the A/D converter.
  - PWM-Diag virtual channel register (ADCAnPWDVCR)
- After completion of A/D conversion, the conversion result can be checked by reading the following registers.
  - PWM-Diag data register (ADCAnPWDTSNDR)
  - PWM-Diag data supplementary information register (ADCAnPWDDIR)
- When A/D conversion result is outside the expected range, it can be confirmed using the upper/lower limit error detection function. The upper/lower limit error detection function is set by the following register.
  - Upper limit/lower limit error register (ADCAnULER)
- The scan end flag of the PWM-Diag scan group can be cleared using the following register.
  - PWM-Diag scan end flag clear register (ADCAnPWDSGSEFCR)

Section 31 A/D Converter (ADCA)

This section contains a generic description of the A/D Converter (ADCA).

The first part of this section describes all RH850/F1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the ADCA.

31.1 Features of RH850/F1M ADCA

31.1.1 Number of Units and Channels

This microcontroller has the following number of ADCA units.

Table 31.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	2		
Name	ADCAn (n = 0, 1)		

An ADCAn unit has the same number of physical channels as the number of A/D input pins and the same number of virtual channels as the number of addresses where the results of A/D conversion will be stored. The numbers of channels on individual products are as listed below.

Table 31.2 Unit Configurations and Physical Channels

Unit Name (Number of Channels) ADCAn		RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
ADCA0	12 bit pin for conversion*1	16		
	10 bit pin for conversion*2	18		
ADCA1	12 bit pin for conversion*1	8	16	
	10 bit pin for conversion*2	4	8	20

Note 1. When 10-bit mode is selected, this pin can be used for 10-bit conversion.

Note 2. When 12-bit mode is selected but a pin is for 10-bit conversion, the 2 low-order bits of the result of conversion must be masked before use.

Table 31.3 Unit Configurations and Virtual Channels

Unit Name (Number of Channels) ADCAn	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
ADCA0	50		
ADCA1	12	24	36

Table 31.4 Indices

Index	Description
n	Throughout this section, the individual ADCA units are identified by the index “n” (n = 0, 1); for example, ADCAnPWDVCR indicates the PWM-Diag virtual channel register.
m	Throughout this section, the individual physical channels (channels in the unit) of ADCAn are identified by the index “m”; for example, ANInm.
j	Throughout this section, the individual virtual channels of ADCAn are identified by the index “j”; for example, ADCAnVCRj indicates the virtual channel register.
x	Throughout this section, the individual scan groups (SG) of ADCAn are identified by the index “x” (x = 1 to 3); for example, ADCAnSGSTCRx indicates the scan group x start control register.
k	Throughout this section, the individual physical channel numbers for T&H are identified by the index “k” (k = 0 to 5); for example, THkE is T&H enable bit of the T&H enable register (ADCAnTHER).

The following table shows values indicated by the indices of each product.

Table 31.5 Indices of Products

Indices of Each Product		
144 pins	**176 pins**	**233 pins**
m = 0 to 27, 30 to 35 (ADCA0) m = 0 to 11 (ADCA1)	m = 0 to 27, 30 to 35 (ADCA0) m = 0 to 23 (ADCA1)	m = 0 to 27, 30 to 35 (ADCA0) m = 0 to 35 (ADCA1)
j = 0 to 49 (ADCA0) j = 0 to 11 (ADCA1)	j = 0 to 49 (ADCA0) j = 0 to 23 (ADCA1)	j = 0 to 49 (ADCA0) j = 0 to 35 (ADCA1)
x = 1 to 3		
k = 0 to 5		

31.1.2 Register Base Address

ADCAn base addresses are listed in the following table.

ADCAn register addresses are given as offsets from the base addresses in general.

Table 31.6 Register Base Addresses

Base Address Name	Base Address
<ADCA0_base>	FFF2 0000_H
<ADCA1_base>	FFD6 $D000_H$

31.1.3 Clock Supply

The ADCAn clock supply is shown in the following table.

Table 31.7 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
ADCA0	ADCLK	CKSCLK_AADCA
	Register access clock	CKSCLK_AADCA
ADCA1	ADCLK	CKSCLK_IADCA
	Register access clock	CPUCLK2

31.1.4 Interrupt Requests

ADCAn interrupt requests are listed in the following table.

Table 31.8 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number	Other Trigger Signals
ADCA0				
INT_ADE	A/D error interrupt	56	—	Motor control
INT_SG1	Scan group 1 (SG1) end interrupt	18	4	LPS
INT_SG2	Scan group 2 (SG2) end interrupt	19	5	LPS
INT_SG3	Scan group 3 (SG3) end interrupt	20, 32	6	LPS
ADC_CONV_END0	Scan group 4 (SG4) A/D conversion end signal	—	7	—
ADCA1				
INT_ADE	A/D error interrupt	212	—	—
INT_SG1	Scan group 1 (SG1) end interrupt	213	103	—
INT_SG2	Scan group 2 (SG2) end interrupt	214	104	—
INT_SG3	Scan group 3 (SG3) end interrupt	215	105	—
ADC_CONV_END1	Scan group 4 (SG4) A/D conversion end signal	—	106	—

31.1.5 Reset Sources

ADCAn reset sources are listed in the following table. ADCAn is initialized by these reset sources.

Table 31.9 Reset Sources

Unit Name	Reset Source
ADCA0	Reset sources other than transition to DeepSTOP mode (AWORES)
ADCA1	All reset sources (ISORES)

31.1.6 External Input/Output Signals

External input/output signals of ADCAn are listed below.

Table 31.10 ADCA0 External Input/Output Signals (1/2)

Unit Signal Name	Description	Alternative Port Pin Signal
ADCA0		
A0Vss	Ground pin for the analog part	A0Vss
A0VREF	Power supply and reference voltage pin for the analog part	A0VREF
ANI000	12-bit resolution analog input pin (for T&H)	ADCA0I0
ANI001	12-bit resolution analog input pin (for T&H)	ADCA0I1
ANI002	12-bit resolution analog input pin (for T&H)	ADCA0I2
ANI003	12-bit resolution analog input pin (for T&H)	ADCA0I3
ANI004	12-bit resolution analog input pin (for T&H)	ADCA0I4
ANI005	12-bit resolution analog input pin (for T&H)	ADCA0I5
ANI006	12-bit resolution analog input pin	ADCA0I6
ANI007	12-bit resolution analog input pin	ADCA0I7
ANI008	12-bit resolution analog input pin	ADCA0I8
ANI009	12-bit resolution analog input pin	ADCA0I9
ANI010	12-bit resolution analog input pin	ADCA0I10
ANI011	12-bit resolution analog input pin	ADCA0I11
ANI012	12-bit resolution analog input pin	ADCA0I12
ANI013	12-bit resolution analog input pin	ADCA0I13
ANI014	12-bit resolution analog input pin	ADCA0I14
ANI015	12-bit resolution analog input pin	ADCA0I15
ANI016	10-bit resolution analog input pin	ADCA0I0S
ANI017	10-bit resolution analog input pin	ADCA0I1S
ANI018	10-bit resolution analog input pin	ADCA0I2S
ANI019	10-bit resolution analog input pin	ADCA0I3S
ANI020	10-bit resolution analog input pin	ADCA0I4S
ANI021	10-bit resolution analog input pin	ADCA0I5S
ANI022	10-bit resolution analog input pin	ADCA0I6S
ANI023	10-bit resolution analog input pin	ADCA0I7S
ANI024	10-bit resolution analog input pin	ADCA0I8S
ANI025	10-bit resolution analog input pin	ADCA0I9S
ANI026	10-bit resolution analog input pin	ADCA0I10S
ANI027	10-bit resolution analog input pin	ADCA0I11S
ANI028	10-bit resolution analog input pin	*1
ANI029	10-bit resolution analog input pin	*1
ANI030	10-bit resolution analog input pin	ADCA0I14S
ANI031	10-bit resolution analog input pin	ADCA0I15S
ANI032	10-bit resolution analog input pin	ADCA0I16S
ANI033	10-bit resolution analog input pin	ADCA0I17S
ANI034	10-bit resolution analog input pin	ADCA0I18S
ANI035	10-bit resolution analog input pin	ADCA0I19S

Table 31.10 ADCA0 External Input/Output Signals (2/2)

Unit Signal Name	Description	Alternative Port Pin Signal
ADCA0TRG0	External trigger pin (scan group 1)*2	ADCA0TRG0
ADCA0TRG1	External trigger pin (scan group 2)*2	ADCA0TRG1
ADCA0TRG2	External trigger pin (scan group 3)*2	ADCA0TRG2
ADCA0SEL0	External analog multiplexer (MPX) output pin 0	ADCA0SEL0
ADCA0SEL1	External analog multiplexer (MPX) output pin 1	ADCA0SEL1
ADCA0SEL2	External analog multiplexer (MPX) output pin 2	ADCA0SEL2

Note 1. This is not supported in the RH850/F1M product.

Note 2. When the external trigger pin is used, the noise filter for the port needs to be set. For details, refer to **Section 2.12, Noise Filter & Edge/Level Detector**.

CAUTION

The RH850/F1M does not have ANI028 (ADCA0I12S) and ANI029 (ADCA0I13S) pins, so do not attempt writing to the corresponding bits or controlling the signals.

For descriptions in this section that mention ANI028 (ADCA0I12S) and ANI029 (ADCA0I13S), read them as stating that ANI028 (ADCA0I12S) and ANI029 (ADCA0I13S) are not supported.

Table 31.11 ADCA1 External Input/Output Signals (1/2)

Unit Signal Name	Description	Alternative Port Pin Signal
ADCA1		
A1Vss	Ground pin for the analog part	A1Vss
A1VREF	Power supply and reference voltage pin for the analog part	A1VREF
ANI100	12-bit resolution analog input pin	ADCA1I0
ANI101	12-bit resolution analog input pin	ADCA1I1
ANI102	12-bit resolution analog input pin	ADCA1I2
ANI103	12-bit resolution analog input pin	ADCA1I3
ANI104	12-bit resolution analog input pin	ADCA1I4
ANI105	12-bit resolution analog input pin	ADCA1I5
ANI106	12-bit resolution analog input pin	ADCA1I6
ANI107	12-bit resolution analog input pin	ADCA1I7
ANI108	12-bit resolution analog input pin	ADCA1I8
ANI109	12-bit resolution analog input pin	ADCA1I9
ANI110	12-bit resolution analog input pin	ADCA1I10
ANI111	12-bit resolution analog input pin	ADCA1I11
ANI112	12-bit resolution analog input pin	ADCA1I12
ANI113	12-bit resolution analog input pin	ADCA1I13
ANI114	12-bit resolution analog input pin	ADCA1I14
ANI115	12-bit resolution analog input pin	ADCA1I15
ANI116	10-bit resolution analog input pin	ADCA1I0S
ANI117	10-bit resolution analog input pin	ADCA1I1S
ANI118	10-bit resolution analog input pin	ADCA1I2S

Table 31.11 ADCA1 External Input/Output Signals (2/2)

Unit Signal Name	Description	Alternative Port Pin Signal
ANI119	10-bit resolution analog input pin	ADCA1I3S
ANI120	10-bit resolution analog input pin	ADCA1I4S
ANI121	10-bit resolution analog input pin	ADCA1I5S
ANI122	10-bit resolution analog input pin	ADCA1I6S
ANI123	10-bit resolution analog input pin	ADCA1I7S
ANI124	10-bit resolution analog input pin	ADCA1I8S
ANI125	10-bit resolution analog input pin	ADCA1I9S
ANI126	10-bit resolution analog input pin	ADCA1I10S
ANI127	10-bit resolution analog input pin	ADCA1I11S
ANI128	10-bit resolution analog input pin	ADCA1I12S
ANI129	10-bit resolution analog input pin	ADCA1I13S
ANI130	10-bit resolution analog input pin	ADCA1I14S
ANI131	10-bit resolution analog input pin	ADCA1I15S
ANI132	10-bit resolution analog input pin	ADCA1I16S
ANI133	10-bit resolution analog input pin	ADCA1I17S
ANI134	10-bit resolution analog input pin	ADCA1I18S
ANI135	10-bit resolution analog input pin	ADCA1I19S
ADCA1TRG0	External trigger pin (scan group 1)*1	ADCA1TRG0
ADCA1TRG1	External trigger pin (scan group 2)*1	ADCA1TRG1
ADCA1TRG2	External trigger pin (scan group 3)*1	ADCA1TRG2

Note 1. When the external trigger pin is used, the noise filter for the port needs to be set. For details, refer to **Section 2.12, Noise Filter & Edge/Level Detector**.

31.2 Overview

31.2.1 Functional Overview

ADCA has the following features.

- 10-bit/12-bit resolution
- Successive approximation conversion method
- Number of A/D input channels
  A/D conversion is available for a maximum of 34 ADCA0 channels and 36 ADCA1 channels. Additionally, ADCA0 supports the connection of an external analog multiplexer (MPX) to extend the number of analog input channels.
- Internal track and hold (T&H) circuit
  ANI000 to ANI005 (ADCA0I0 to ADCA0I5) of ADCA0 include the track and hold circuit. The track and hold circuit can sample up to 6 channels of analog input simultaneously.
- A/D conversion control by scan groups
  The A/D conversion channel or conversion mode (scan mode) can be set for each scan group.
- Two scan modes
  Multi-cycle scan mode: Specified number of scans are executed.
  Continuous scan mode: Scans are executed repeatedly and continuously.
- Asynchronous/synchronous suspend and resume function
  A processing for a scan group can be interrupted to run the processing for another scan group.
- Start trigger for each scan group
  Software, hardware, and external trigger can start processing of each scan group.
- Scan end interrupt and DMA transfer are supported.
  For each scan group, an interrupt request to INTC can be issued or DMA transfer can be started, each time a processing for the virtual channel indicated by the end virtual channel pointer ends, or a virtual channel ends.
- A/D conversion channel repeat function
  A/D conversion is performed for the same channel two or four times sequentially, and the result is stored in the data register.
- Abundant safety functions
  Abundant safety functions are provided, such as A/D converter diagnosis, diagnosis of the channel multiplexer, diagnosis of open pins, diagnosis of the T&H circuit, upper limit/lower limit check for the A/D converter, overwrite check for data registers, and read and clear function for data registers.
- Shortest A/D conversion time per channel
  1.15 μs (when MPX is not used)
  2.30 μs (when MPX is used)

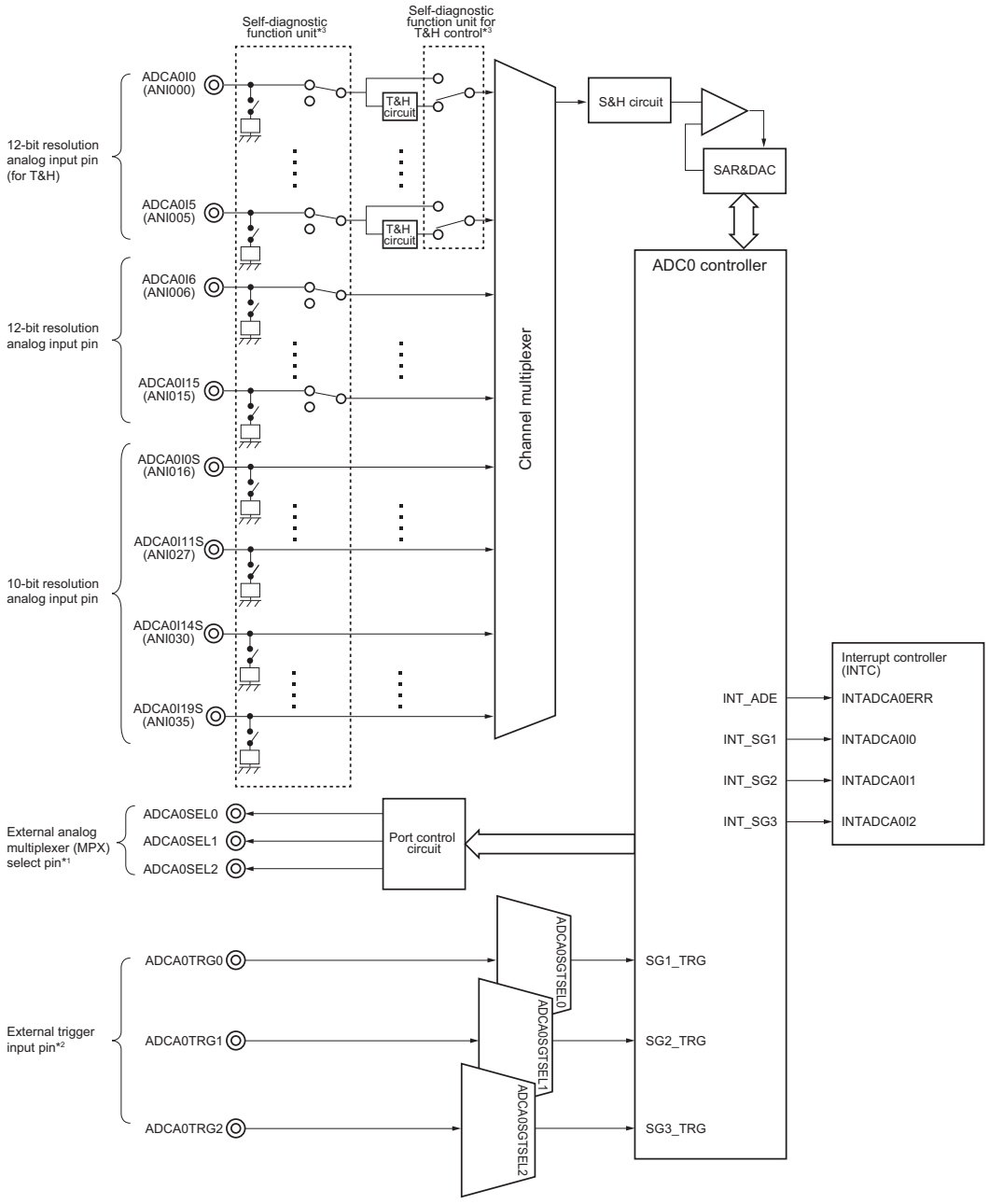
NOTE

- Physical channel (ANInm)
  Each A/D input channel of ADCA0 and ADCA1 units is called a physical channel. The physical channel of each unit is represented as ANI0m (m = 0 to 27, 30 to 35) for ADCA0 and ANI1m (m = 0 to 35) for ADCA1.
  In RH850/F1M, the alternative port pins for 12-bit resolution A/D input channel and 10-bit resolution A/D input channel are represented as ADCAnIm and ADCAnImS, respectively. In this section, the physical channels and the corresponding alternative port pins are listed together.
- Virtual channel (ADCAnVCRj)
  ADCA0 has a maximum of 50 virtual channels, and ADCA1 has a maximum of 36 virtual channels. The virtual channel specifies the physical channel to be scanned.
  Scans are executed in sequence from the smallest virtual channel number. The scan order can be arbitrarily-specified by using virtual channels. In addition, the scanned result is stored in the data register (ADCAnDRj) corresponding to the virtual channel.
- Scan group (SGx)
  ADCA has three scan groups (SG1, SG2, SG3) and one PWM-Diag group (SG4). A/D conversion is executed in scan group unit. The channel to be scanned can be selected for each group by specifying the scan range, that is, the conversion start virtual channel and the conversion end virtual channel.

31.2.2 Block Diagram

The block diagram of ADCA0 is shown in **Figure 31.1**. The block diagram of ADCA1 is shown in **Figure 31.2**.

(1) Configuration of ADCA0



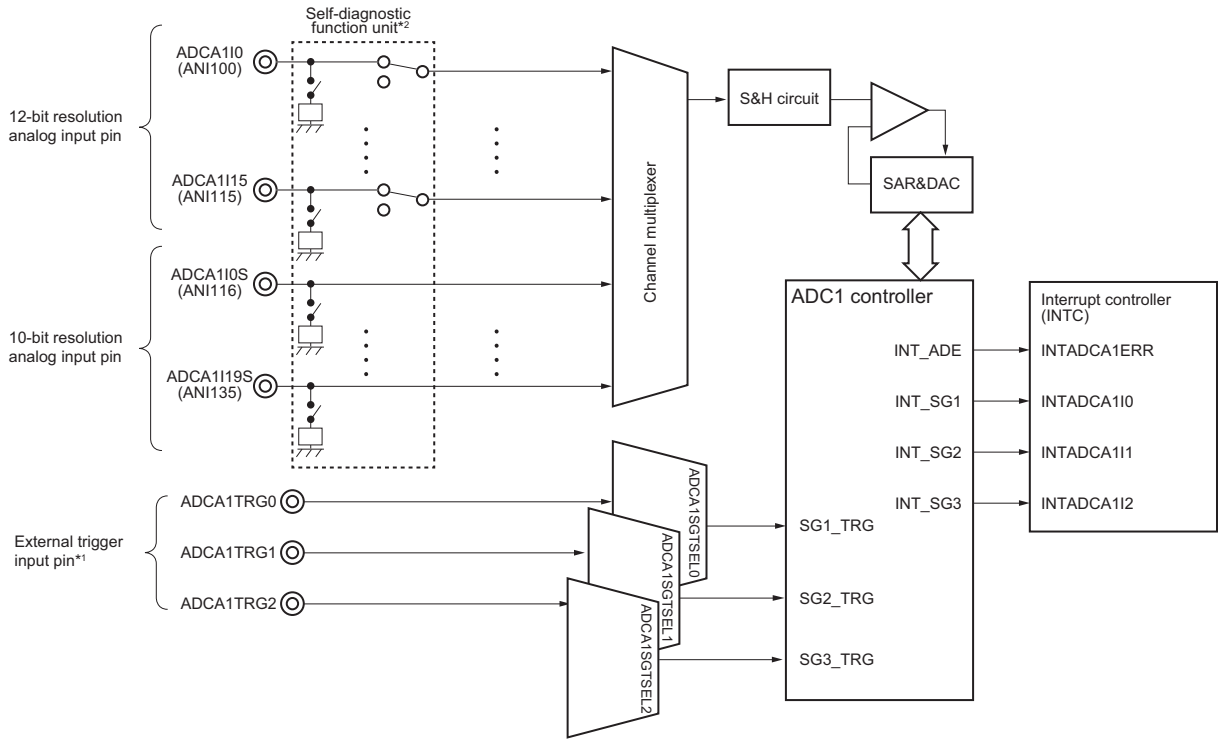
Note 1. For an example of external analog multiplexer connection, see **Figure 31.4, Example of External Analog Multiplexer Connection**.

Note 2. For details, see **Figure 31.3, Internal Connection Diagram of External Trigger Input Pins**.

Note 3. For the self-diagnostic function unit, see **Section 31.5, Self-Diagnostic Function**.

Figure 31.1 ADCA0 Block Diagram

(2) Configuration of ADCA1



Note 1. For details, see **Figure 31.3, Internal Connection Diagram of External Trigger Input Pins**.

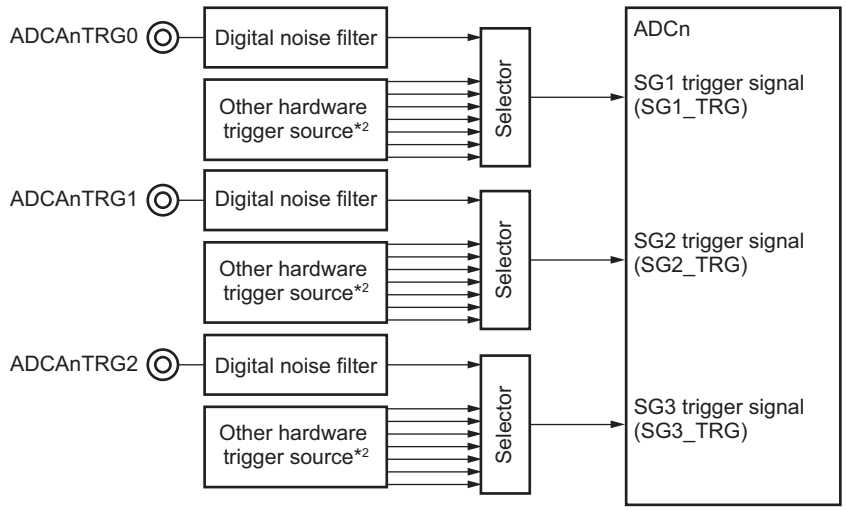
Note 2. For the self-diagnostic function unit, see **Section 31.5, Self-Diagnostic Function**.

Figure 31.2 ADCA1 Block Diagram

(3) Configuration of external trigger input pins

An external trigger input pin is a hardware trigger source to activate ADCAn.

The configuration of external trigger input pins is shown below.



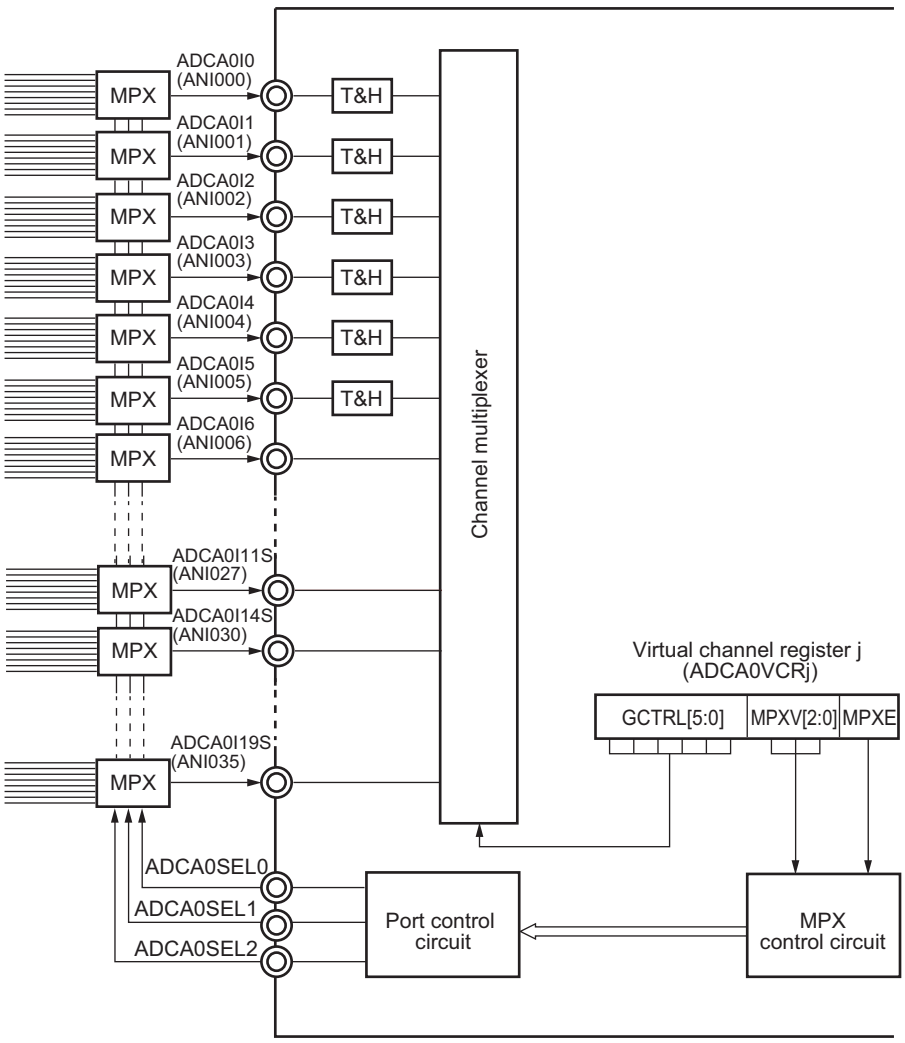
Note 1. For the condition setting to accept a trigger from external trigger input pin, see **Section 2.12, Noise Filter & Edge/Level Detector**.

Note 2. For the hardware trigger sources, see **Table 31.47, List of A/D Conversion Hardware Triggers**.

Figure 31.3 Internal Connection Diagram of External Trigger Input Pins

(4) Configuration of external analog multiplexer (MPX)

The external analog multiplexer (MPX) can be connected to any input signal pins ADCA0I0 to ADCA0I19S. An example of the external analog multiplexer connection is shown below.

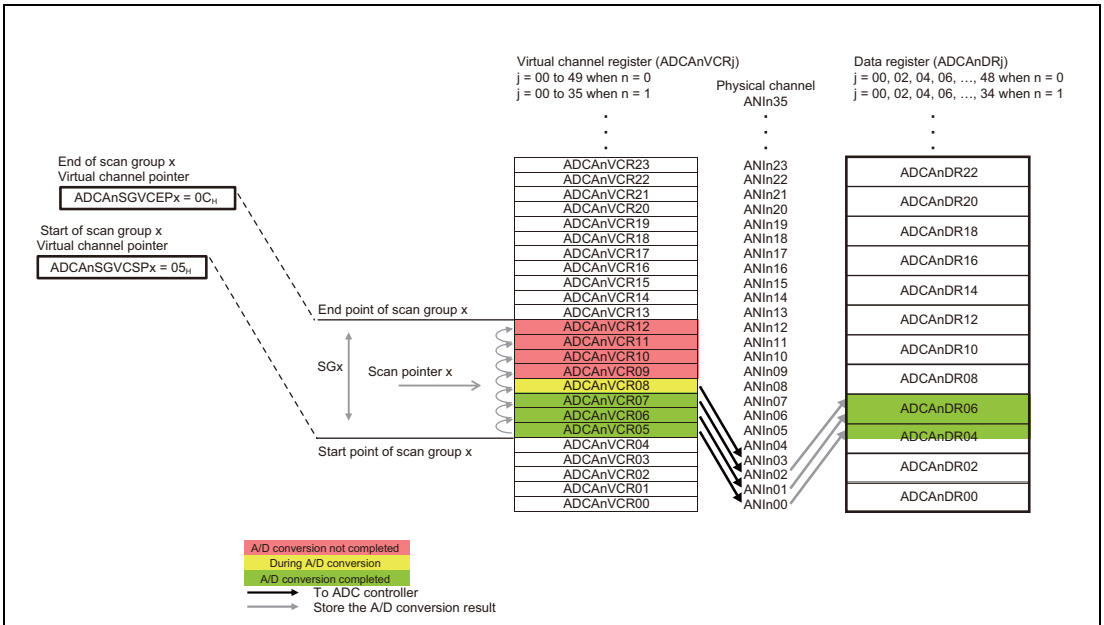


Note: The T&H function is disabled when the external analog multiplexer (MPX) is used.

Figure 31.4 Example of External Analog Multiplexer Connection

(5) Virtual channel

The virtual channel specifies the physical address to be scanned.
The virtual channel is controlled by the ADCAnVCRj register.
A usage example of the virtual channel is shown below.



Example) When scan group x starts scanning 5th to 12th virtual channels

1. The ADCAnVCRj register specifies the physical channel (ANInm) for which virtual channel j executes the A/D conversion. The A/D conversion pins can be set in any order. (The figure above shows an example when ANIn0 to ANIn7 are selected.)
2. The scan pointer x sequentially reads the physical channels specified in the ADCAnVCRj register from “start point of scan group x” to “end point of scan group x”, and starts the A/D conversion for the read physical address.
3. The scanned result of virtual channel j is stored in data register j.

Figure 31.5 Usage Example of Virtual Register

31.3 Registers

31.3.1 List of Registers

ADCA registers are listed in the following table.

For details about <ADCAn_base>, see **Section 31.1.2, Register Base Address**.

Table 31.12 List of Registers (1/2)

Module Name	Register Name	Symbol	Address
ADCA Specific Registers (Virtual Channel)			
ADCAn	Virtual Channel Register j	ADCAnVCRj	<ADCAn_base> + j × 4_H
ADCAn	PWM-Diag Virtual Channel Register	ADCAnPWDVCR	<ADCAn_base> + $0F4_H$
ADCAn	Data Register j	ADCAnDRj	<ADCAn_base> + 100_H + j × 2_H
ADCAn	Data Supplementary Information Register j	ADCAnDIRj	<ADCAn_base> + 200_H + j × 4_H
ADCAn	PWM-Diag data register	ADCAnPWDTSNDR	<ADCAn_base> + 178_H
ADCAn	PWM-Diag Data Supplementary Information Register	ADCAnPWDDIR	<ADCAn_base> + $2F4_H$
ADCA Specific Registers (Control)			
ADCAn	A/D Force Halt Register	ADCAnADHALTR	<ADCAn_base> + 300_H
ADCAn	A/D Control Register	ADCAnADCR	<ADCAn_base> + 304_H
ADCAn	MPX Current Register	ADCAnMPXCURR	<ADCAn_base> + $30C_H$
ADCAn	T&H Sampling Start Control Register	ADCAnTHSMPSTCR	<ADCAn_base> + 314_H
ADCAn	T&H Control Register	ADCAnTHCR	<ADCAn_base> + 318_H
ADCAn	T&H Group A Hold Start Control Register	ADCAnTHAHLDSTCR	<ADCAn_base> + $31C_H$
ADCAn	T&H Group B Hold Start Control Register	ADCAnTHBHLDSTCR	<ADCAn_base> + 320_H
ADCAn	T&H Group A Control Register	ADCAnTHACR	<ADCAn_base> + 324_H
ADCAn	T&H Group B Control Register	ADCAnTHBCR	<ADCAn_base> + 328_H
ADCAn	T&H Enable Register	ADCAnTHER	<ADCAn_base> + $32C_H$
ADCAn	T&H Group Select Register	ADCAnTHGSR	<ADCAn_base> + 330_H
ADCAn	Sampling Control Register	ADCAnSMPCR	<ADCAn_base> + 380_H
ADCA Specific Registers (Safety-related)			
ADCAn	Safety Control Register	ADCAnSFTCR	<ADCAn_base> + 334_H
ADCAn	Upper Limit/Lower Limit Table Register 0	ADCAnULLMTBR0	<ADCAn_base> + 338_H
ADCAn	Upper Limit/Lower Limit Table Register 1	ADCAnULLMTBR1	<ADCAn_base> + $33C_H$
ADCAn	Upper Limit/Lower Limit Table Register 2	ADCAnULLMTBR2	<ADCAn_base> + 340_H
ADCAn	Error Clear Register	ADCAnECR	<ADCAn_base> + 344_H
ADCAn	Upper Limit/Lower Limit Error Register	ADCAnULER	<ADCAn_base> + 348_H
ADCAn	Overwrite Error Register	ADCAnOWER	<ADCAn_base> + $34C_H$
Scan Group Specific Registers			
ADCAn	Scan Group x Start Control Register	ADCAnSGSTCRx	<ADCAn_base> + x × 40_H + 400_H
ADCAn	PWM-Diag Scan Group Control Register	ADCAnPWDSGCR	<ADCAn_base> + 508_H
ADCAn	Scan Group x Control Register	ADCAnSGCRx	<ADCAn_base> + x × 40_H + 408_H
ADCAn	Scan Group x Start Virtual Channel Pointer	ADCAnSGVCSPx	<ADCAn_base> + x × 40_H + $40C_H$
ADCAn	Scan Group x End Virtual Channel Pointer	ADCAnSGVCEPx	<ADCAn_base> + x × 40_H + 410_H
ADCAn	Scan Group x Multicycle Register	ADCAnSGMCYCRx	<ADCAn_base> + x × 40_H + 414_H
ADCAn	PWM-Diag Scan End Flag Clear Register	ADCAnPWDSGSEFCR	<ADCAn_base> + 518_H
ADCAn	Scan Group x Scan End Flag Clear Register	ADCAnSGSEFCRx	<ADCAn_base> + x × 40_H + 418_H
ADCAn	Scan Group Status Register	ADCAnSGSTR	<ADCAn_base> + 308_H
H/W Trigger Specific Register			
ADCAn	Scan Group x Start Trigger Control Register	ADCAnSGTSELx	<ADCAn_base> + x × 40_H + $41C_H$

Table 31.12 List of Registers (2/2)

Module Name	Register Name	Symbol	Address
Self-Diagnosis Specific Registers			
ADCAn	Self-Diagnostic Control Register 0	ADCAnDGCTL0	<ADCAn_base> + 350_H
ADCAn	Self-Diagnostic Control Register 1	ADCAnDGCTL1	<ADCAn_base> + 354_H
ADCAn	Pull Down Control Register 1	ADCAnPDCTL1	<ADCAn_base> + 358_H
ADCAn	Pull Down Control Register 2	ADCAnPDCTL2	<ADCAn_base> + $35C_H$
Emulation Specific Register			
ADCAn	Emulation Control Register	ADCAnEMU	<ADCAn_base> + 388_H

31.3.2 ADCA Specific Registers

This section describes the registers that are equipped in each of ADCA0 and ADCA1.

31.3.2.1 ADCAnVCRj — Virtual Channel Register j

This register is used to control the virtual channel.

Access: ADCAnVCRj can be read or written in 32-bit units.
ADCAnVCRjL can be read or written in 16-bit units.
ADCAnVCRjLL and ADCAnVCRjLH can be read or written in 8-bit units.

Address: ADCAnVCRj: <ADCAn_base> + j × 4_H
ADCAnVCRjL: <ADCAn_base> + j × 4_H
ADCAnVCRjLL: <ADCAn_base> + j × 4_H
ADCAnVCRjLH: <ADCAn_base> + j × 4_H + 1_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MPXE*1	MPXV[2:0]*1			—	—	CNVCLS*2	ADIE	ULS[1:0]		GCTRL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.13 ADCAnVCRj Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	MPXE*1	MPX Enable 0: The use of MPX is prohibited. No wait is inserted before A/D conversion is performed. 1: The use of MPX is permitted. The MPXV[2:0] bits are output from ADCAnSEL0 to ADCAnSEL2 when the virtual channel starts, and a wait of one A/D-conversion time is inserted before A/D conversion is performed.
14 to 12	MPXV[2:0]*1	These bits are used to set the MPX value to be transferred to an external analog multiplexer.
11, 10	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
9	CNVCLS*2	A/D Conversion Type Select for Self-Diagnosis 0: A/D conversion of the hold value is performed during a self-diagnosis. 1: Normal A/D conversion is performed during a self-diagnosis. When normal A/D conversion is performed during a self-diagnosis and MPX is in use (MPXE is set), however, a wait of one A/D conversion time is inserted before A/D conversion is performed. On the other hand, MPX cannot be used when A/D conversion of the hold value is performed during a self-diagnosis.
8	ADIE	A/D Conversion End Interrupt Enable 0: A scan group x end interrupt (INT_SGx) is not generated when A/D conversion for virtual channel j ends in SGx. 1: A scan group x end interrupt (INT_SGx) is generated when A/D conversion for virtual channel j ends in SGx.
7, 6	ULS[1:0]	Upper Limit/Lower Limit Table Select 00: Upper limit and lower limit are not checked. 01: Upper limit and lower limit are checked for ADCAnULLMTBR0. 10: Upper limit and lower limit are checked for ADCAnULLMTBR1. 11: Upper limit and lower limit are checked for ADCAnULLMTBR2.

Table 31.13 ADCAnVCRj Register Contents (2/2)

Bit Position	Bit Name	Function
5 to 0	GCTRL[5:0]	Physical Channel Select 0_H to 23_H: Corresponding ANInm is selected. 24_H: Diagnosis channel for A/D converter is selected. Others: Setting prohibit **Note:** These bits shall set the physical channel which is supported. See **Table 31.5, Indices of Products**, **Table 31.10, ADCA0 External Input/Output Signals** and **Table 31.11, ADCA1 External Input/Output Signals**.

Note 1. These bits are only supported for ADCA0. For ADCA1, when writing, write the value after reset.

Note 2. This bit is only supported when j = 33 to 35. Otherwise, when writing, write the value after reset.

CAUTION

To prevent malfunction, ADCAnVCRj should be set when SGACT of applicable scan groups is 0 (before scan groups are started) and TRGMD of applicable scan groups is 0.

Table 31.14 Selection of Physical Channels (1/2)

GCTRL5	GCTRL4	GCTRL3	GCTRL2	GCTRL1	GCTRL0	Analog Input Pin to be Selected
0	0	0	0	0	0	ADCAnI0 (Physical channel ANIn00)
0	0	0	0	0	1	ADCAnI1 (Physical channel ANIn01)
0	0	0	0	1	0	ADCAnI2 (Physical channel ANIn02)
0	0	0	0	1	1	ADCAnI3 (Physical channel ANIn03)
0	0	0	1	0	0	ADCAnI4 (Physical channel ANIn04)
0	0	0	1	0	1	ADCAnI5 (Physical channel ANIn05)
0	0	0	1	1	0	ADCAnI6 (Physical channel ANIn06)
0	0	0	1	1	1	ADCAnI7 (Physical channel ANIn07)
0	0	1	0	0	0	ADCAnI8 (Physical channel ANIn08)
0	0	1	0	0	1	ADCAnI9 (Physical channel ANIn09)
0	0	1	0	1	0	ADCAnI10 (Physical channel ANIn10)
0	0	1	0	1	1	ADCAnI11 (Physical channel ANIn11)
0	0	1	1	0	0	ADCAnI12 (Physical channel ANIn12)
0	0	1	1	0	1	ADCAnI13 (Physical channel ANIn13)
0	0	1	1	1	0	ADCAnI14 (Physical channel ANIn14)
0	0	1	1	1	1	ADCAnI15 (Physical channel ANIn15)
0	1	0	0	0	0	ADCAnI0S (Physical channel ANIn16)
0	1	0	0	0	1	ADCAnI1S (Physical channel ANIn17)
0	1	0	0	1	0	ADCAnI2S (Physical channel ANIn18)
0	1	0	0	1	1	ADCAnI3S (Physical channel ANIn19)
0	1	0	1	0	0	ADCAnI4S (Physical channel ANIn20)
0	1	0	1	0	1	ADCAnI5S (Physical channel ANIn21)
0	1	0	1	1	0	ADCAnI6S (Physical channel ANIn22)
0	1	0	1	1	1	ADCAnI7S (Physical channel ANIn23)
0	1	1	0	0	0	ADCAnI8S (Physical channel ANIn24)
0	1	1	0	0	1	ADCAnI9S (Physical channel ANIn25)
0	1	1	0	1	0	ADCAnI10S (Physical channel ANIn26)
0	1	1	0	1	1	ADCAnI11S (Physical channel ANIn27)
0	1	1	1	0	0	ADCAnI12S (Physical channel ANIn28)*1
0	1	1	1	0	1	ADCAnI13S (Physical channel ANIn29)*1

Table 31.14 Selection of Physical Channels (2/2)

GCTRL5	GCTRL4	GCTRL3	GCTRL2	GCTRL1	GCTRL0	Analog Input Pin to be Selected
0	1	1	1	1	0	ADCAnI14S (Physical channel ANIn30)
0	1	1	1	1	1	ADCAnI15S (Physical channel ANIn31)
1	0	0	0	0	0	ADCAnI16S (Physical channel ANIn32)
1	0	0	0	0	1	ADCAnI17S (Physical channel ANIn33)
1	0	0	0	1	0	ADCAnI18S (Physical channel ANIn34)
1	0	0	0	1	1	ADCAnI19S (Physical channel ANIn35)
1	0	0	1	0	0	Diagnosis channel for A/D converter
Other than above						Setting prohibited

Note 1. This setting only applies to ADCA1. Setting is prohibited in ADCA0.

31.3.2.2 ADCAnPWDVCR — PWM-Diag Virtual Channel Register

This register is used to indicate virtual channel setting (PWSAnPVCRx_y register setting) of the PWM-Diag (SG4).

Access: ADCAnPWDVCR is a read-only register that can be read in 32-bit units.
ADCAnPWDVCRL is a read-only register that can be read in 16-bit units.
ADCAnPWDVCRLL is a read-only register that can be read in 8-bit units.
ADCAnPWDVCRLH is a read-only register that can be read in 8-bit units.

Address: ADCAnPWDVCR: <ADCAn_base> + $0F4_H$
ADCAnPWDVCRL: <ADCAn_base> + $0F4_H$
ADCAnPWDVCRLL: <ADCAn_base> + $0F4_H$
ADCAnPWDVCRLH: <ADCAn_base> + $0F4_H$ + 1_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MPXE *1	MPXV[2:0]*1			—	—	—	—	ULS[1:0]		GCTRL[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.15 ADCAnPWDVCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	MPXE*1	The following setting is made by setting the PWSAnPVCRx_y.PWSAnVRDTy[27] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[11] (even channel) bit. MPX Enable Set this bit to 1 when an external analog multiplexer is used. 0: The use of MPX is prohibited. 1: The use of MPX is permitted. The MPXV[2:0] bits are output from ADCAnSEL0 to ADCAnSEL2 when the virtual channel starts, and a wait of one A/D-conversion time is inserted before A/D conversion is performed.
14 to 12	MPXV[2:0]*1	These bits are used to set the MPX value to be transferred to an external analog multiplexer by using the PWSAnPVCRx_y.PWSAnVRDTy[26:24] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[10:8] (even channel) bit.
11 to 8	Reserved	When read, the value after reset is returned.

Table 31.15 ADCAnPWDVCR Register Contents

Bit Position	Bit Name	Function
7 to 6	ULS[1:0]	The following setting is made by setting the PWSAnPVCRx_y.PWSAnVRDTy[23:22] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[7:6] (even channel) bit. Upper Limit/Lower Limit Table Select 00: Upper limit and lower limit are not checked. 01: Upper limit and lower limit are checked for ADCAnULLMTBR0. 10: Upper limit and lower limit are checked for ADCAnULLMTBR1. 11: Upper limit and lower limit are checked for ADCAnULLMTBR2.
5 to 0	GCTRL[5:0]	The following setting is made by setting the PWSAnPVCRx_y.PWSAnVRDTy[21:16] (odd channel) or PWSAnPVCRx_y.PWSAnVRDTx[5:0] (even channel) bit. Physical Channel Select These bits are used to specify a physical channel to be assigned to virtual channel j. For the selection of the channel, see **Table 31.14, Selection of Physical Channels**.

Note 1. These bits are only supported for ADCA0.
For ADCA1, when read, the value after reset is returned.

31.3.2.3 ADCAnDRj — Data Register j

This register is a 32/16-bit read-only register that stores the A/D conversion results corresponding to ADCAnVCRj and ADCAnVCR(j+1). As the A/D conversion results, the conversion result for ADCAnVCR(j+1) is stored in the upper bits (ADCAnDR(j+1)), and the conversion result for ADCAnVCRj is stored in the lower bits (ADCAnDRj).

Access: ADCAnDRj is a read-only register that can be read in 32-bit units.
ADCAnDRjL and ADCAnDRjH are the read-only registers that can be read in 16-bit units.

Address: ADCAnDRj: <ADCAn_base> + 100_H + j × 2_H
ADCAnDRjL: <ADCAn_base> + 100_H + j × 2_H
ADCAnDRjH: <ADCAn_base> + 100_H + j × 2_H + 2_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DR(j+1)[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRj[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.16 ADCAnDRj Register Contents

Bit Position	Bit Name	Function
31 to 16	DR(j+1)[15:0]	These bits are used to store the A/D conversion result data. (The A/D conversion result for the channel set in ADCAnVCR(j+1) is transferred.)
15 to 0	DRj[15:0]	These bits are used to store the A/D conversion result data. (The A/D conversion result for the channel set in ADCAnVCRj is transferred.)

CAUTION

If the number of channels is odd, the higher-order bits (DR(j+1)[15:0]) in the ADCAnDRj register cannot be used. If virtual channels 33, 34, and 35 are used exclusively for self-diagnosis, the lower-order bits (DRj[15:0]) for channel 32 cannot be used.

NOTES

1. j = 00, 02, …, 46, 48 (for ADCA0)
   j = 00, 02, …, 32, 34 (for ADCA1)
2. By controlling ADCAnADCR.CRAC and ADCAnADCR.CTYP, the data format of this register becomes as follows:
   - ADCAnADCR.CTYP = 0 and ADCAnADCR.CRAC = 0 → Right alignment is used.
     →The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 27 to 16, and the A/D conversion result for ADCAnVCRj is transferred to bits 11 to 0.
   - ADCAnADCR.CTYP = 0 and ADCAnADCR.CRAC = 1 → Left alignment is used.
     →The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 31 to 20, and the A/D conversion result for ADCAnVCRj is transferred to bits 15 to 4.
   - ADCAnADCR.CTYP = 1 and ADCAnADCR.CRAC = 0 → Right alignment is used.
     →The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 25 to 16, and the A/D conversion result for ADCAnVCRj is transferred to bits 9 to 0.
   - ADCAnADCR.CTYP = 1 and ADCAnADCR.CRAC = 1 → Left alignment is used.
     →The A/D conversion result for ADCAnVCR(j+1) is transferred to bits 31 to 22, and the A/D conversion result for ADCAnVCRj is transferred to bits 15 to 6.

31.3.2.4 ADCAnDIRj — Data Supplementary Information Register j

This register is a 32-bit read-only register that stores the A/D conversion result for ADCAnDRj and information incidental to the A/D converted value.

As the A/D conversion result, the ADCAnDRj value is transferred. As information incidental to the A/D converted value, information about the write flag (WFLG), the MPX value (MPXV[2:0]), and the physical channel (ID[5:0]) is transferred. The data format of the A/D conversion result stored in ADCAnDIRj is the same as that for the ADCAnDRj register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ADCAn_base> + 200_H + j × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MPXE *1	MPXV[2:0]*1			—	—	WFLG	—	—	—	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.17 ADCAnDIRj Register Contents

Bit Position	Bit Name	Function
31	MPXE*1	MPX Enable Flag 0: MPX function is not used. 1: MPX function is used.
30 to 28	MPXV[2:0]*1	These bits are used to store the MPX value. The MPX value to be stored is the MPX value of the most recent conversion result.
27, 26	Reserved	When read, the value after reset is returned.
25	WFLG	Write Flag 0: ADCAnDRj or ADCAnDIRj is read (cleared when read). 1: A/D converted value is stored in ADCAnDRj (set when the value is stored).
24 to 22	Reserved	When read, the value after reset is returned.
21 to 16	ID[5:0]	These bits store the physical channel number (GCTRL) corresponding to the conversion result. The physical channel number to be stored is the number corresponding to the most recent conversion result.
15 to 0	DR[15:0]	These bits are used to store the A/D conversion result.

Note 1. These bits are only supported by ADCA0.
For ADCA1, when read, the value after reset is returned.

31.3.2.5 ADCAnPWDTSNDR — PWM-Diag Data Register

This register is a 32/16-bit read-only register that stores the A/D conversion results corresponding to the PWM-Diag. As the A/D conversion results, the conversion result for the PWM-Diag (PWDDR) is stored in the upper bits.

Access: ADCAnPWDTSNDR is a read-only register that can be read in 32-bit units.
ADCAnPWDTSNDRH is a read-only register that can be read in 16-bit units.

Address: ADCAnPWDTSNDR: <ADCAn_base> + 178_H
ADCAnPWDTSNDRH: <ADCAn_base> + $17A_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWDDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.18 ADCAnPWDTSNDR Register Contents

Bit Position	Bit Name	Function
31 to 16	PWDDR[15:0]	These bits are used to store the A/D conversion result data for the PWM-Diag.
15 to 0	Reserved	When read, the value after reset is returned.

NOTE

The data format of this register is controlled by ADCAnADCR.CRAC and ADCAnADCR.CTYP, as shown below.

- ADCAnADCR.CTYP = 0, ADCAnADCR.CRAC = 0 → Right alignment is used.
  → The A/D conversion result for ADCAnPWDVCR is transferred to bits 27 to 16.
- ADCAnADCR.CTYP = 0, ADCAnADCR.CRAC = 1 → Left alignment is used.
  → The A/D conversion result for ADCAnPWDVCR is transferred to bits 31 to 20.
- ADCAnADCR.CTYP = 1, ADCAnADCR.CRAC = 0 → Right alignment is used.
  → The A/D conversion result for ADCAnPWDVCR is transferred to bits 25 to 16.
- ADCAnADCR.CTYP = 1, ADCAnADCR.CRAC = 1 → Left alignment is used.
  → The A/D conversion result for ADCAnPWDVCR is transferred to bits 31 to 22.

31.3.2.6 ADCAnPWDDIR — PWM-Diag Data Supplementary Information Register

This register is a 32-bit read-only register that stores the A/D conversion result when PWM-Diag is used, and information incidental to the A/D converted value.

As the A/D conversion result, the ADCAnPWDSNDR.PWDDR[15:0] value is transferred. As supplementary information to the A/D converted value, the write flag (WFLG), MPX value (MPXV[2:0]), and physical channel (ID[5:0]) are transferred. The data format of the A/D conversion result stored in ADCAnPWDDIR is the same as that for the ADCAnPWDTSNDR register.

Access: This register is a read-only register that can be read in 32-bit units.

Address: <ADCAn_base> + $2F4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MPXE*1	MPXV[2:0]*1			—	—	WFLG	—	—	—	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWDDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.19 ADCAnPWDDIR Register Contents

Bit Position	Bit Name	Function
31	MPXE*1	MPX Enable Flag 0: The MPX function is not used. 1: The MPX function is used.
30 to 28	MPXV[2:0]*1	These bits are used to store the MPX value. The MPX value to be stored is the MPX value of the most recent conversion result.
27, 26	Reserved	When read, the value after reset is returned.
25	WFLG	Write Flag 0: ADCAnPWDTSNDR or ADCAnPWDDIR is read (cleared when read). 1: The A/D converted value is stored in ADCAnPWDTSNDR (set when the value is stored).
24 to 22	Reserved	When read, the value after reset is returned.
21 to 16	ID[5:0]	These bits are used to store the physical channel number (GCTRL) corresponding to the conversion result. The physical channel number to be stored is the physical channel number corresponding to the most recent conversion result.
15 to 0	PWDDR[15:0]	These bits are used to store the A/D conversion result for PWM-Diag.

Note 1. These bits are only supported for ADCA0.
For ADCA1, when read, the value after reset is returned.

31.3.2.7 ADCAnADHALTR — A/D Force Halt Register

This register is used to halt conversion for all SGs of ADCAn. The bits are always read as 0.

Access: ADCAnADHALTR is a write-only register that can be written in 32-bit units.
ADCAnADHALTRL is a write-only register that can be written in 16-bit units.
ADCAnADHALTRLL is a write-only register that can be written in 8-bit units.

Address: ADCAnADHALTR:<ADCAn_base> + 300_H
ADCAnADHALTRL:<ADCAn_base> + 300_H
ADCAnADHALTRLL:<ADCAn_base> + 300_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HALT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 31.20 ADCAnADHALTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	HALT	ADCA Force Halt Trigger All scan groups are halted and initialized, and ADCA becomes idle state. Writing of 0: No effect Writing of 1: Scan groups are halted.

31.3.2.8 ADCAnADCR — A/D Control Register

This register is used for ADCAn common control.

Access: ADCAnADCR can be read or written in 32-bit units.
ADCAnADCRL can be read or written in 16-bit units.
ADCAnADCRLL can be read or written in 8-bit units.

Address: ADCAnADCR: <ADCAn_base> + 304_H
ADCAnADCRL: <ADCAn_base> + 304_H
ADCAnADCRLL: <ADCAn_base> + 304_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DGON	—	CRAC	CTYP	—	—	SUSMTD[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R	R/W	R/W

Table 31.21 ADCAnADCR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7	DGON	Self-Diagnostic Voltage Standby Control 0: The self-diagnostic voltage circuit is turned off. 1: The self-diagnostic voltage circuit is turned on, or the reference voltage is updated.
6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	CRAC	Alignment Control 0: The results of conversion to PWDDR and ADCAnDRj are stored right-aligned. 1: The results of conversion to PWDDR and ADCAnDRj are stored left-aligned.
4	CTYP	12/10 Bit Select Mode 0: 12-bit mode 1: 10-bit mode
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SUSMTD [1:0]	Suspend Mode Select These bits are used to select the suspend method when a higher-priority scan group interrupts a lower-priority scan group. 00: Synchronous suspend when a higher-priority SG or SVSTOP interrupts. 01: Asynchronous suspend when a higher-priority SG (SG2, SG3, SG4) and SVSTOP interrupt SG1, and synchronous suspend when a higher-priority SG (SG3, SG4) and SVSTOP interrupt SG2, or when a higher-priority SG (SG4) and SVSTOP interrupt SG3. 10: Asynchronous suspend when a higher-priority SG or SVSTOP interrupts. 11: Setting prohibited

CAUTION

To prevent malfunction, ADCAnADCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

- Synchronous suspend:
  If a request from a higher-priority SG occurs while a lower-priority SG is being processed, the A/D conversion for the higher-priority SG is performed after the on-going A/D conversion of a channel is completed. After processing for the higher-priority SG is completed, the suspended channel processing for the lower-priority SG is resumed.
- Asynchronous suspend:
  If a request from a higher-priority SG occurs while a lower-priority SG is being processed, the on-going channel processing is suspended, and then the A/D conversion for the higher-priority SG is performed. After processing for the higher-priority SG is completed, the suspended A/D channel conversion for the lower-priority SG is resumed.

For details, see **Figure 31.21, Example of Synchronous Suspend and Resume Operation** and **Figure 31.22, Example of Asynchronous Suspend and Resume Operation**.

31.3.2.9 ADCAnMPXCURR — MPX Current Register

This register is used to store the MPX value for an external analog multiplexer.

Access: ADCAnMPXCURR is a read-only register that can be read in 32-bit units.
ADCAnMPXCURRL is a read-only register that can be read in 16-bit units.
ADCAnMPXCURRLL is a read-only register that can be read in 8-bit units.

Address: ADCAnMPXCURR: <ADCAn_base> + $30C_H$
ADCAnMPXCURRL: <ADCAn_base> + $30C_H$
ADCAnMPXCURRLL: <ADCAn_base> + $30C_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MPXCUR[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.22 ADCAnMPXCURR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2 to 0	MPXCUR[2:0]	These bits are used to store the current MPX value. If conversion of a virtual channel starts after setting ADCAnVCRj.MPXE to 1, the setting of ADCAnVCRj.MPXV[2:0] is stored. If conversion of a virtual channel starts after setting ADCAnPWDVCR.MPXE to 1, the setting of ADCAnPWDVCR.MPXV[2:0] is stored.

NOTE

In RH850/F1M, only ADCA0 supports this function.

31.3.2.10 ADCAnTHSMPSTCR — T&H Sampling Start Control Register

This register is used to control the start of sampling for all T&Hk (k = 0 to 5). The bits are always read as 0.

Access: ADCAnTHSMPSTCR is a write-only register that can be written in 32-bit units.
ADCAnTHSMPSTCRL is a write-only register that can be written in 16-bit units.
ADCAnTHSMPSTCRLL is a write-only register that can be written in 8-bit units.

Address: ADCAnTHSMPSTCR: <ADCAn_base> + 314_H
ADCAnTHSMPSTCRL: <ADCAn_base> + 314_H
ADCAnTHSMPSTCRLL: <ADCAn_base> + 314_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SMPST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 31.23 ADCAnTHSMPSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SMPST	T&H Sampling Start Control Trigger 0: No effect 1: Sampling for all T&H is started. The conditions to place the T&H circuit in the sampling state are as follows: • Condition to start sampling while T&H is stopped: 1 being written to ADCAnTHSMPSTCR.SMPST while ADCAnTHER.THkE = 1 (k = 0 to 5). • Condition to start continuous sampling in automatic sampling: A/D conversion of the hold value for T&Hk being completed while ADCAnTHER.THkE = 1 (k = 0 to 5) and ADCAnTHCR.ASMPMSK = 1.

NOTE

In RH850/F1M, only ADCA0 supports this function.

31.3.2.11 ADCAnTHCR — T&H Control Register

This register controls the sampling transition after A/D conversion of the hold value for T&H is completed.

Automatic start of sampling on the T&H circuit after A/D conversion of the hold value for T&H is completed shortens the time required for the generation of succeeding hold completion triggers.

Access: ADCAnTHCR can be read or written in 32-bit units.
ADCAnTHCRL can be read or written in 16-bit units.
ADCAnTHCRLL can be read or written in 8-bit units.

Address: ADCAnTHCR: <ADCAn_base> + 318_H
ADCAnTHCRL: <ADCAn_base> + 318_H
ADCAnTHCRLL: <ADCAn_base> + 318_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ASMPMSK
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 31.24 ADCAnTHCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ASMPMSK	Automatic Sampling Mask Control 0: Automatic sampling is not performed. 1: Automatic sampling is performed.

CAUTION

To prevent malfunction, ADCAnTHCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

In RH850/F1M, only ADCA0 supports this function.

31.3.2.12 ADCAnTHAHLDSTCR — T&H Group A Hold Start Control Register

This register is used to control the start of the hold for T&H group A. The bits are always read as 0.

Access: ADCAnTHAHLDSTCR is a write-only register that can be written in 32-bit units.
ADCAnTHAHLDSTCRL is a write-only register that can be written in 16-bit units.
ADCAnTHAHLDSTCRLL is a write-only register that can be written in 8-bit units.

Address: ADCAnTHAHLDSTCR: <ADCAn_base> + $31C_H$
ADCAnTHAHLDSTCRL: <ADCAn_base> + $31C_H$
ADCAnTHAHLDSTCRLL: <ADCAn_base> + $31C_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HLDST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 31.25 ADCAnTHAHLDSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	HLDST	T&H Group A Hold Start Control Trigger 0: No effect 1: Hold for T&H group A is started. The condition to place T&H group A in the hold state is as follows: • 1 being written to ADCAnTHAHLDSTCR.HLDST while ADCAnTHER.THkE = 1 (k = 0 to 5) and ADCAnTHGSR.THkGS = 0 (k = 0 to 5).

NOTE

In RH850/F1M, only ADCA0 supports this function.

31.3.2.13 ADCAnTHBHLDSTCR — T&H Group B Hold Start Control Register

This register is used to control the start of the hold for T&H group B. The bits are always read as 0.

Access: ADCAnTHBHLDSTCR is a write-only register that can be written in 32-bit units.
ADCAnTHBHLDSTCRL is a write-only register that can be written in 16-bit units.
ADCAnTHBHLDSTCRLL is a write-only register that can be written in 8-bit units.

Address: ADCAnTHBHLDSTCR: <ADCAn_base> + 320_H
ADCAnTHBHLDSTCRL: <ADCAn_base> + 320_H
ADCAnTHBHLDSTCRLL: <ADCAn_base> + 320_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HLDST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 31.26 ADCAnTHBHLDSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	HLDST	T&H Group B Hold Start Control Trigger 0: No effect 1: Hold for T&H group B is started. The condition to place T&H group B in the hold state is as follows: • 1 being written to ADCAnTHBHLDSTCR.HLDST while ADCAnTHER.THkE = 1 (k = 0 to 5) and ADCAnTHGSR.THkGS = 1 (k = 0 to 5).

NOTE

In RH850/F1M, only ADCA0 supports this function.

31.3.2.14 ADCAnTHACR — T&H Group A Control Register

This register is used to control T&H group A.

Access: ADCAnTHACR can be read or written in 32-bit units.
ADCAnTHACRL can be read or written in 16-bit units.
ADCAnTHACRLL can be read or written in 8-bit units.

Address: ADCAnTHACR: <ADCAn_base> + 324_H
ADCAnTHACRL: <ADCAn_base> + 324_H
ADCAnTHACRLL: <ADCAn_base> + 324_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	HLDCTE	HLDTE	—	—	SGS[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 31.27 ADCAnTHACR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	HLDCTE	T&H Group A Hold Completion Trigger Enable This bit is used when self-diagnosis of the T&H circuit is to proceed. 0: Self-diagnosis does not proceed. 1: Self-diagnosis proceeds. Note: The SGx_TRG (x = 1 to 3) trigger is selected for the trigger input of the scan group that is not selected in SGS[1:0] of ADCAnTHACR and SGS[1:0] of ADCAnTHBCR.
4	HLDTE	T&H Group A Hold Trigger Enable 0: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is selected for the hold start trigger of T&H group A. 1: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is not selected for the hold start trigger of T&H group A. Note: ADCAnTHAHLDSTCR.HLDST becomes a hold start trigger regardless of the ADCAnTHACR.HLDTE setting. Set this bit to 0 when self-diagnosis of the T&H circuit is to proceed.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	SGS[1:0]	T&H Group A Scan Group Select 00: No scan group is selected for T&H group A. 01: SG1 is selected for T&H group A. 10: SG2 is selected for T&H group A. 11: SG3 is selected for T&H group A. Note: 1. If ADCAnTHACR.SGS[1:0] is set to 0_H, T&H does not operate. When you enable T&Hk in ADCAnTHER.THkE, make sure that a scan group is specified in SGS[1:0]. 2. Selecting the same scan group as T&H group B is prohibited.

CAUTION

To prevent malfunction, ADCAnTHACR should be set when SGACT of all scan groups is 0 (before scan

groups are started) and TRGMD of all scan groups is 0.

NOTE

In RH850/F1M, only ADCA0 supports this function.

31.3.2.15 ADCAnTHBCR — T&H Group B Control Register

This register is used to control T&H group B.

Access: ADCAnTHBCR can be read or written in 32-bit units.
ADCAnTHBCRL can be read or written in 16-bit units.
ADCAnTHBCRLL can be read or written in 8-bit units.

Address: ADCAnTHBCR: <ADCAn_base> + 328_H
ADCAnTHBCRL: <ADCAn_base> + 328_H
ADCAnTHBCRLL: <ADCAn_base> + 328_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	HLDCTE	HLDTE	—	—	SGS[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 31.28 ADCAnTHBCR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	HLDCTE	T&H Group B Hold Completion Trigger Enable This bit is used when self-diagnosis of the T&H circuit is to proceed. 0: Self-diagnosis does not proceed. 1: Self-diagnosis proceeds. Note: The SGx_TRG (x = 1 to 3) trigger is selected for the trigger input of the scan group that is not selected in SGS[1:0] of ADCAnTHACR and SGS[1:0] of ADCAnTHBCR.
4	HLDTE	T&H Group B Hold Trigger Enable 0: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is selected for the hold start trigger of T&H group B. 1: The SGx (x = 1 to 3) trigger selected in SGS[1:0] is not selected for the hold start trigger of T&H group B. Note: ADCAnTHBHLDSTCR.HLDST becomes a hold start trigger regardless of the ADCAnTHBCR.HLDTE setting. Set this bit to 0 when self-diagnosis of the T&H circuit is to proceed.
3, 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Table 31.28 ADCAnTHBCR Register Contents

Bit Position	Bit Name	Function
1 to 0	SGS[1:0]	T&H Group B Scan Group Select 00: No scan group is selected for T&H group B. 01: SG1 is selected for T&H group B. 10: SG2 is selected for T&H group B. 11: SG3 is selected for T&H group B. Note: 1. If ADCAnTHBCR.SGS[1:0] is set to 0_H, T&H does not operate. When you enable T&Hk in ADCAnTHER.THkE, make sure that a scan group is specified in SGS[1:0]. 2. Selecting the same scan group as T&H group A is prohibited.

CAUTION

To prevent malfunction, ADCAnTHBCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

In RH850/F1M, only ADCA0 supports this function.

31.3.2.16 ADCAnTHER — T&H Enable Register

This register controls enabling and disabling of each T&H.

Access: ADCAnTHER can be read or written in 32-bit units.
ADCAnTHERL can be read or written in 16-bit units.
ADCAnTHERLL can be read or written in 8-bit units.

Address: ADCAnTHER: <ADCAn_base> + $32C_H$
ADCAnTHERL: <ADCAn_base> + $32C_H$
ADCAnTHERLL: <ADCAn_base> + $32C_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	TH5E	TH4E	TH3E	TH2E	TH1E	TH0E
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.29 ADCAnTHER Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	TH5E	T&H5 Enable 0: T&H5 is disabled. 1: T&H5 is enabled. Note: If TH5E is set to 0, T&H5 is always stopped.
4	TH4E	T&H4 Enable 0: T&H4 is disabled. 1: T&H4 is enabled. Note: If TH4E is set to 0, T&H4 is always stopped.
3	TH3E	T&H3 Enable 0: T&H3 is disabled. 1: T&H3 is enabled. Note: If TH3E is set to 0, T&H3 is always stopped.
2	TH2E	T&H2 Enable 0: T&H2 is disabled. 1: T&H2 is enabled Note: If TH2E is set to 0, T&H2 is always stopped.
1	TH1E	T&H1 Enable 0: T&H1 is disabled. 1: T&H1 is enabled. Note: If TH1E is set to 0, T&H1 is always stopped.
0	TH0E	T&H0 Enable 0: T&H0 is disabled. 1: T&H0 is enabled Note: If TH0E is set to 0, T&H0 is always stopped.

CAUTION

To prevent malfunction, ADCAnTHER should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

In RH850/F1M, only ADCA0 supports this function.

31.3.2.17 ADCAnTHGSR — T&H Group Select Register

This register is used to select a T&H group for each T&H.

Access: ADCAnTHGSR can be read or written in 32-bit units.
ADCAnTHGSRL can be read or written in 16-bit units.
ADCAnTHGSRLL can be read or written in 8-bit units.

Address: ADCAnTHGSR: <ADCAn_base> + 330_H
ADCAnTHGSRL: <ADCAn_base> + 330_H
ADCAnTHGSRLL: <ADCAn_base> + 330_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	TH5GS	TH4GS	TH3GS	TH2GS	TH1GS	TH0GS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.30 ADCAnTHGSR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	TH5GS	T&H5 Group Select 0: T&H5 is selected to group A. 1: T&H5 is selected to group B.
4	TH4GS	T&H4 Group Select 0: T&H4 is selected to group A. 1: T&H4 is selected to group B.
3	TH3GS	T&H3 Group Select 0: T&H3 is selected to group A. 1: T&H3 is selected to group B.
2	TH2GS	T&H2 Group Select 0: T&H2 is selected to group A. 1: T&H2 is selected to group B.
1	TH1GS	T&H1 Group Select 0: T&H1 is selected to group A. 1: T&H1 is selected to group B.
0	TH0GS	T&H0 Group Select 0: T&H0 is selected to group A. 1: T&H0 is selected to group B.

CAUTION

- Do not set T&H0 to T&H2 to the same group as T&H3 to T&H5.
  Example
  - Group A: 0ch, 1ch, 2ch
    Group B: 3ch, 4ch, 5ch → Setting allowed
  - Group A: 0ch
    Group B: 1ch, 2ch → Setting allowed
  - Group A: 0ch, 1ch, 3ch
    Group B: 2ch, 4ch → Setting prohibited
- To prevent malfunction, ADCAnTHGSR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

In RH850/F1M, only ADCA0 supports this function.

31.3.2.18 ADCAnSMPCR — Sampling Control Register

This register controls sampling.

ADCTLnSMPCR controls the sampling time for SG4 (PWM-Diag) and SG1 to SG3.

Access: ADCAnSMPCR can be read or written in 32-bit units.
ADCAnSMPCRL can be read or written in 16-bit units.
ADCAnSMPCRLL can be read or written in 8-bit units.

Address: ADCAnSMPCR: <ADCAn_base> + 380_H
ADCAnSMPCRL: <ADCAn_base> + 380_H
ADCAnSMPCRLL: <ADCAn_base> + 380_H

Value after reset: 0000 0018_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SMPT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.31 ADCAnSMPCR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
7 to 0	SMPT[7:0]	These bits are used to set the sampling time (the number of cycles). 12_H: 18 cycles (ADCLK = 8 MHz to 32 MHz) 18_H: 24 cycles (ADCLK = 8 MHz to 40 MHz) Settings other than above are prohibited.

CAUTION

- To prevent malfunction, ADCATLnSMPCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
- When SMPT is changed, the A/D conversion wait time is also changed when MPX is used by virtual channel register j (ADCAnVCRj) or PWM-Diag virtual channel register (ADCAnPWDVCR).

31.3.2.19 ADCAnSFTCR — Safety Control Register

This is a register for safety control.

Access: ADCAnSFTCR can be read or written in 32-bit units.
ADCAnSFTCRL can be read or written in 16-bit units.
ADCAnSFTCRLL can be read or written in 8-bit units.

Address: ADCAnSFTCR: <ADCAn_base> + 334_H
ADCAnSFTCRL: <ADCAn_base> + 334_H
ADCAnSFTCRLL: <ADCAn_base> + 334_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	RDCLRE	ULEIE	OWEIE	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 31.32 ADCAnSFTCR Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4	RDCLRE	Read & Clear Enable When the A/D conversion result is read, this bit selects whether the A/D conversion result is cleared by hardware. 0: ADCAnPWDTSNDR/ADCAnDRj and ADCAnPWDDIR/ADCAnDIRj are not cleared when ADCAnPWDTSNDR/ADCAnDRj or ADCAnPWDDIR/ADCAnDIRj is read. 1: ADCAnPWDTSNDR/ADCAnDRj and ADCAnPWDDIR/ADCAnDIRj are cleared when ADCAnPWDTSNDR/ADCAnDRj or ADCAnPWDDIR/ADCAnDIRj is read. WFLG of ADCAnDIRj is cleared regardless of the RDCLRE setting when ADCAnDRj or ADCAnDIRj is read.
3	ULEIE	A/D Error Interrupt (INT_ADE) Enable on Upper/Lower Limit Error Detection 0: Disabled 1: Enabled
2	OWEIE	A/D Error Interrupt (INT_ADE) Enable on Overwrite Error Detection 0: Disabled 1: Enabled
1, 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

To prevent malfunction, ADCAnSFTCR should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

31.3.2.20 ADCAnULLMTBR0 to 2 — Upper Limit/Lower Limit Table Registers 0 to 2

These registers are used to set the threshold for detection of an upper limit or lower limit error in the A/D converted value. Any of ADCAnULLMTBR0 to ADCAnULLMTBR2 is specified by setting ADCAnPWDVCR.ULS[1:0] and ADCAnVCRj.ULS[1:0] and compared with ADCAnPWDTSNDR and ADCAnDRj.

Access: ADCAnULLMTBR0 to 2 can be read or written in 32-bit units.
ADCAnULLMTBR0L to 2L and ADCAnULLMTBR0H to 2H can be read or written in 16-bit units.

Address: ADCAnULLMTBR0: <ADCAn_base> + 338_H
ADCAnULLMTBR1: <ADCAn_base> + $33C_H$
ADCAnULLMTBR2: <ADCAn_base> + 340_H

Address: ADCAnULLMTBR0L: <ADCAn_base> + 338_H
ADCAnULLMTBR1L: <ADCAn_base> + $33C_H$
ADCAnULLMTBR2L: <ADCAn_base> + 340_H

Address: ADCAnULLMTBR0H: <ADCAn_base> + 338_H + 2_H
ADCAnULLMTBR1H: <ADCAn_base> + $33C_H$+ 2_H
ADCAnULLMTBR2H: <ADCAn_base> + 340_H + 2_H

Value after reset: $FFF0\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ULMTB[11:0]												—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LLMTB[11:0]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 31.33 ADCAnULLMTBR0 to 2 Registers Contents

Bit Position	Bit Name	Function
31 to 20	ULMTB[11:0]	Upper Limit Table Specify the threshold for detection of an upper limit error in the A/D converted value. The upper limit error (ADCAnULER.UE) is set when the following condition is met: ULMTB[11:0] < A/D converted value
19 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 4	LLMTB[11:0]	Lower Limit Table Specify the threshold for detection of a lower limit error in the A/D converted value. The lower limit error (ADCAnULER.LE) is set when the following condition is met: LLMTB[11:0] > A/D converted value
3 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

CAUTION

- When A/D conversion is executed in 10-bit mode (ADCAnADCR.CTYP = 1), ULMTB[1:0] and LLMTB[1:0] should be set to 11_B and 00_B, respectively.
- To prevent malfunction, ADCAnULLMTBR0 to 2 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
- The upper-limit table (ULMTB[11:0]) must be greater than the lower-limit table (LLMTB[11:0]).

31.3.2.21 ADCAnECR — Error Clear Register

This register is used to control clearing of an error. The bits are always read as 0.

Access: ADCAnECR is a write-only register that can be written in 32-bit units.
ADCAnECRL is a write-only register that can be written in 16-bit units.
ADCAnECRLL is a write-only register that can be written in 8-bit units.

Address: ADCAnECR: <ADCAn_base> + 344_H
ADCAnECRL: <ADCAn_base> + 344_H
ADCAnECRLL: <ADCAn_base> + 344_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ULEC	OWEC	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	R	R

Table 31.34 ADCAnECR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing, write the value after reset.
3	ULEC	Upper Limit Error Flag (ADCAnULER.UE) / Lower Limit Error Flag (ADCAnULER.LE) Clear, Upper/Lower Limit Error Capture (ADCAnULER.ULECAP[5:0]), Scan Group Bit (ULSG[1:0]) When Upper/Lower Limit Error Occurs, MPX Usage Bit (MPXE), and the MPX Value Storing Bit (MPXV[2:0]) Clear When Upper/Lower Limit Error Occurs 0: No effect. 1: Clears the flag.
2	OWEC	Overwrite Error Flag (ADCAnOWER.OWE) and Overwrite Error Capture (ADCAnOWER.OWECAP[5:0]) Clear 0: No effect. 1: Clears the flag.
1, 0	Reserved	When writing, write the value after reset.

31.3.2.22 ADCAnULER — Upper Limit/Lower Limit Error Register

This register is a read-only register that indicates information regarding the upper limit/lower limit errors.

Access: ADCAnULER is a read-only register that can be read in 32-bit units.
ADCAnULERL is a read-only register that can be read in 16-bit units.

ADCAnULERLH is a read-only register that can be read in 8-bit units.
ADCAnULERLL is a read-only register that can be read in 8-bit units.

Address: ADCAnULER: <ADCAn_base> + 348_H
ADCAnULERL: <ADCAn_base> + 348_H

ADCAnULERLL: <ADCAn_base> + 348_H
ADCAnULERLH: <ADCAn_base> + 348_H + 1_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UE	LE	ULSG[1:0]		MPXE	MPXV[2:0]			—	—	ULECAP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.35 ADCAnULER Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	UE	Upper Limit Error Flag 0: An upper limit error is not detected. 1: An upper limit error is detected. Setting condition: The A/D converted value exceeds the upper limit threshold specified by the upper limit/lower limit table registers 0 to 2 (ADCAnULLMTBR0 to 2). If a subsequent upper limit error is detected in A/D conversion while this bit is set to 1, the ADCAnULER register is not updated. Clearing condition: 1 is written to ADCAnECR.ULEC.
14	LE	Lower Limit Error Flag 0: A lower limit error is not detected. 1: A lower limit error is detected. Setting condition: The A/D converted value is lower than the lower limit threshold specified by the upper limit/lower limit table registers 0 to 2 (ADCAnULLMTBR0 to 2). If a subsequent lower limit error is detected in A/D conversion while this bit is set to 1, the ADCAnULER register is not updated. Clearing condition: 1 is written to ADCAnECR.ULEC.
13, 12	ULSG[1:0]	Scan Group where an Upper Limit/Lower Limit Error Occurs 00: No upper limit/lower limit error occurred. 01: The scan group where an upper limit/lower limit error occurred is SG1 to SG3. 10: The scan group where an upper limit/lower limit error occurred is PWM-Diag. Capture condition: When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0. Clearing condition: When 1 is written to ADCAnECR.ULEC.

Table 31.35 ADCAnULER Register Contents (2/2)

Bit Position	Bit Name	Function
11	MPXE*1	MPX Usage 0: The MPX function was not used when an upper limit/lower limit error occurred. 1: The MPX function was used when an upper limit/lower limit error occurred. Capture condition: When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0 Clearing condition: When 1 is written to ADCAnECR.ULEC.
10 to 8	MPXV[2:0]*1	The value of MPX is stored when the errors of the upper and lower limit occurred Capture condition: When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0. Clearing condition: When 1 is written to ADCAnECR.ULEC.
7	Reserved	When read, an undefined value is read.
6	Reserved	When read, the value after reset is returned.
5 to 0	ULECAP[5:0]	Upper Limit/Lower Limit Error Capture The physical channel is captured when an upper limit/lower limit error occurred. Capturing condition: When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0. Clearing condition: 1 is written to ADCAnECR.ULEC.

Note 1. These bits are only supported for ADCA0.
For ADCA1, when read, the value after reset is returned.

NOTE

ADCAnULER is updated when the A/D converted value is set in ADCAnDRj or ADCAnPWDTSNDR.

31.3.2.23 ADCAnOWER — Overwrite Error Register

This register is a 32/16/8-bit read-only register that indicates an overwrite error. The target for overwrite errors is SG1 to SG3, and not PWM-Diag.

Access: ADCAnOWER is a read-only register that can be read in 32-bit units.
ADCAnOWERL is a read-only register that can be read in 16-bit units.
ADCAnOWERLL is a read-only register that can be read in 8-bit units.

Address: ADCAnOWER: <ADCAn_base> + $34C_H$
ADCAnOWERL: <ADCAn_base> + $34C_H$
ADCAnOWERLL: <ADCAn_base> + $34C_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OWE	—	OWECAP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.36 ADCAnOWER Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	OWE	Overwrite Error Flag 0: An overwrite error is not detected. 1: An overwrite error is detected. Setting condition: ADCAnDIRj.WFLG = 1, and the A/D converted value is written to ADCAnDRj. If a subsequent overwrite error is detected in A/D conversion while this bit is set to 1, the ADCAnOWER register is not updated. Clearing condition: 1 is written to ADCAnECR.OWEC.
6	Reserved	When read, the value after reset is returned.
5 to 0	OWECAP[5:0]	Overwrite Error Capture The virtual channel number is captured when an overwrite error occurs. Capturing condition: OWE = 0 and ADCAnDIRj.WFLG = 1, and the A/D converted value is written to ADCAnDRj Clearing condition: 1 is written to ADCAnECR.OWEC.

NOTE

ADCAnOWER is updated when the A/D converted value is set in ADCAnDRj.

31.3.3 Scan Group (SG) Specific Registers

This section describes the registers provided for each scan group.

31.3.3.1 ADCAnSGSTCRx — Scan Group x Start Control Register

This register is used to control the start of scan group x. The bits are always read as 0.

Access: ADCAnSGSTCRx is a write-only register that can be written in 32-bit units.
ADCAnSGSTCRxL is a write-only register that can be written in 16-bit units.
ADCAnSGSTCRxLL is a write-only register that can be written in 8-bit units.

Address: ADCAnSGSTCRx: <ADCAn_base> + 400_H + x × 40_H
ADCAnSGSTCRxL: <ADCAn_base> + 400_H + x × 40_H
ADCAnSGSTCRxLL: <ADCAn_base> + 400_H + x × 40_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SGST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 31.37 ADCAnSGSTCRx Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SGST	Scan Group Start Trigger Writing 1 to SGST while ADCAnSGSTR.SGACT[3:1] = 0 starts the target SGx.

31.3.3.2 ADCAnSGCRx — Scan Group x Control Register

This register controls scan group x.

Access: ADCAnSGCRx can be read or written in 32-bit units.
ADCAnSGCRxL can be read or written in 16-bit units.
ADCAnSGCRxLL can be read or written in 8-bit units.

Address: ADCAnSGCRx: <ADCAn_base> + x × 40_H + 408_H
ADCAnSGCRxL: <ADCAn_base> + x × 40_H + 408_H
ADCAnSGCRxLL: <ADCAn_base> + x × 40_H + 408_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	SCANMD	ADIE	SCT[1:0]		—	TRGMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W

Table 31.38 ADCAnSGCRx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5	SCANMD	Scan Mode 0: Multicycle scan mode 1: Continuous scan mode Write 0 to this bit for SG2 and SG3.
4	ADIE	Scan End Interrupt Enable 0: INT_SGx is not output when the scan for SGx ends. 1: INT_SGx is output when the scan for SGx ends.
3, 2	SCT[1:0]	Channel Repeat Times Select 00: The selected number of channel repeat times is one. 01: The selected number of channel repeat times is two. 10: The selected number of channel repeat times is four. 11: Setting prohibited
1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	TRGMD	Trigger Mode 0: Trigger input to SGx_TRG is disabled (Hardware trigger disabled). 1: SGx_TRG start trigger or hold complete trigger A/B is selected for the trigger input to SGx. **NOTE** The software trigger is valid regardless of the TRGMD bit setting.

CAUTION

To prevent malfunction, ADCAnSGCRx should be set (except clearing TRGMD upon completion of A/D conversion) when SGACT of all scan groups is 0 (before the scan group is started) and TRGMD of all scan groups is 0.

31.3.3.3 ADCAnPWDSGCR — PWM-Diag Scan Group Control Register

This register is used to control PWM-Diag.

Access: ADCAnPWDSGCR can be read or written in 32-bit units.
ADCAnPWDSGCRL can be read or written in 16-bit units.
ADCAnPWDSGCRLL can be read or written in 8-bit units.

Address: ADCAnPWDSGCR: <ADCAn_base> + 508_H
ADCAnPWDSGCRL: <ADCAn_base> + 508_H
ADCAnPWDSGCRLL: <ADCAn_base> + 508_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWDTRGMD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 31.39 ADCAnPWDSGCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	PWDTRGMD	PWM-Diag Trigger Mode Select 0: PWSA_ADTRG trigger input is disabled. 1: PWSA_ADTRG is selected for the trigger input to the PWM-Diag scan group.

CAUTION

To prevent malfunction, ADCAnPWDSGCR should be set when SGACT of the PWM-Diag scan group (SG4) is 0 (before the scan group is started).

31.3.3.4 ADCAnSGVCSPx — Scan Group x Start Virtual Channel Pointer

This register specifies the start pointer of a virtual channel.

Access: ADCAnSGVCSPx can be read or written in 32-bit units.
ADCAnSGVCSPxL can be read or written in 16-bit units.
ADCAnSGVCSPxLL can be read or written in 8-bit units.

Address: ADCAnSGVCSPx: <ADCAn_base> + x × 40_H + $40C_H$
ADCAnSGVCSPxL: <ADCAn_base> + x × 40_H + $40C_H$
ADCAnSGVCSPxLL: <ADCAn_base> + x × 40_H + $40C_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	VCSP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.40 ADCAnSGVCSPx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	VCSP[5:0]	Start Virtual Channel Pointer These bits are used to specify the virtual channel from which the SGx scan is to be started.

CAUTION

- ADCAnSGVCSPx must be equal to or smaller than ADCAnSGVCEPx.
- When writing to the channel pointers, be sure to write in the following order: ADCAnSGVCSPx → ADCAnSGVCEPx. When SGx is started, the A/D conversion for the virtual channels within the range specified in ADCAnSGVCSPx and ADCAnSGVCEPx is executed.
- Though ADCAnSGVCSPx can be written during the A/D conversion, the register is updated at the time when ADCAnSGVCEPx is written. The new setting is applied when SGx is started next time.
- When the hardware trigger is used, writing to this register during operation is prohibited.

31.3.3.5 ADCAnSGVCEPx — Scan Group x End Virtual Channel Pointer

This register specifies the end pointer of a virtual channel.

Access: ADCAnSGVCEPx can be read or written in 32-bit units.
ADCAnSGVCEPxL can be read or written in 16-bit units.
ADCAnSGVCEPxLL can be read or written in 8-bit units.

Address: ADCAnSGVCEPx: <ADCAn_base> + x × 40_H + 410_H
ADCAnSGVCEPxL: <ADCAn_base> + x × 40_H + 410_H
ADCAnSGVCEPxLL: <ADCAn_base> + x × 40_H + 410_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	VCEP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.41 ADCAnSGVCEPx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
5 to 0	VCEP[5:0]	End Virtual Channel Pointer These bits are used to specify the virtual channel at which the SGx scan is to be ended.

CAUTION

- ADCAnSGVCSPx must be equal to or smaller than ADCAnSGVCEPx.
- When SGx is started, processing for the virtual channels within the range specified in ADCAnSGVCSPx and ADCAnSGVCEPx is executed.
  ADCAnSGVCEPx can be rewritten even when SGx is being processed. The new setting is applied when SGx is started next time.

31.3.3.6 ADCAnSGMCYCRx — Scan Group x Multicycle Register

This register is a 32/16/8-bit read/write register that indicates the number of scan times in multicycle scan mode.

Access: ADCAnSGMCYCRx can be read or written in 32-bit units.
ADCAnSGMCYCRxL can be read or written in 16-bit units.
ADCAnSGMCYCRxLL can be read or written in 8-bit units.

Address: ADCAnSGMCYCRx: <ADCAn_base> + x × 40_H + 414_H
ADCAnSGMCYCRxL: <ADCAn_base> + x × 40_H + 414_H
ADCAnSGMCYCRxLL: <ADCAn_base> + x × 40_H + 414_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCYC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 31.42 ADCAnSGMCYCRx Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	MCYC[1:0]	Multicycle Number Specification These bits are used to specify the number of scan times in multicycle scan mode. 00_B: Number of scans = 1 01_B: Number of scans = 2 10_B: Setting prohibited 11_B: Number of scans = 4

CAUTION

- To prevent malfunction, ADCAnSGMCYCRx should be set when SGACT of scan group x is 0 (before the scan group is started) and TRGMD is 0.
- When SGx is started, the scan for the virtual channels within the range specified in ADCAnSGVCSPx and ADCAnSGVCEPx is repeatedly executed as many times as specified in ADCAnSGMCYCRx.

31.3.3.7 ADCAnPWDSGSEFCR — PWM-Diag Scan End Flag Clear Register

This register is used to control the clearing of PWM-Diag scan end flag (SEF). The bits are always read as 0.

Access: ADCAnPWDSGSEFCR is a write-only register that can be written in 32-bit units.
ADCAnPWDSGSEFCRL is a write-only register that can be written in 16-bit units.
ADCAnPWDSGSEFCRLL is a write-only register that can be written in 8-bit units.

Address: ADCAnPWDSGSEFCR: <ADCAn_base> + 518_H
ADCAnPWDSGSEFCRL: <ADCAn_base> + 518_H
ADCAnPWDSGSEFCRLL: <ADCAn_base> + 518_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWDSEFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 31.43 ADCAnPWDSGSEFCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	PWDSEFC	PWM-Diag Scan End Flag Clear Trigger 0: No effect. 1: Clears the PWM-Diag scan end flag (ADCAnSGSTR.SEF[4]).

31.3.3.8 ADCAnSGSEFCRx — Scan Group x Scan End Flag Clear Register

This register is a write-only register that clears the scan end flag (ADCAnSGSTR.SEFx). The bits are always read as 0.

Access: ADCAnSGSEFCRx is a write-only register that can be written in 32-bit units.
ADCAnSGSEFCRxL is a write-only register that can be written in 16-bit units.
ADCAnSGSEFCRxLL is a write-only register that can be written in 8-bit units.

Address: ADCAnSGSEFCRx: <ADCAn_base> + x × 40_H + 418_H
ADCAnSGSEFCRxL: <ADCAn_base> + x × 40_H + 418_H
ADCAnSGSEFCRxLL: <ADCAn_base> + x × 40_H + 418_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEFC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 31.44 ADCAnSGSEFCRx Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	SEFC	Scan End Flag Clear Trigger 0: No effect. 1: Clears the target SG scan end flag (ADCAnSGSTR.SEF[3:1]).

31.3.3.9 ADCAnSGSTR — Scan Group Status Register

This register indicates the state of T&H, SVSTOP, scan group x, and PWM-Diag scan group. The SHACT and SGACT bits are cleared when HALT is executed.

Access: ADCAnSGSTR is a read-only register that can be read in 32-bit units.
ADCAnSGSTRL is a read-only register that can be read in 16-bit units.

Address: ADCAnSGSTR: <ADCAn_base> + 308_H
ADCAnSGSTRL: <ADCAn_base> + 308_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	SHACT	SGACT[5:1]					—	—	—	—	SEF[4:1]				—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.45 ADCAnSGSTR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 15	Reserved	When read, the value after reset is returned.
14	SHACT	T&H Status Flag 0: T&H is stopped. 1: T&H conversion or sampling is in progress.
13	SGACT[5]	SVSTOP Status Flag 0: SVSTOP is canceled. 1: SVSTOP is accepted.
12	SGACT[4]	PWM-Diag Scan Group (SG4) Status Flag 0: A/D conversion for PWM-Diag (SG4) is completed. 1: A/D conversion for PWM-Diag (SG4) is in processing or suspension.
11	SGACT[3]	Scan Group 3 (SG3) Status Flag 0: A/D conversion for SG3 is completed. 1: A/D conversion for SG3 is in processing or suspension.
10	SGACT[2]	Scan Group 2 (SG2) Status Flag 0: A/D conversion for SG2 is completed. 1: A/D conversion for SG2 is in processing or suspension.
9	SGACT[1]	Scan Group 1 (SG1) Status Flag 0: A/D conversion for SG1 is completed. 1: A/D conversion for SG1 is in processing or suspension.
8 to 5	Reserved	When read, the value after reset is returned.
4	SEF[4]	PWM-Diag Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: • ADCAnPWDTSNDR for PMW-Diag is read. • ADCAnPWDDIR for PWM-Diag is read. • ADCAnPWDSGSEFCR.PWDSEFC is written as 1. 1: The A/D conversion result is written to ADCAnPWDTSNDR for PWM-Diag.

Table 31.45 ADCAnSGSTR Register Contents (2/2)

Bit Position	Bit Name	Function
3	SEF[3]	SG3 Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: • ADCAnDRj for the virtual channel which ADCAnSGVCEP3 indicates is read. • ADCAnDIRj for the virtual channel which ADCAnSGVCEP3 indicates is read. • ADCAnSGSEFCRx.SEFC is written as 1. 1: The A/D conversion result is written to ADCAnDRj for the virtual channel which ADCAnSGVCEP3 indicates.
2	SEF[2]	SG2 Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: • ADCAnDRj for the virtual channel which ADCAnSGVCEP2 indicates is read. • ADCAnDIRj for the virtual channel which ADCAnSGVCEP2 indicates is read. • ADCAnSGSEFCRx.SEFC is written as 1. 1: The A/D conversion result is written to ADCAnDRj for the virtual channel which ADCAnSGVCEP2 indicates.
1	SEF[1]	SG1 Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: • ADCAnDRj for the virtual channel which ADCAnSGVCEP1 indicates is read. • ADCAnDIRj for the virtual channel which ADCAnSGVCEP1 indicates is read. • ADCAnSGSEFCRx.SEFC is written as 1. 1: The A/D conversion result is written to ADCAnDRj for the virtual channel which ADCAnSGVCEP1 indicates.
0	Reserved	When read, the value after reset is returned.

31.3.4 Hardware Trigger Specific Register

31.3.4.1 ADCAnSGTSELx — Scan Group x Start Trigger Control Register x

This register is used to select the A/D conversion trigger (hardware trigger) for SGx.

Access: ADCAnSGTSELx can be read or written in 32-bit units.
ADCAnSGTSELxL can be read or written in 16-bit units.

Address: ADCAnSGTSELx: <ADCAn_base> + x × 40_H + $41C_H$
ADCAnSGTSELxL: <ADCAn_base> + x × 40_H + $41C_H$

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TxSEL8 *1	TxSEL7 *1	TxSEL6 *1	TxSEL5 *1	TxSEL4 *1	TxSEL3 *1	TxSEL2 *1	TxSEL1 *1	TxSEL0 *1
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. ADCA1 supports only TxSEL0 to TxSEL3. When writing to the other bits, write the value after reset.

Table 31.46 ADCAnSGTSELx Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
8 to 0	TxSELp (p = 0 to 8)	A/D Conversion Trigger (Hardware Trigger) Select 0: Hardware trigger is disabled. 1: Hardware trigger is enabled. **CAUTION** When setting TxSELp to 1, set only one of the bits to 1.

The list below shows the hardware triggers to be selected.

Table 31.47 List of A/D Conversion Hardware Triggers (1/2)

	Control Register/Bit		Trigger Input Signal	
Unit	Register Name	Bit Name	Symbol	Connection Destination Unit
ADCA0	ADCA0SGTSEL1	T1SEL0	ADCA0TRG0	External trigger pin
		T1SEL1	INTTAUJ0I3	TAUJ0
		T1SEL2	INTTAUD0I7	TAUD0
		T1SEL3	INTTAUD0I15	TAUD0
		T1SEL4	SEQADTRG	LPS
		T1SEL5	INTENCA0I1	ENCA0
		T1SEL6	TAPATADOUT0	Motor control (TAPA0)
		T1SEL7	TAPATADOUT1	Motor control (TAPA0)
		T1SEL8	ADOPA0ADCATTIN00	Motor control (PIC0)

Table 31.47 List of A/D Conversion Hardware Triggers (2/2)

Unit	Control Register/Bit		Trigger Input Signal	
	Register Name	Bit Name	Symbol	Connection Destination Unit
ADCA0	ADCA0SGTSEL2	T2SEL0	ADCA0TRG1	External trigger pin
		T2SEL1	INTTAUJ0I3	TAUJ0
		T2SEL2	INTTAUD0I7	TAUD0
		T2SEL3	INTTAUD0I15	TAUD0
		T2SEL4	SEQADTRG	LPS
		T2SEL5	INTENCA0I1	ENCA0
		T2SEL6	TAPATADOUT0	Motor control (TAPA0)
		T2SEL7	TAPATADOUT1	Motor control (TAPA0)
		T2SEL8	ADOPA1ADCATTIN00	Motor control (PIC0)
	ADCA0SGTSEL3	T3SEL0	ADCA0TRG2	External trigger pin
		T3SEL1	INTTAUJ0I3	TAUJ0
		T3SEL2	INTTAUD0I7	TAUD0
		T3SEL3	INTTAUD0I15	TAUD0
		T3SEL4	SEQADTRG	LPS
		T3SEL5	INTENCA0I1	ENCA0
		T3SEL6	TAPATADOUT0	Motor control (TAPA0)
		T3SEL7	TAPATADOUT1	Motor control (TAPA0)
		T3SEL8	ADOPA2ADCATTIN00	Motor control (PIC0)
ADCA1	ADCA1SGTSEL1	T1SEL0	ADCA1TRG0	External trigger pin
		T1SEL1	INTTAUJ1I3	TAUJ1
		T1SEL2	INTTAUB0I7	TAUB0
		T1SEL3	INTTAUB0I15	TAUB0
	ADCA1SGTSEL2	T2SEL0	ADCA1TRG1	External trigger pin
		T2SEL1	INTTAUJ1I3	TAUJ1
		T2SEL2	INTTAUB0I7	TAUB0
		T2SEL3	INTTAUB0I15	TAUB0
	ADCA1SGTSEL3	T3SEL0	ADCA1TRG2	External trigger pin
		T3SEL1	INTTAUJ1I3	TAUJ1
		T3SEL2	INTTAUB0I7	TAUB0
		T3SEL3	INTTAUB0I15	TAUB0

CAUTIONS

1. When enabling the LPS trigger factor (SEQADTRG), select and enable only one of the ADCA0SGTSEL1.T1SEL4, ADCA0SGTSEL2.T2SEL4, and ADCA0SGTSEL3.T3SEL4.
2. To prevent malfunction, ADCAnSGTSELx should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

31.3.5 Self-Diagnosis Specific Registers

31.3.5.1 ADCAnDGCTL0 — Self-Diagnosis Control Register 0

This register controls the self-diagnostic voltage level.

Access: ADCAnDGCTL0 can be read or written in 32-bit units.
ADCAnDGCTL0L can be read or written in 16-bit units.
ADCAnDGCTL0LL can be read or written in 8-bit units.

Address: ADCAnDGCTL0: <ADCAn_base> + 350_H
ADCAnDGCTL0L: <ADCAn_base> + 350_H
ADCAnDGCTL0LL: <ADCAn_base> + 350_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PSEL2	PSEL1	PSEL0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 31.48 ADCAnDGCTL0 Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2 to 0	PSEL[2:0]	Self-Diagnostic Voltage Level Select

ADCAnDGCTL0			Output Signal			
PSEL2	PSEL1	PSEL0	ADDIAGOUT	DIAGOUT2	DIAGOUT1	DIAGOUT0
0	0	0	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	AnVSS	$2/3AnV_{REF}$	$1/2AnV_{REF}$	$1/3AnV_{REF}$
0	1	0	$1/3AnV_{REF}$	$1/3AnV_{REF}$	$2/3AnV_{REF}$	$1/2AnV_{REF}$
0	1	1	$1/2AnV_{REF}$	$1/2AnV_{REF}$	$1/3AnV_{REF}$	$2/3AnV_{REF}$
1	0	0	$2/3AnV_{REF}$	Hi-z	Hi-z	Hi-z
1	0	1	AnV_{REF}	$1/3AnV_{REF}$	$1/3AnV_{REF}$	$1/3AnV_{REF}$
1	1	0	AnV_{REF}	$1/2AnV_{REF}$	$1/2AnV_{REF}$	$1/2AnV_{REF}$
1	1	1	AnV_{REF}	$2/3AnV_{REF}$	$2/3AnV_{REF}$	$2/3AnV_{REF}$

31.3.5.2 ADCAnDGCTL1 — Self-Diagnosis Control Register 1

This register controls the self-diagnostic channel.

Access: ADCAnDGCTL1 can be read or written in 32-bit units.
ADCAnDGCTL1L can be read or written in 16-bit units.

Address: ADCAnDGCTL1: <ADCAn_base> + 354_H
ADCAnDGCTL1L: <ADCAn_base> + 354_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.49 ADCAnDGCTL1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15, 12, 9, 6, 3, 0	CDG [15, 12, 9, 6, 3, 0]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT0 is selected.
13, 10, 7, 4, 1	CDG [13, 10, 7, 4, 1]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT1 is selected.
14, 11, 8, 5, 2	CDG [14, 11, 8, 5, 2]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT2 is selected.

CAUTION

To prevent malfunction, ADCAnDGCTL1 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

31.3.5.3 ADCAnPDCTL1 — Pull Down Control Register 1

This register specifies a channel to which the pull down resistor is connected.

For details, see **Section 31.5.3, Diagnosis of Open Pins**.

Access: ADCAnPDCTL1 can be read or written in 32-bit units.
ADCAnPDCTL1L can be read or written in 16-bit units.

Address: ADCAnPDCTL1: <ADCAn_base> + 358_H
ADCAnPDCTL1L: <ADCAn_base> + 358_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDNA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.50 ADCAnPDCTL1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 0	PDNA[15:0]	Pull Down Enable Control These bits set whether an on-chip pull-down resistor is to be connected to the corresponding physical channel (ANIn[00:15]). 0: An on-chip pull-down resistor is not connected. 1: An on-chip pull-down resistor is connected.

CAUTION

To prevent malfunction, ADCAnPDCTL1 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

For on-chip pull-down resistor values, see the Electrical Characteristics section in the Data Sheet document.

31.3.5.4 ADCAnPDCTL2 — Pull Down Control Register 2

This register specifies a channel to which the pull down resistor is connected.

For details, see **Section 31.5.3, Diagnosis of Open Pins**.

Access: ADCAnPDCTL2 can be read or written in 32-bit units.
ADCAnPDCTL2H and ADCAnPDCTL2L can be read or written in 16-bit units.
ADCAnPDCTL2HL, ADCAnPDCTL2LH, and ADCAnPDCTL2LL can be read or written in 8-bit units.

Address: ADCAnPDCTL2: <ADCAn_base> + $35C_H$
ADCAnPDCTL2L: <ADCAn_base> + $35C_H$
ADCAnPDCTL2H: <ADCAn_base> + $35C_H$ + 2_H
ADCAnPDCTL2HL: <ADCAn_base> + $35C_H$ +2_H
ADCAnPDCTL2LL: <ADCAn_base> + $35C_H$
ADCAnPDCTL2LH: <ADCAn_base> + $35C_H$ + 1_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	PDNB[19:16]*1			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDNB[15:0]*1															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The PDNB[13:12] bits for ADCA0 are reserved. When writing, write the value after reset.

Table 31.51 ADCAnPDCTL2 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
19 to 0	PDNB[19:0]	Pull Down Enable Control These bits set whether the on-chip pull-down resistor is to be connected with the corresponding physical channel (ANIn[16:35]). 0: The on-chip pull-down resistor is not connected. 1: The on-chip pull-down resistor is connected.

CAUTION

To prevent malfunction, ADCAnPDCTL2 should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

For on-chip pull-down resistor values, see the Electrical Characteristics section in the Data Sheet document.

31.3.6 Emulation Specific Register

31.3.6.1 ADCAnEMU — Emulation Control Register

This register controls the SVSTOP disable signal.

Access: This register can be read or written in 8-bit units.

Address: <ADCAn_base> + 388_H

Value after reset: 00_H

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

Table 31.52 ADCAnEMU Register Contents

Bit Position	Bit Name	Function
7	SVSDIS	SVSTOP Disable 0: SVSTOP is enabled 1: SVSTOP is disabled For the A/D conversion when SVSTOP is enabled, see **Section 31.4.10.3, SVSTOP Operation**.
6 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

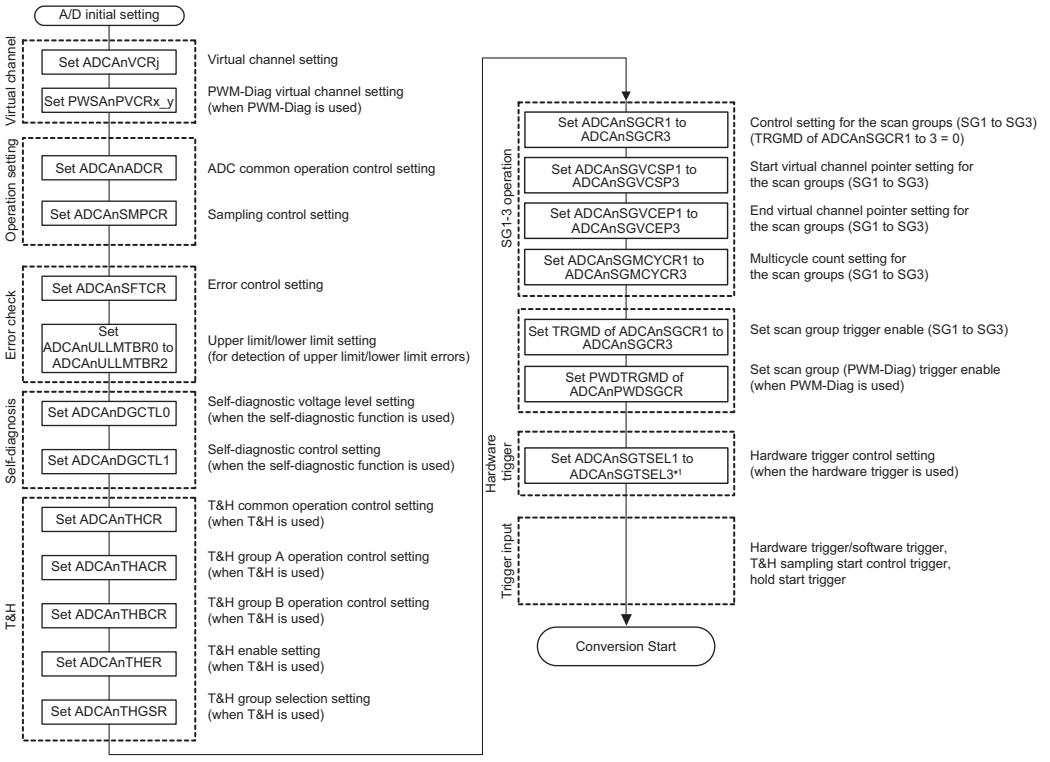
CAUTION

To prevent malfunction, SVSDIS should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

31.4 Operation

31.4.1 Initial Setting

Figure 31.6 shows an initial setting example of the A/D conversion. For trigger input, see **Figure 31.7** in the next section. For interrupt request signals, see **Section 31.4.12, Scan End Interrupt Request**.



Note: Set these registers as necessary.

Note 1. If triggers shown in **Table 31.47, List of A/D Conversion Hardware Triggers** are activated before setting of ADCAnSGTSEL1-3, apply notes of **Table 31.47**.

Figure 31.6 Flowchart for Initial Setting

31.4.2 Trigger Input

The following figure shows the flowchart for trigger input.

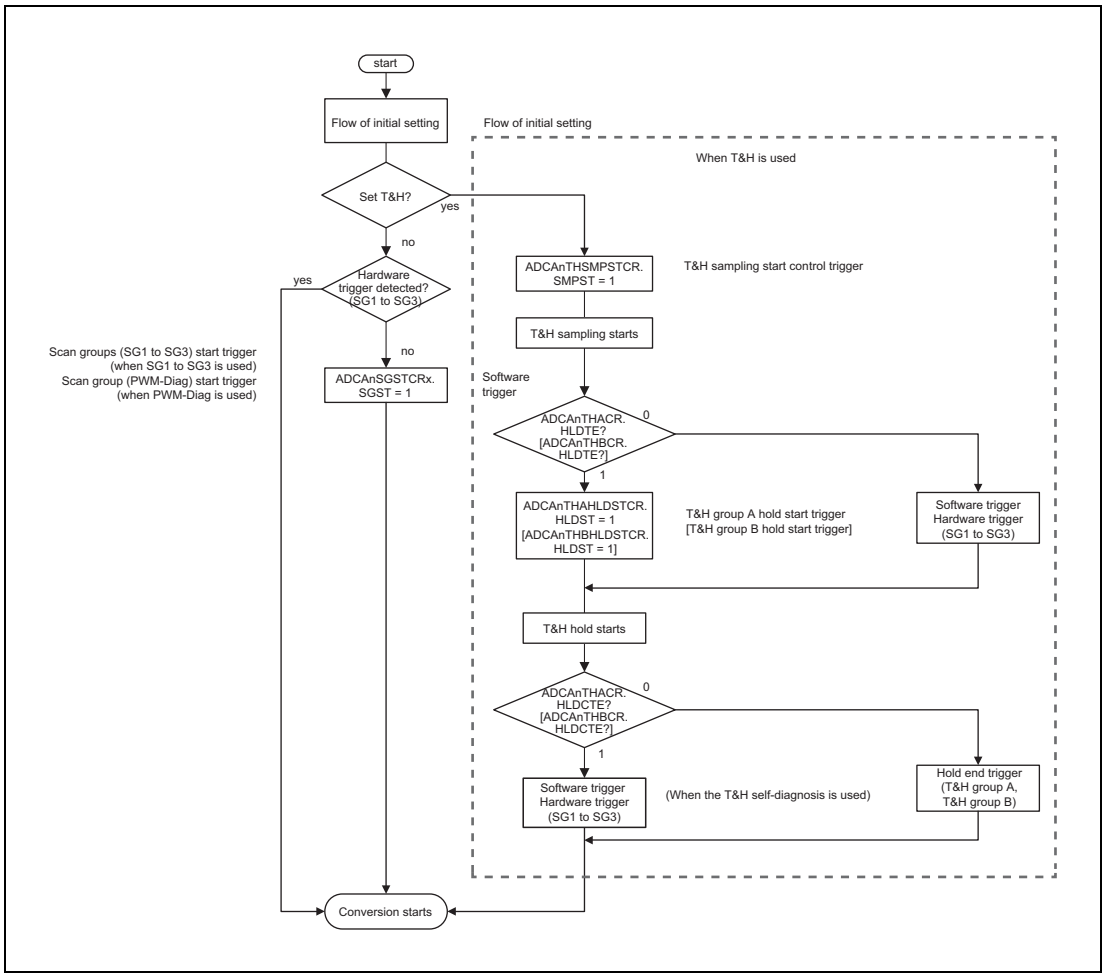


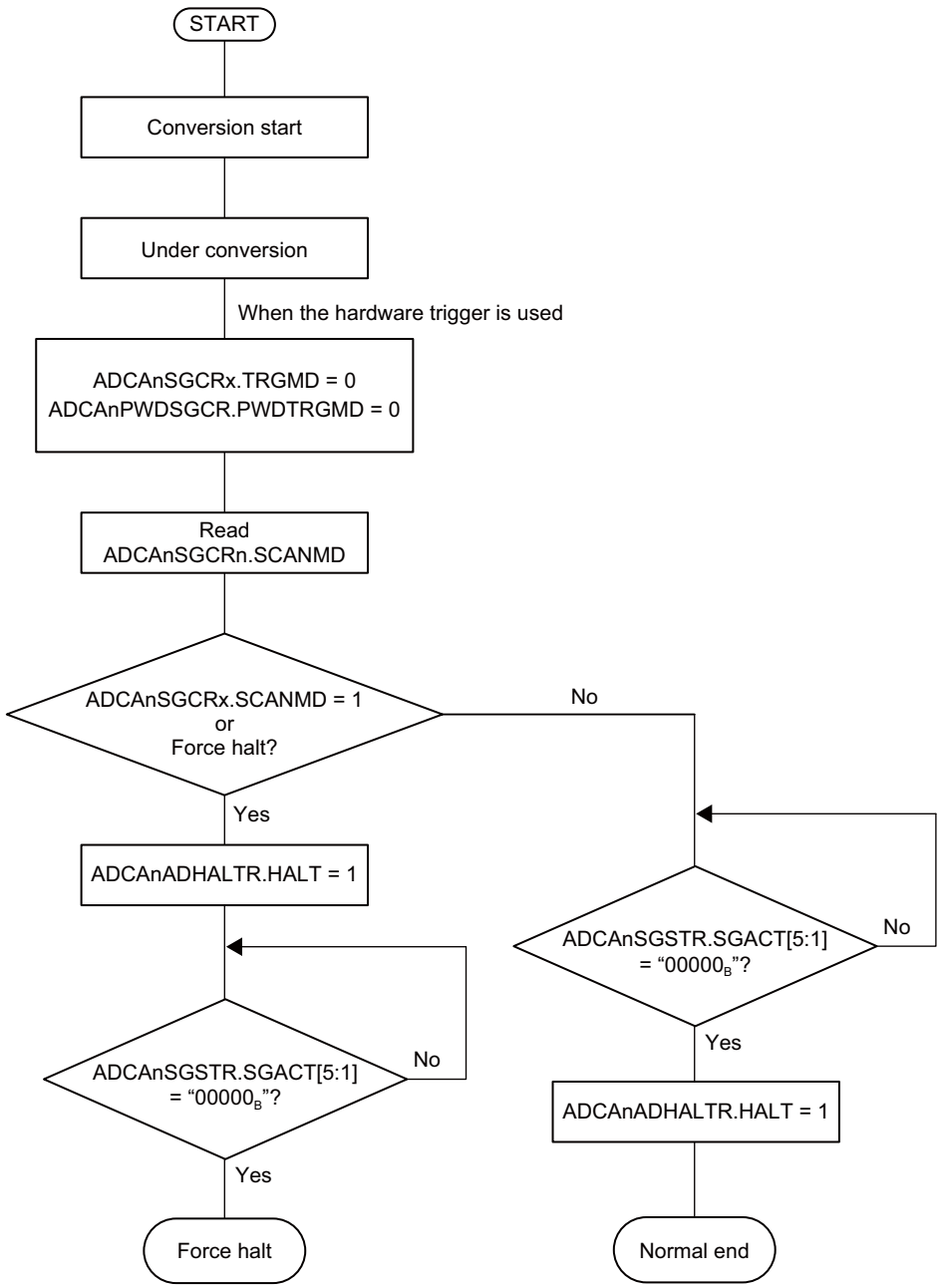
Figure 31.7 Flowchart for Trigger Input

NOTE

When an SG start trigger is generated during scanning, the SG start trigger is ignored.

31.4.3 Ending A/D Conversion

The flow for ending the A/D conversion is shown below.



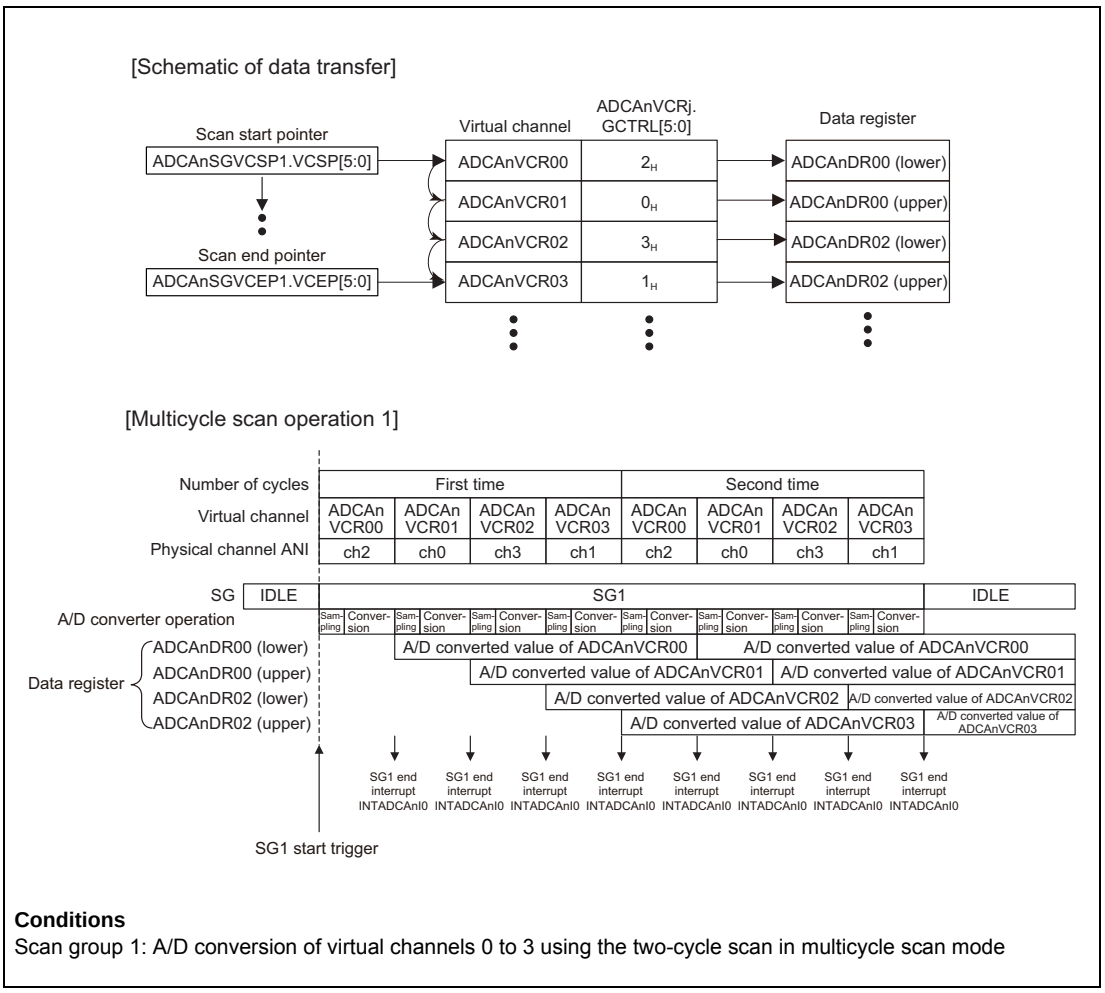
Note: The final setting of HALT when A/D conversion ends normally is done to clarify whether the status is "Conversion" or "IDLE". Whether HALT is set or not when A/D conversion ends normally does not effect the A/D conversion accuracy.

Figure 31.8 Flowchart for Ending A/D Conversion

31.4.4 Example of Scan Group Operation

(1) Multicycle scan mode

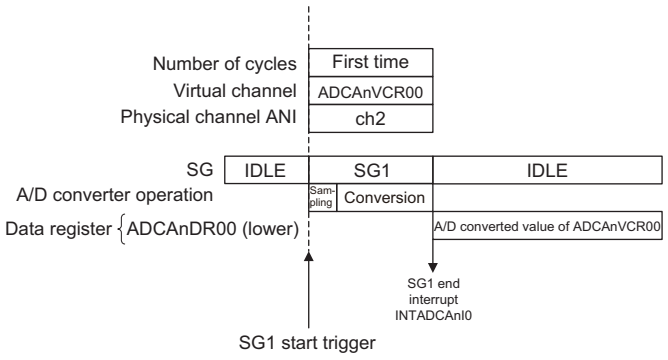
The following figure illustrates an operation example where four virtual channels of scan group 1 are converted using the two-cycle scan in multicycle scan mode.



Conditions
Scan group 1: A/D conversion of virtual channels 0 to 3 using the two-cycle scan in multicycle scan mode

Figure 31.9 Example of Multicycle Scan Operation 1

The following figure illustrates an operation example where a pin is scanned once in multicycle scan mode.



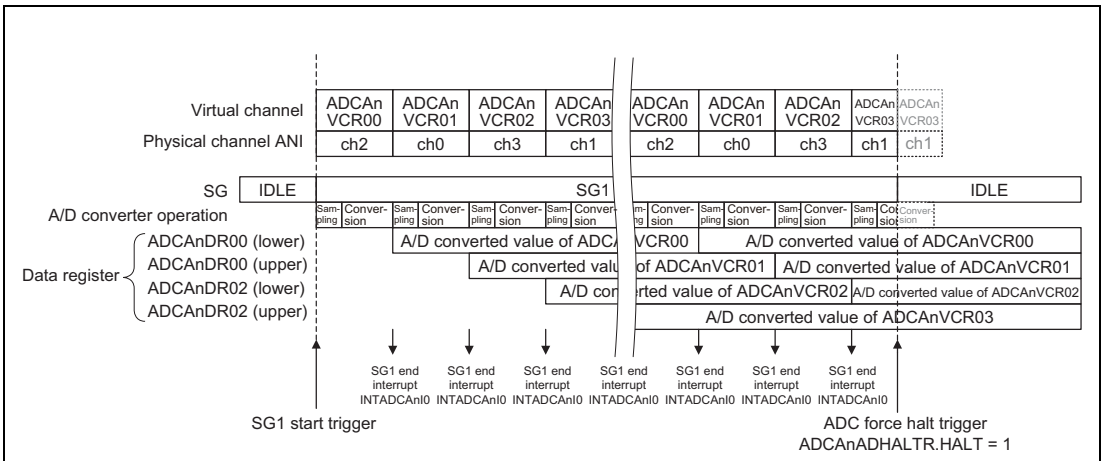
Conditions
Scan group 1: A/D conversion of virtual channel 0 in which a pin is scanned once in multicycle scan mode

Figure 31.10 Example of Multicycle Scan Operation 2

(2) Continuous scan mode

Continuous scan mode allows A/D conversion of the SG channels indicated by the pointers specified by ADCAnSGVCSPx.VCSP[5:0] to ADCAnSGVCEPx.VCEP[5:0] to continue until ADCAnADHALTR.HALT is asserted. This mode can be used only with SG1.

The following figure shows an example of operation in continuous scan mode.



Conditions
Scan group 1: A/D conversion of virtual channels 0 to 3 in continuous scan mode

Figure 31.11 Example of Continuous Scan Operation

31.4.5 Channel Repeat Mode

Channel repeat mode allows A/D conversion of the SG channel indicated by the pointer specified by ADCAnSGVCSPx.VCSP[5:0] to ADCAnSGVCEPx.VCEP[5:0] to repeat number of channel repeat times specified by ADCAnSGCRx.SCT[1:0]. This mode operates exclusively in each SG. The number of channel repeat times is selectable from 1, 2, and 4.

The following figures show examples of operation under respective conditions.

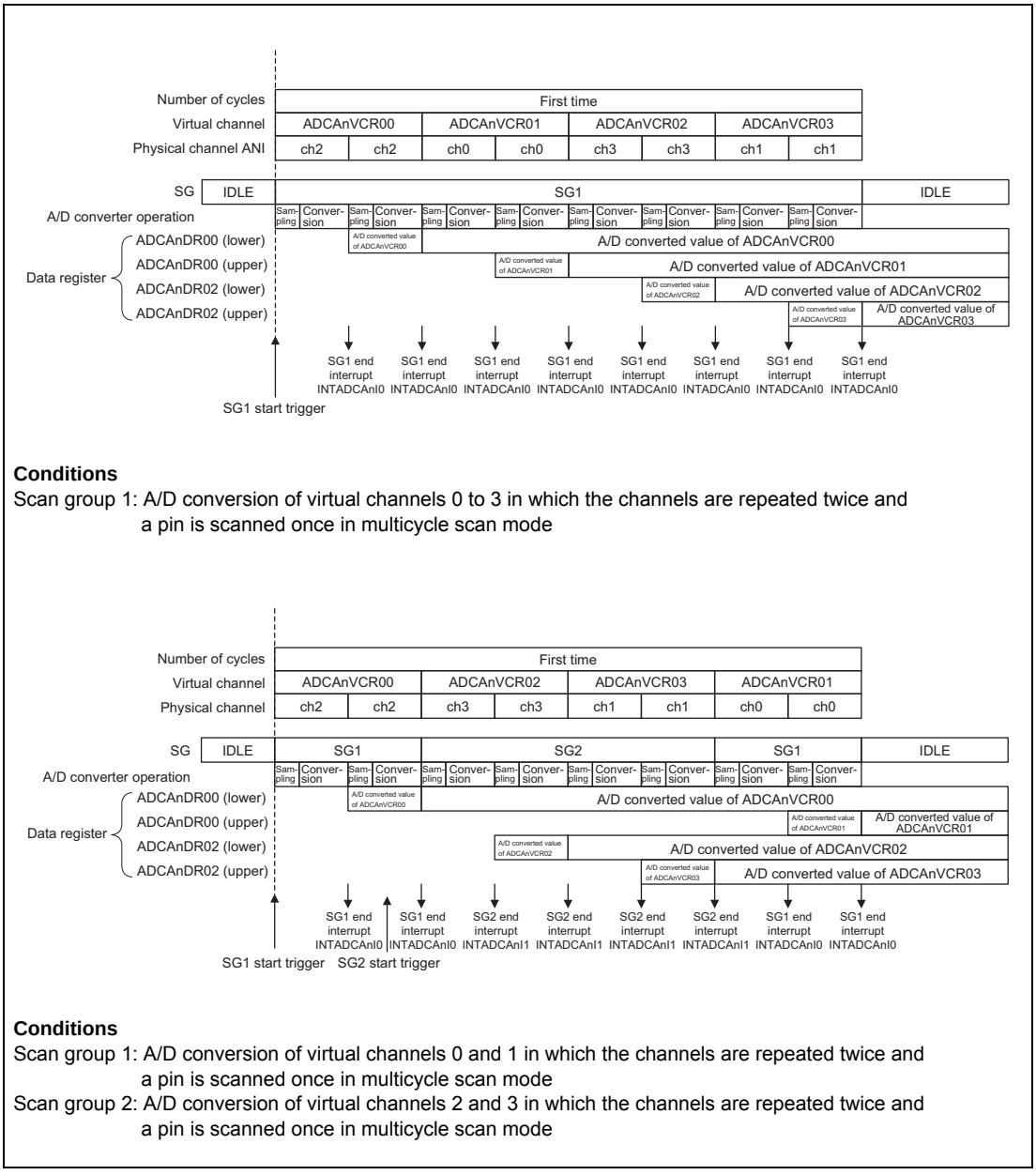
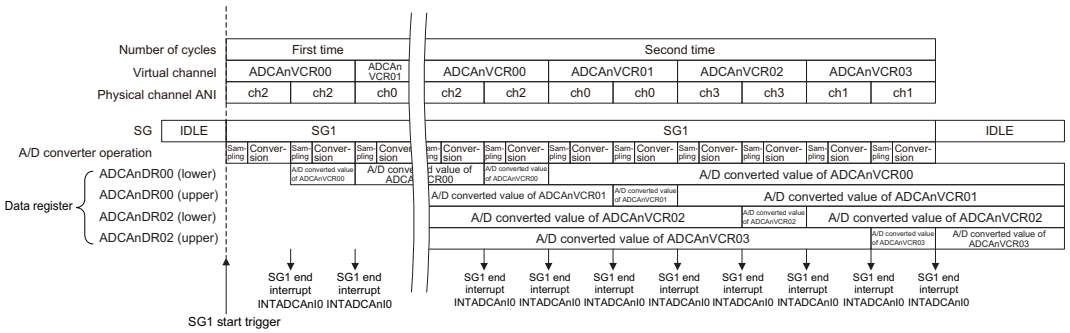


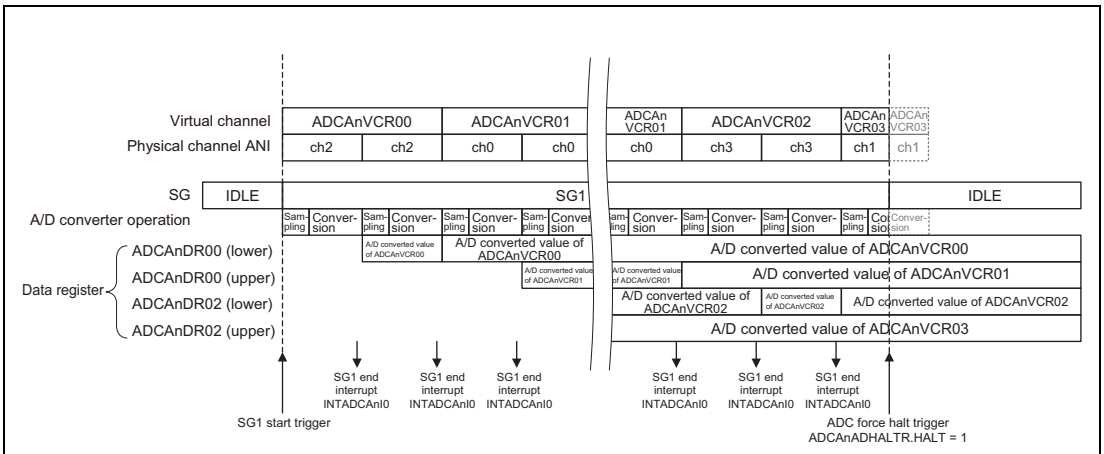
Figure 31.12 Example of Channel Repeat Operation 1



Conditions

Scan group 1: A/D conversion of virtual channels 0 to 3 in which the channels are repeated twice and the two-cycle scan is used in multicycle scan mode

Figure 31.13 Example of Channel Repeat Operation 2



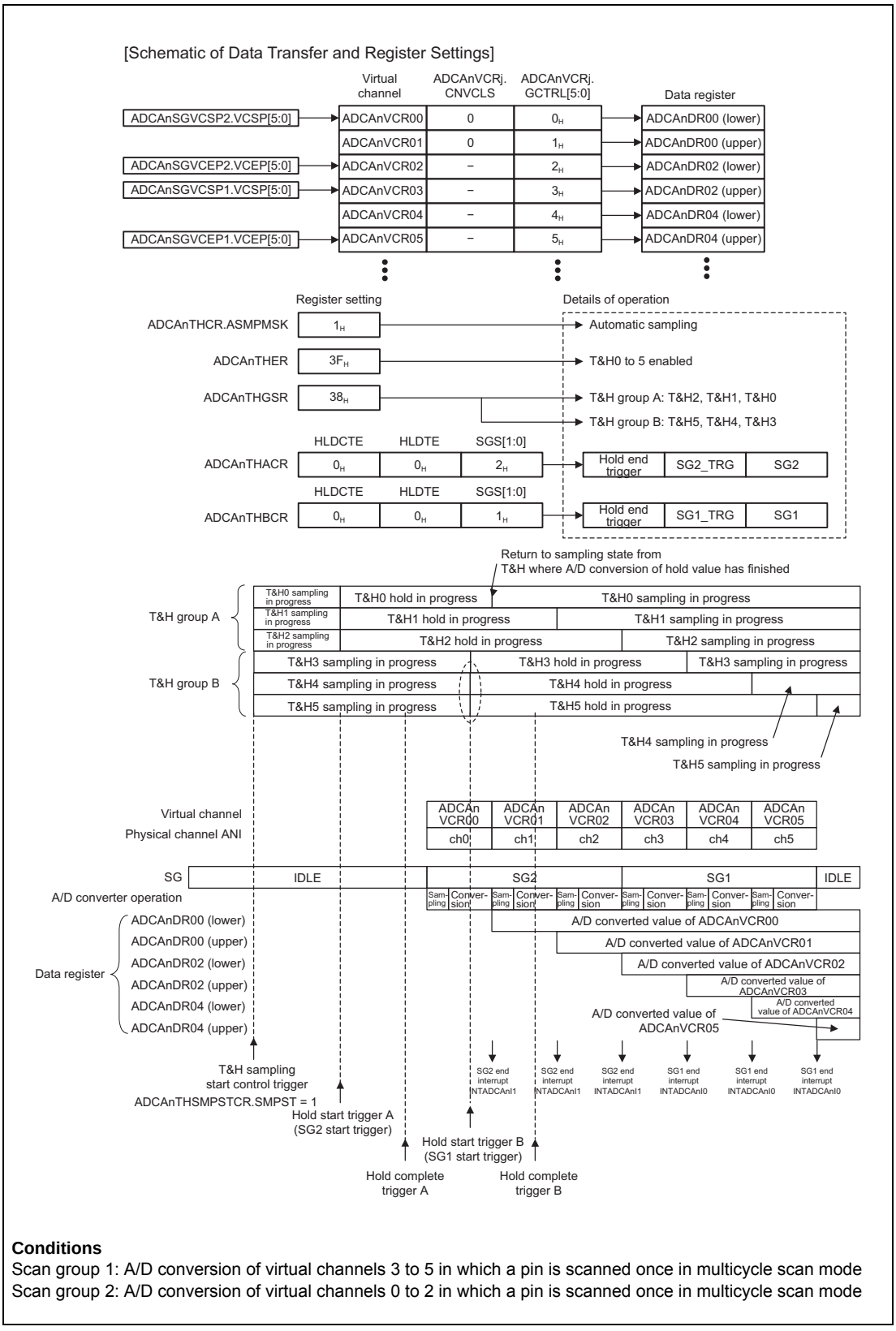
Conditions

Scan group 1: A/D conversion of virtual channels 0 to 3 in which the channels are repeated twice in continuous scan mode

Figure 31.14 Example of Channel Repeat Operation 3

31.4.6 Example of Simultaneous Track and Hold Operation

Figure 31.15 shows an operation example of simultaneous track and hold.



Conditions
Scan group 1: A/D conversion of virtual channels 3 to 5 in which a pin is scanned once in multicycle scan mode
Scan group 2: A/D conversion of virtual channels 0 to 2 in which a pin is scanned once in multicycle scan mode

Figure 31.15 Example of Simultaneous Track & Hold Operation 1

CAUTION

- **Do not specify the same physical channel in different groups.**
- **Two-cycle (or more) scan in multicycle scan mode and track & hold operation using continuous scan mode are prohibited.**
- **Because ADCAnTHSMPSTCR.SMPST is common to group A and group B, set SMPST after T&H operation for both group A and group B has been completed.**
- **If the hardware trigger is asserted before HLDCTE = 1 and HLDTE = 1 of ADCAnTHACR register or ADCAnTHBCR register are set and HLDST of ADCAnTHAHLDSTCR register or ADCAnTHBHLDSTCR register is written to hold T&H, scan operation starts. In that case, the channel switch opens with T&H staying in the sampling state. Therefore, all scan results are undefined. Do not assert the hardware trigger by setting HLDCTE = 1 and HLDTE = 1 before writing HLDST.**
- **Setting any channel from among 0 to 2 and any channel from among 3 to 5 in a same scan group is prohibited.**
- **Set the interval between T&H sampling start control trigger and hold start trigger to be 450 ns or more.**
- **Set the interval between hold start trigger and completion of the group A/D conversion to be 10 µs or less.**
  In suspend mode, do not use T&H for the channel of a scan group with low priority setting, if suspend time exceeds 10 µs.

31.4.7 A/D Conversion with External Analog Multiplexer

The following figures show examples of A/D conversion in each case.

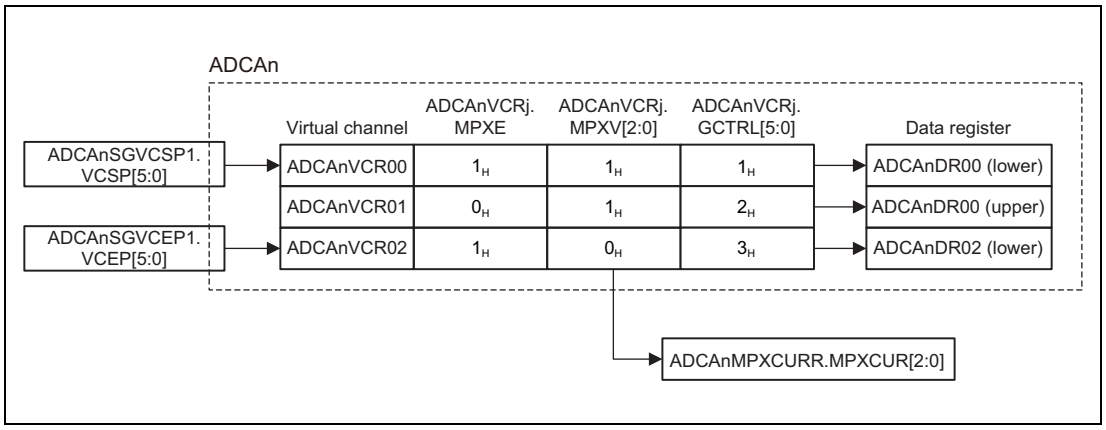
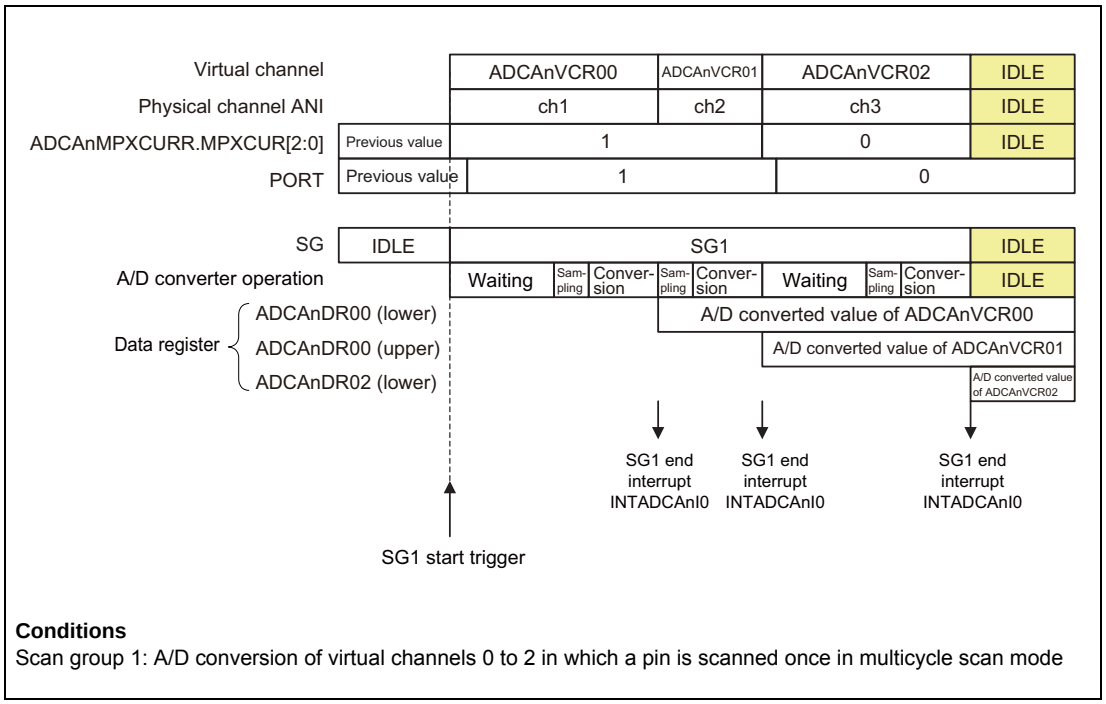
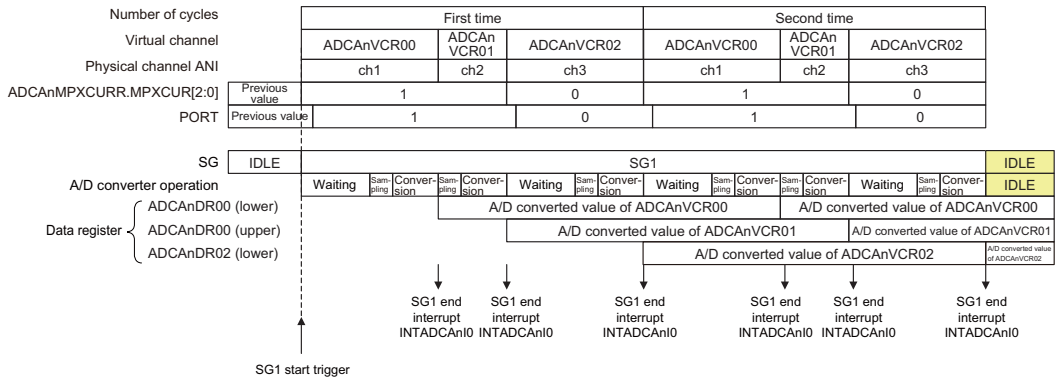


Figure 31.16 Schematic of Data Transfer and Register Settings



Conditions
Scan group 1: A/D conversion of virtual channels 0 to 2 in which a pin is scanned once in multicycle scan mode

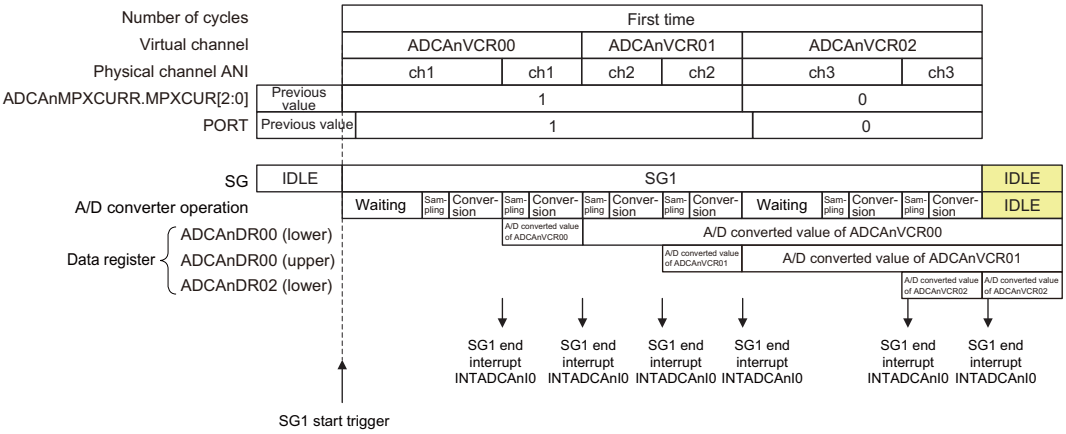
Figure 31.17 A/D Conversion 1 at an External Analog Multiplexer



Conditions
Scan group 1: A/D conversion of virtual channels 0 to 2 using the two-cycle scan in multicycle scan mode

Note: Waiting period is 1 A/D conversion time.

Figure 31.18 A/D Conversion 2 at an External Analog Multiplexer



Conditions
Scan group 1: A/D conversion of virtual channels 0 to 2 in which the channels are repeated twice and a pin is scanned once in multicycle scan mode

Figure 31.19 A/D Conversion 3 at an External Analog Multiplexer

31.4.7.1 A/D Conversion with PWM-Diag Enabled

With the PWM-Diag function enabled, A/D conversion is performed by the signal from the PWM-Diag.

For details on the PWM-Diag function, see **Section 30, PWM Output/Diagnostic (PWM-Diag)**.

To control the A/D conversion, the A/D converter receives the setting information on the MPX by the A/D conversion trigger select (PWSA) signal. The flow of A/D conversion with PWM-Diag is as follows.

(1) Set the channel MPX value of the MPX to ADCAnPWDVCR.MPXV[2:0]. Up to 8 channels can be specified to the MPX.

(2) The A/D conversion is started by the trigger signal PWSA_ADTRG from the PWM-Diag. In addition, when the MPX enable bit (ADCAnPWDVCR.MPXE) is 1, a wait of one A/D-conversion time is inserted before A/D conversion is performed.

(3) At the end of A/D conversion, the scan end is notified to the PWM-Diag.

CAUTION

As the trigger signal PWSA_ADTRG of PWM-Diag function has a higher-priority than SGx_TRG (x = 1 to 3), the operation of other scan groups may be kept waiting until PWM-Diag function is ended.

Figure 31.20 shows an example of PWM-Diag operation using an MPX.

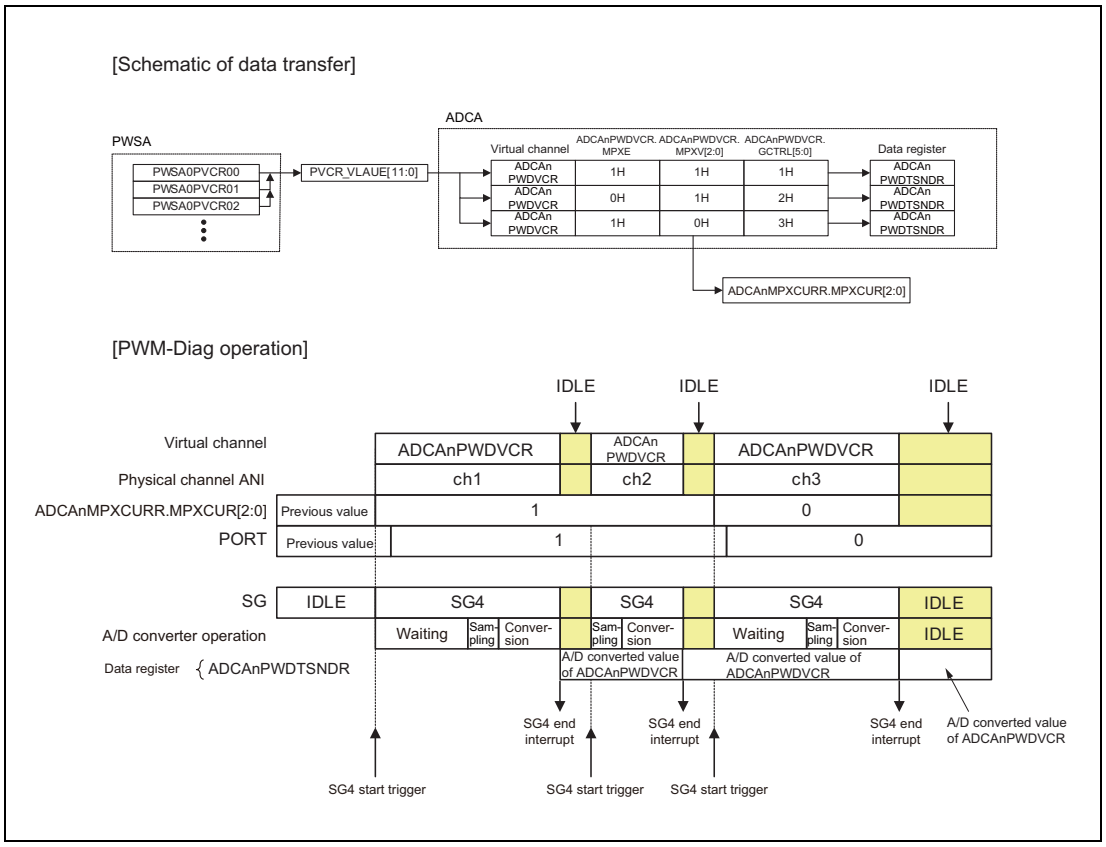


Figure 31.20 PWM-Diag Operation

31.4.8 Example of Synchronous Suspend and Resume Operation

Figure 31.21 shows an example of synchronous suspend and resume operation. In this example, a higher-priority SG interrupts a lower-priority SG.

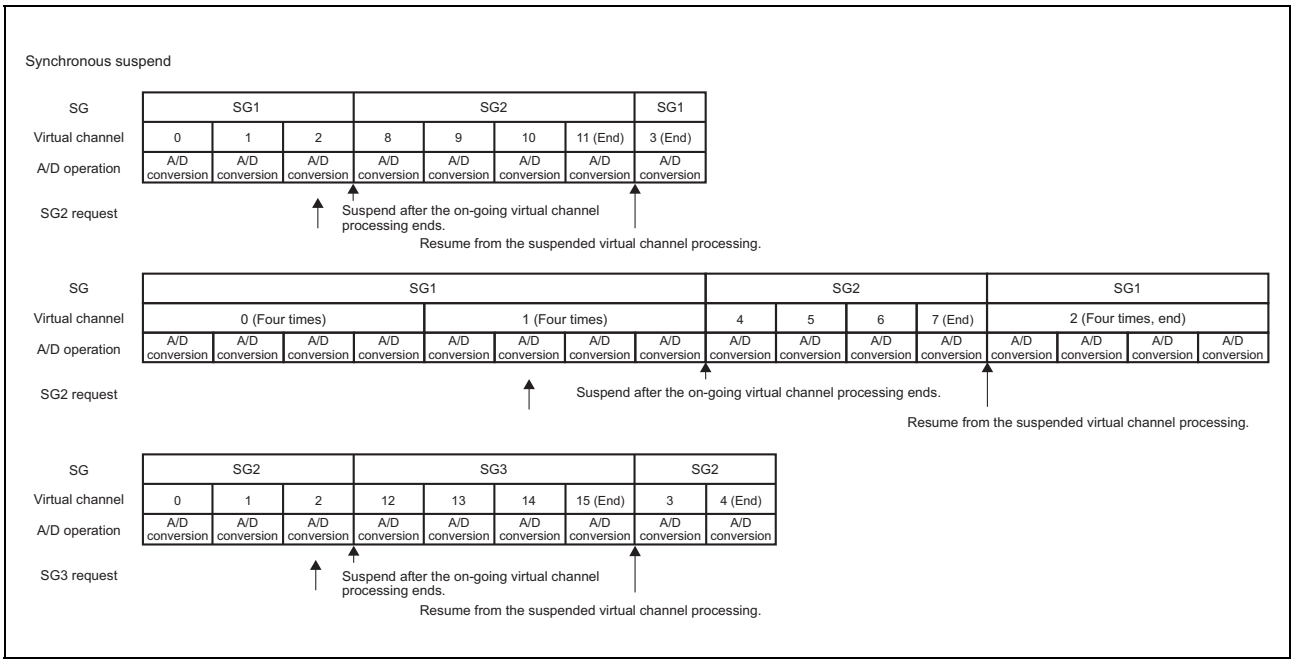


Figure 31.21 Example of Synchronous Suspend and Resume Operation

NOTE

Priority of scan groups is as follows.

Lower Higher

SG1 < SG2 < SG3 < PWM-Diag (SG4)

31.4.9 Example of Asynchronous Suspend and Resume Operation

Figure 31.22 shows an example of asynchronous suspend and resume operation. In this example, a higher-priority SG interrupts a lower-priority SG.

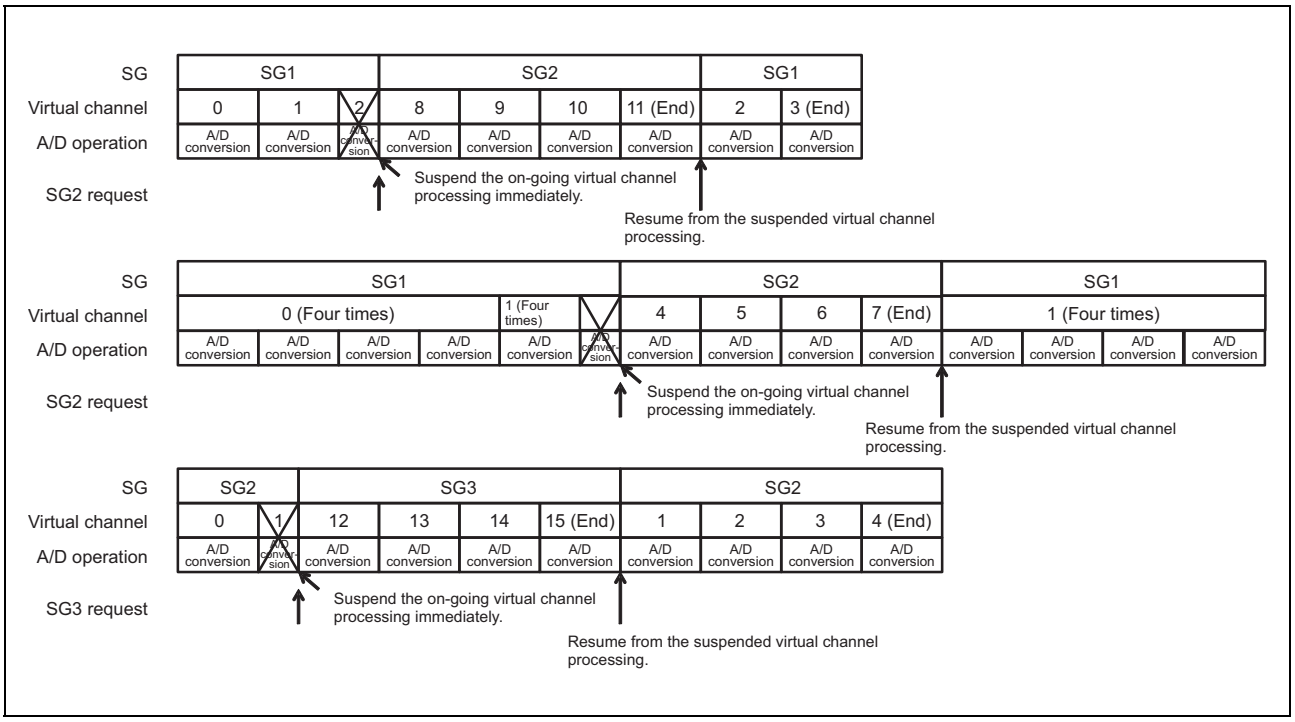


Figure 31.22 Example of Asynchronous Suspend and Resume Operation

NOTE

Priority of scan groups is as follows.

Lower Higher

SG1 < SG2 < SG3 < PWM-Diag (SG4)

31.4.10 Error Detecting Functions

ADCAn covers upper-limit error, lower-limit error, and overwrite error.

31.4.10.1 Upper-Limit/Lower-Limit Error Detecting Function

The upper-limit/lower-limit error detecting function determines whether the A/D converted data is larger than the upper-limit table ADCAnULLMTBR0.ULMTB[11:0] or smaller than the lower-limit table ADCAnULLMTBR0.LLMTB[11:0] at the end of A/D conversion.

31.4.10.2 Overwrite Error Detecting Function

If the ADCAnDIRj register or ADCAnDRj register of a virtual channel is not read while ADCAnDIRj.WFLG = 1 (A/D converted value is stored) and the next A/D converted value is written in the ADCAnDRj register, an overwrite error is detected.

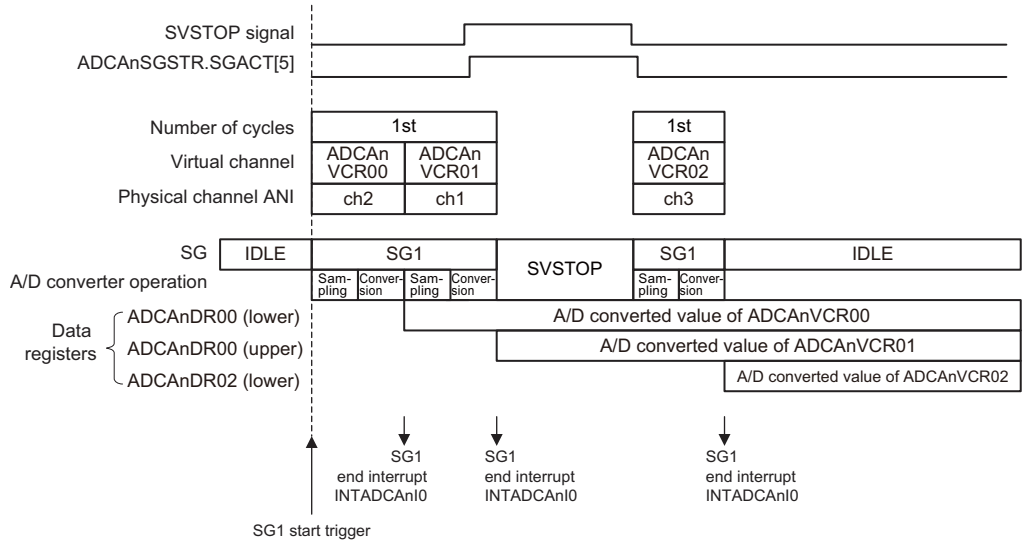
31.4.10.3 SVSTOP Operation

The SVSTOP function is supported by the SVSTOP signal sent from the on-chip debugger control unit. The SVSTOP function stops conversion of the A/D converter when the SVSTOP signal is input during an emulation break. While the SVSTOP signal is high, reading registers ADCAnDRj, ADCAnDIRj, ADCAnSGSTR, ADCAnULER, ADCAnOWER, ADCAnPWDTSNDR, and ADCAnPWDDIR by the external access does not affect these registers.

When the high level is input to SVSTOP while ADCAnEMU.SVSDIS = 0, ADCAnSGSTR.SGACT[5] is set to 1 to make a transition to the SVSTOP state. Hardware triggers and software triggers are valid in the SVSTOP state. When the high level is input to SVSTOP while ADCAnEMU.SVSDIS = 1, the ADCA does not make a transition to the SVSTOP state. ADHALT (forced termination of A/D conversion) should not be performed in the SVSTOP state.

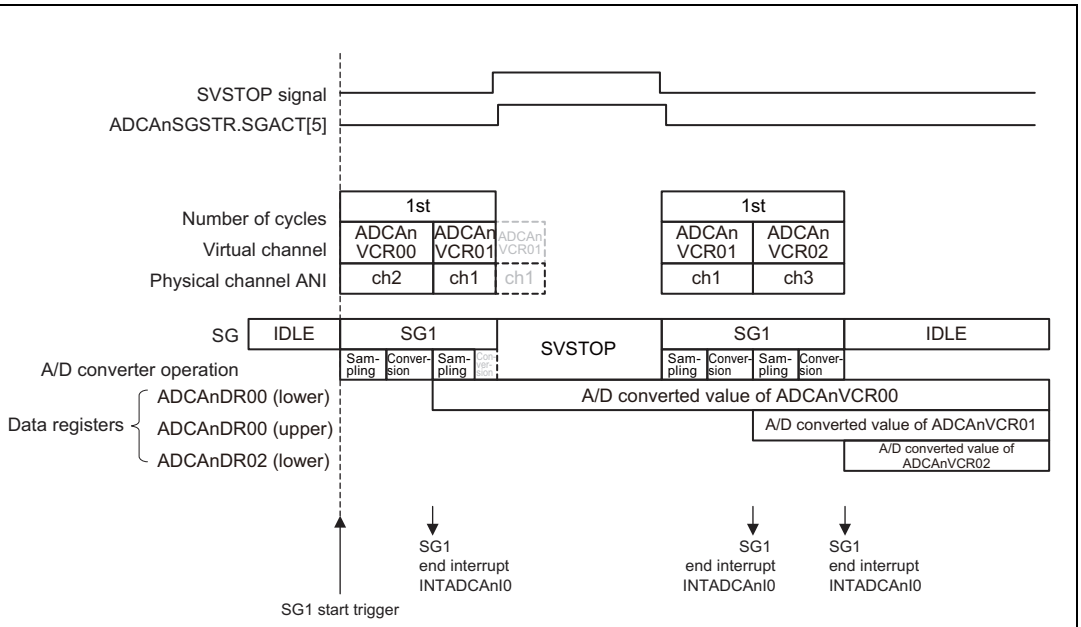
In operations for synchronous suspension, a new start trigger cannot be accepted over the time from when the high level is input to SVSTOP to the completion of conversion on the channel where conversion is currently proceeding. This time can be up to the time taken for one A/D conversion.

The following example illustrates a SVSTOP operation example.



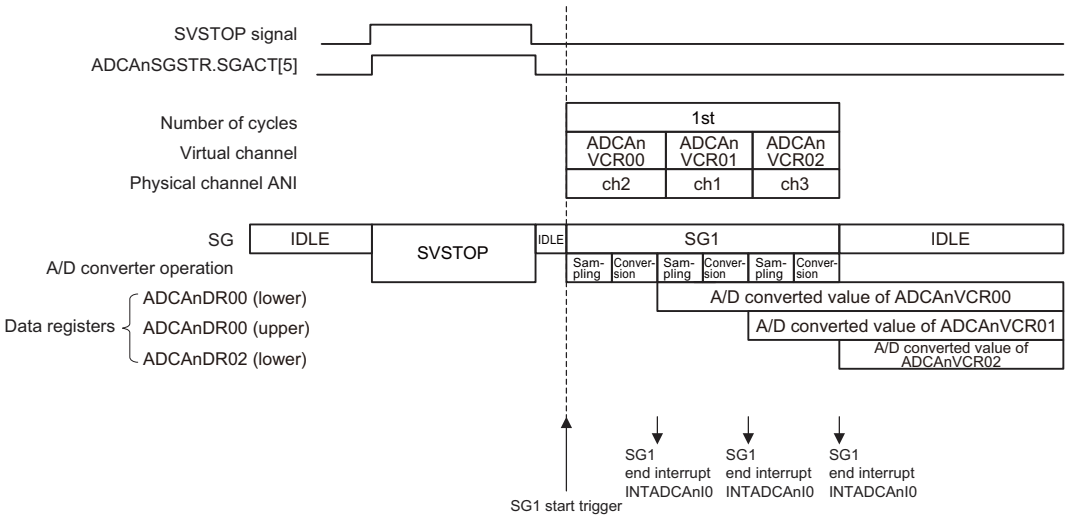
Note: An SG start trigger is accepted even during SVSTOP after the completion of conversion.

Figure 31.23 Example of SVSTOP Operation (ADCAnADCR.SUSMTD = 00 and ADCAnEMU.SVSDIS = 0)



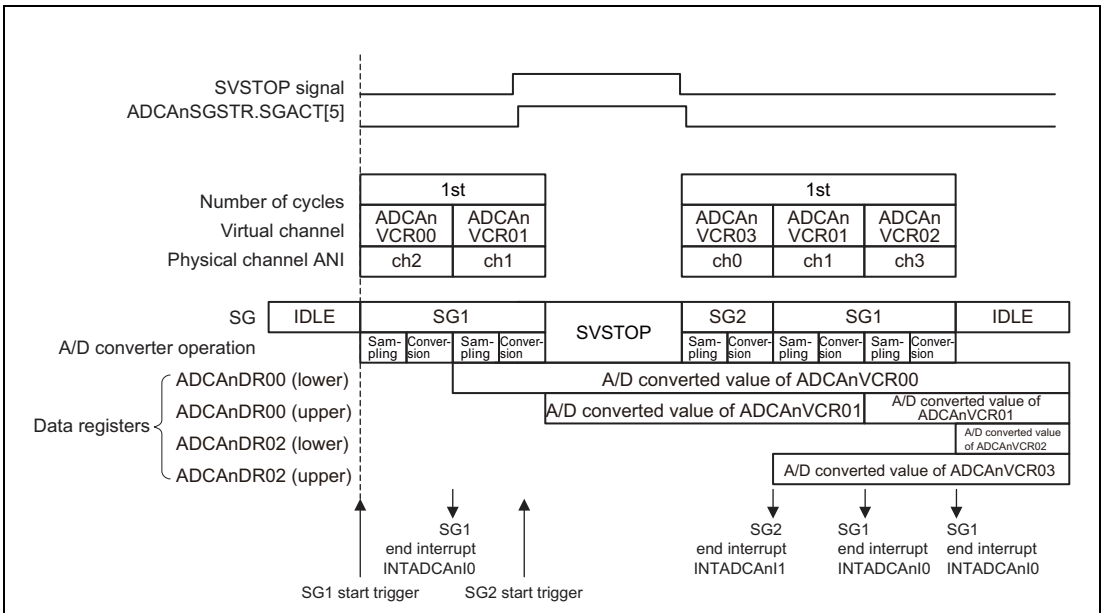
Note: An SG start trigger is accepted even during SVSTOP.

Figure 31.24 Example of SVSTOP Operation (ADCAnADCR.SUSMTD = 10 and ADCAnEMU.SVSDIS = 0)



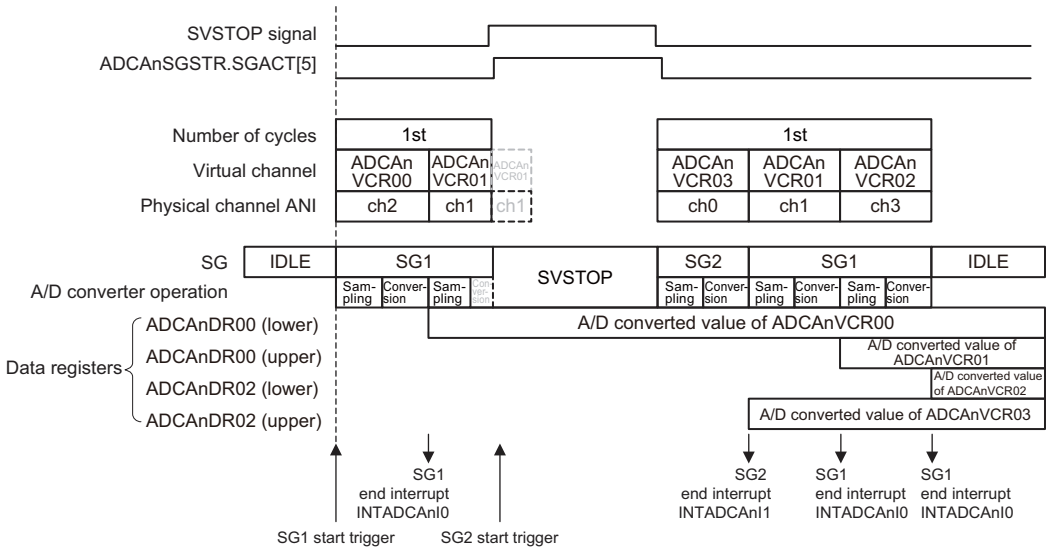
Note: An SG start trigger is accepted even during SVSTOP.

Figure 31.25 Example of SVSTOP Operation in the IDLE State (ADCAnADCR.SUSMTD = 00 and ADCAnEMU.SVSDIS = 0)



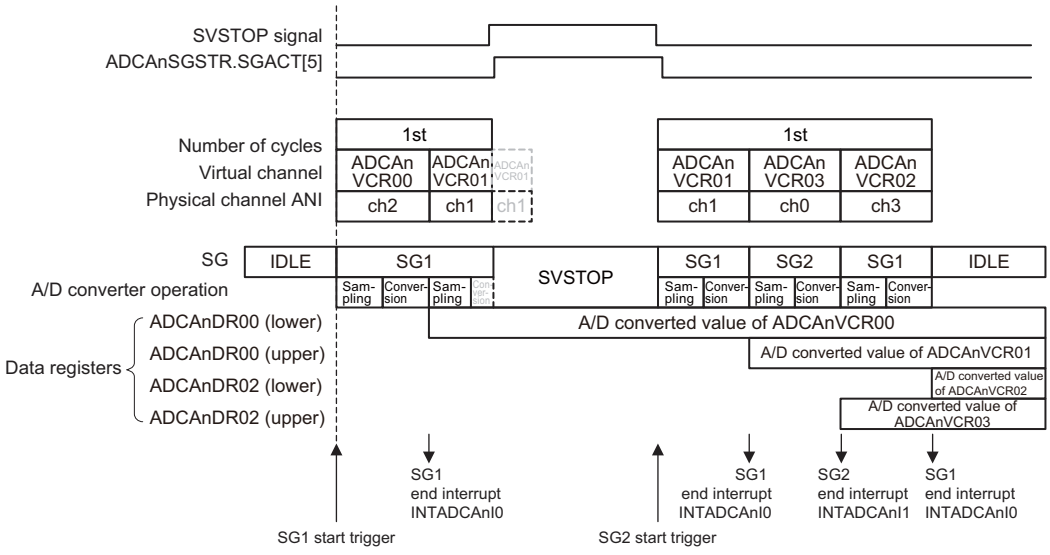
Note: Though the channel where conversion is currently proceeding is not stopped even after the acceptance of SVSTOP, an SG end interrupt is not generated.
This channel is converted again after SVSTOP has been cleared.
After the completion of conversion, the SG start trigger is accepted even during SVSTOP.

Figure 31.26 Conflict of SVSTOP Start and High-Priority SG Start Trigger (ADCAnADCR.SUSMTD = 00, ADCAnEMU.SVSDIS = 0)



Note: An SG start trigger is accepted even during SVSTOP.

Figure 31.27 Conflict of SVSTOP Start and High-Priority SG Start Trigger (ADCAnADCR.SUSMTD = 10, ADCAnEMU.SVSDIS = 0)



Note: An SG start trigger is accepted even during SVSTOP.
In cases of conflict between clearing of SVSTOP and setting of the scan group start trigger bit, conversion of the suspended scan group is resumed regardless of its priority.

Figure 31.28 Conflict of SVSTOP Clear and High-Priority SG Start Trigger (ADCAnADCR.SUSMTD = 10, ADCAnEMU.SVSDIS = 0)

31.4.11 Activating Scan Group by a Hardware Trigger

Scan group x can be activated by the hardware trigger input to SGx_TRG. As for the hardware trigger sources to be used, see **Table 31.47, List of A/D Conversion Hardware Triggers**. When activating SGx_TRG by the hardware trigger, set the peripheral function to be used by the trigger and set the start trigger in the A/D conversion trigger select control register (ADCAnSGTSELx).
A hardware trigger from external trigger input pin requires digital filter setting. For details, see **Section 2.12, Noise Filter & Edge/Level Detector**. More than one start trigger can be specified.

31.4.11.1 Stopping Scan Group by ADHALT

Setting ADCAnADHALTR.HALT (A/D force halt trigger) to 1 forcibly halts the A/D conversion and clears the scan group status register (ADCAnSGSTR). The error flag of ADCAnULER (upper limit/ lower limit error register) is not cleared. When ADCAnADHALTR.HALT is set, make sure that ADCAnSGSTR.SGACT has been cleared.

31.4.12 Scan End Interrupt Request

Scan group x can issue a scan end interrupt request (INT_SGx) to INTC. If ADIE of ADCAnSGCRx is set to 1, INT_SGx can be output after the SGx scan ends. If ADIE of ADCAnSGCRx is set to 0, the INT_SGx output when the SGx scan ends can be disabled. If ADIE of ADCAnVCRj is set to 1, INT_SGx can be output when A/D conversion for virtual channel j in SGx ends. If ADIE of ADCAnVCRj is set to 0, the INT_SGx output when A/D conversion for virtual channel j in SGx ends can be disabled. Since SGx scan ending is simultaneous with A/D conversion ending for virtual channel j in SGx when ADIEs of both ADCAnSGCRx and ADCAnVCRj are set to 1, the INT_SGx occurs only once.

Example 1: A scan is executed for virtual channel 0 or 1 in SG1 when ADIE of ADCAnSGCR1 is 0, ADIE of VCR0 is 1, and ADIE of VCR1 is 0.
INT_SG1 is output when A/D conversion ends for virtual channel 0.

Example 2: A scan is executed for virtual channel 0 or 1 in SG2 when ADIE of ADCAnSGCR2 is 0, ADIE of VCR0 is 1, and ADIE of VCR1 is 1.
INT_SG2 is output when A/D conversion ends for virtual channel 0 and virtual channel 1.

Example 3: A scan is executed for virtual channel 0 or 1 in SG3 when ADIE of ADCAnSGCR3 is 1, ADIE of VCR0 is 0, and ADIE of VCR1 is 0.
INT_SG3 is output when a scan ends (when A/D conversion for virtual channel 1 ends).

Furthermore, the DMAC can be started when scan ends.

For the setting of DMAC, see **Section 8, DMA**.

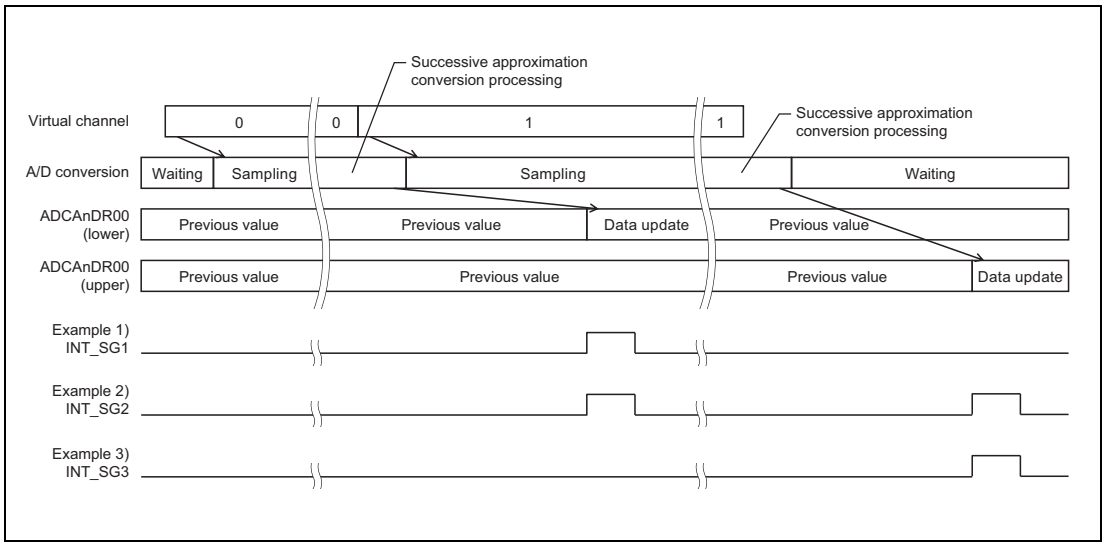


Figure 31.29 Example of a Scan Conversion End Interrupt Occurrence

31.4.13 A/D Error Interrupt Request

ADCA can issue an A/D error interrupt request (INT_ADE) to INTC. For an error source for which ULEIE and OWEIE of ADCAnSFTCR are set to 1, the OR condition of the error source is issued as INT_ADE. For an error source for which ULEIE and OWEIE of ADCAnSFTCR are set to 0, INT_ADE does not output interrupt.

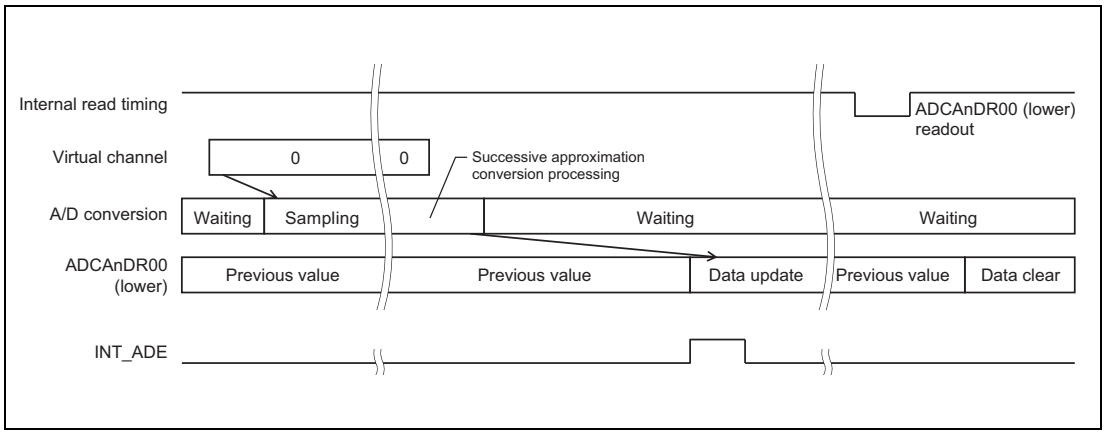


Figure 31.30 A/D Error Interrupt (Example: Overwrite Error)

31.5 Self-Diagnostic Function

To check the ADCAn function, the following self-diagnostic functions are available.

Section 31.5.1, Diagnosis of A/D Conversion Circuit

Section 31.5.2, Diagnosis of Channel Multiplexer

Section 31.5.3, Diagnosis of Open Pins

Section 31.5.4, Diagnosis of T&H Circuit

The overview of the self-diagnostic functions is shown in the figure below. The detailed description is given in the following sections.

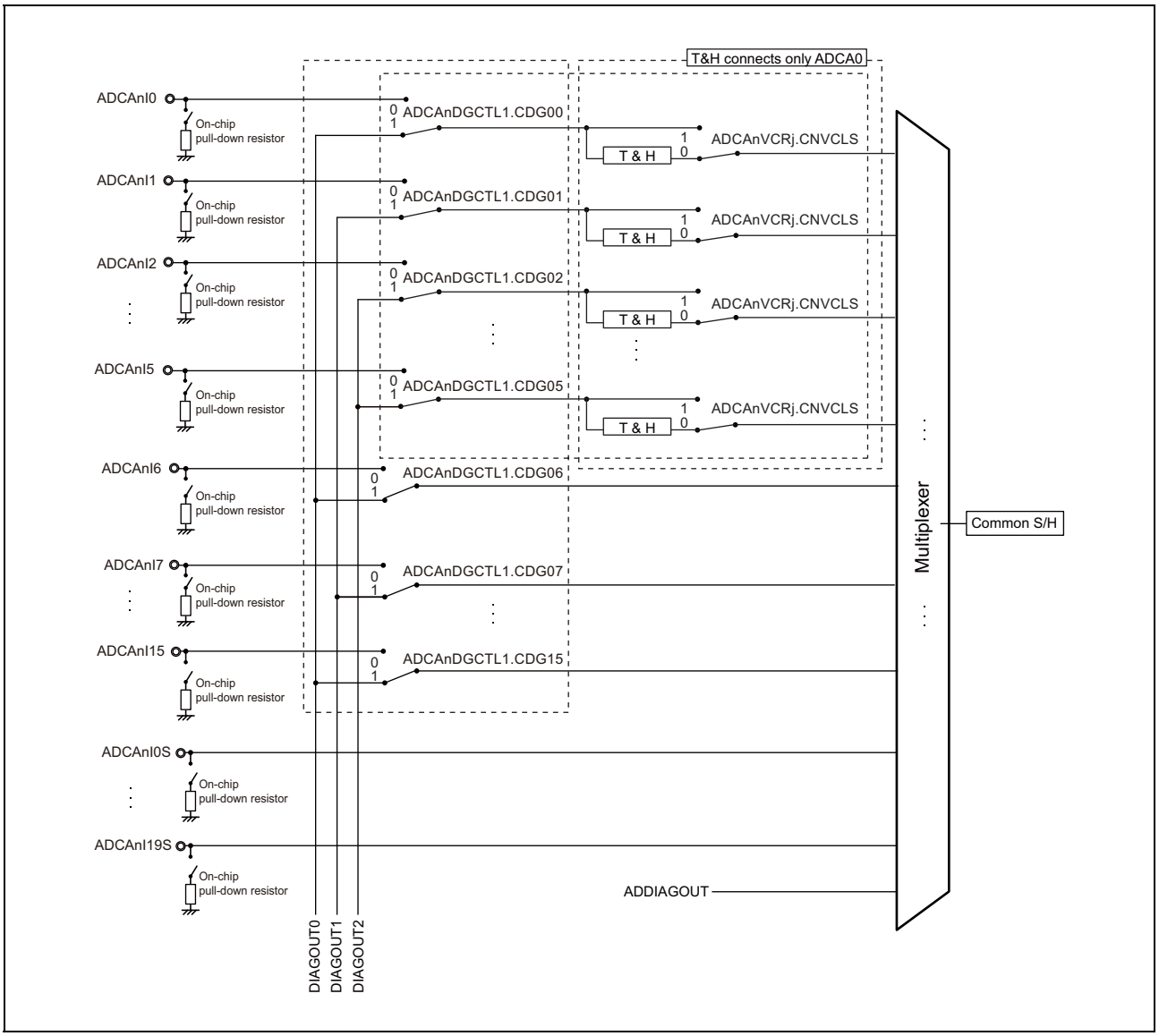


Figure 31.31 Overview of Self-Diagnostic Functions

NOTE

The functions in the dashed-line frames depend on the product.

31.5.1 Diagnosis of A/D Conversion Circuit

This function checks whether the A/D converter is operating normally by verifying the A/D conversion for self-diagnostic voltage (ADDIAGOUT) and the result of conversion. If the result of A/D conversion differs from the expected value, an internal circuit may be broken. The features of self-diagnosis of the A/D converter are as follows:

- As the self-diagnostic voltage (ADDIAGOUT) level, AnV_{REF}, $2/3AnV_{REF}$, $1/3AnV_{REF}$, $1/2AnV_{REF}$, and AnVSS are selectable by the PSEL[2:0] bits in the ADCAnDGCTL0 register.
- Self-diagnosis of the A/D converter is enabled by performing A/D conversion on one of SG1 to SG3.

31.5.1.1 Diagnostic procedure

The diagnostic procedures are shown below.

Common settings for ADC should be made before self-diagnosis is to proceed.

1. Set ADCAnADCR.DGON = 1 to enable the self-diagnostic voltage circuit.
2. Wait for 500 ns.
3. Set ADCAnDGCTL0.PSEL[2:0] to select a self-diagnostic voltage level.
4. Set ADCAnADCR.DGON = 1 to update the voltage level.
5. Wait for 500 ns.
6. Set an arbitrary bit of ADCAnVCRj.GCTRL[5:0] to 100100_B to select the diagnosis channel.
7. Set ADCAnVCRj.ADIE = 1 to enable the A/D conversion end interrupt.
8. Set ADCAnSGVCSPx to specify the start pointer of virtual channel.
9. Set ADCAnSGVCEPx to specify the end pointer of virtual channel.
10. Generate the start trigger of scan group to perform the A/D conversion.
11. When the conversion interrupt occurs, read the result and compare it with the expected one.
12. If the result is the expected one, the A/D conversion was performed successfully.

NOTE

- During A/D conversion, the self-diagnostic voltage level can be changed by writing to ADCAnDGCTL0.PSEL[2:0]. However, the value of ADCAnDGCTL0.PSEL[2:0] becomes effective from the next A/D conversion.
- To clear ADCAnADCR.DGON, follow the procedure below:
  1. Confirm that SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
  2. Clear ADCAnDGCTL0.PSEL[2:0].
  3. Clear ADCAnADCR.DGON.

31.5.2 Diagnosis of Channel Multiplexer

This function checks whether the path from the analog input to the A/D converter is normal.
Set the A/D conversion reference voltage (DIAGOUT0, DIAGOUT1, DIAGOUT2) by ADCAnDGCTL0.PSEL[2:0] and the channels to be connected by the ADCAnDGCTL1 register to perform A/D conversion using multiple analog channels.
If the result of A/D conversion differs from the expected value, an internal circuit may be broken. The features of self-diagnosis of the channel multiplexer are as follows:

- Channels for diagnosis can be arbitrarily selected from among ADCA0I0 to ADCA0I15 and ADCA1I0 to ADCA1I15.
- As the self-diagnostic voltage level, $2/3AnV_{REF}$, $1/3AnV_{REF}$, and $1/2AnV_{REF}$ are selectable and one of the three reference voltage levels can be allocated to each channel.

Table 31.53 Selection of Channel to be Diagnosed

Connection	Select Channel
DIAGOUT0	Channels 0, 3, 6, 9, 12, and 15
DIAGOUT1	Channels 1, 4, 7, 10, and 13
DIAGOUT2	Channels 2, 5, 8, 11, and 14

- Self-diagnosis of the channel multiplexer is enabled by performing A/D conversion on one of SG1 to SG3 by using multiple channels.

31.5.2.1 Diagnostic procedure

The diagnostic procedure is shown below.

Common settings for ADC should be made before self-diagnosis is to proceed.

1. Set ADCAnADCR.DGON = 1 to enable the self-diagnostic voltage circuit.
2. Wait for 500 ns.
3. Set ADCAnDGCTL0.PSEL[2:0] to select a self-diagnostic voltage level.
4. Set ADCAnADCR.DGON = 1 to update the voltage level.
5. Wait for 500 ns.
6. Use two or more ADCAnVCRj registers.
   Set ADCAnVCRj.GCTRL[5:0] bits to select physical channels.
   Set ADCAnVCRj.ADIE bit to enable the A/D conversion end interrupt.
7. Set ADCAnSGVCSPx register to specify the start pointer of virtual channel.
8. Set ADCAnSGVCEPx register to specify the end pointer of virtual channel.
9. Set ADCAnDGCTL1 register to specify the physical channel to the self-diagnostic channel.
10. Generate the start trigger of scan group to perform the A/D conversion.
11. When the conversion interrupt occurs, read the result and compare it with the expected one.
12. If the result is the expected one, the A/D conversion was performed successfully.

NOTE

- During A/D conversion, the self-diagnostic voltage level can be changed by writing to ADCAnDGCTL0.PSEL[2:0]. However, the value of ADCAnDGCTL0.PSEL[2:0] becomes effective from the next A/D conversion.
- To clear ADCAnADCR.DGON, follow the procedure below:
  1. Confirm that SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
  2. Clear ADCAnDGCTL0.PSEL[2:0].
  3. Clear ADCAnADCR.DGON.

31.5.3 Diagnosis of Open Pins

This function detects whether the analog input pin (ADCAnIm, ADCAnImS) is open due to disconnection, etc.
An internal pull-down resistor can be connected to diagnose the analog input pin.
Connect the analog input pin (ADCAnIm, ADCAnImS) with the pull-down resistor for self-diagnosis for A/D conversion of the target channels.

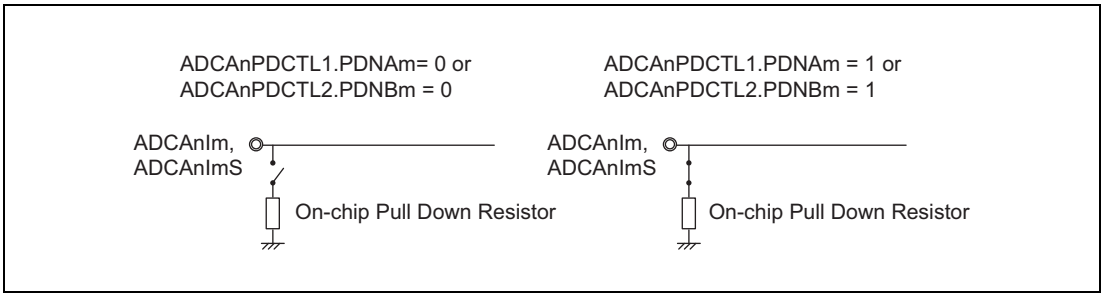


Figure 31.32 Setting of On-chip Pull Down Resistor

When there is a disconnection, the conversion result is almost 0 V and it indicates an open detection.

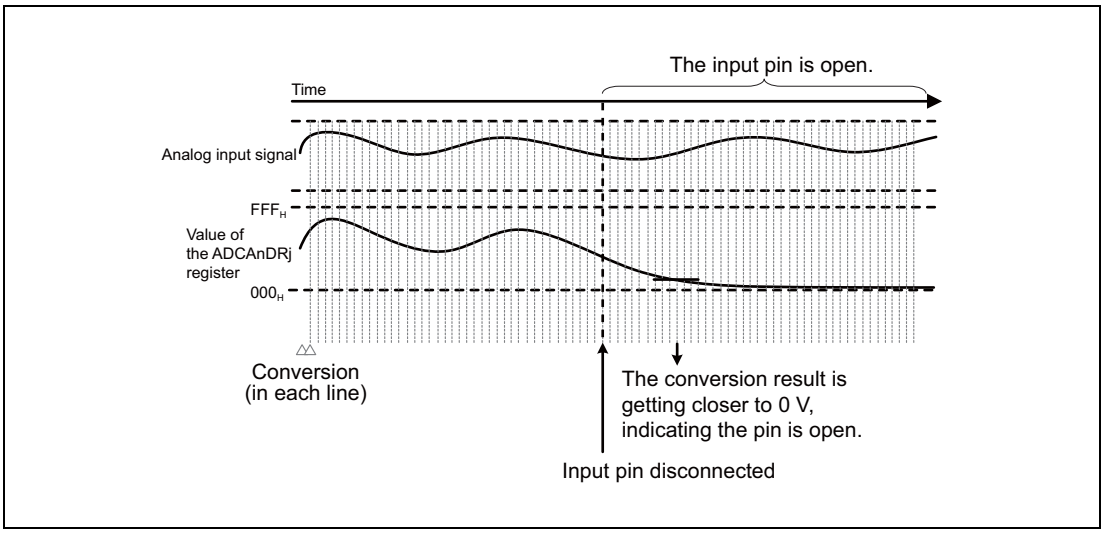


Figure 31.33 Analog Input Signal Disconnection Detection

CAUTIONS

1. **The pull-down resistors must not be connected during normal A/D conversion operation. Connected pull-down resistors may lead to a drop in the input voltage and result in erroneous A/D conversion results.**
2. **When the analog input voltage is nearly equal to the voltage level which is pulled down, a disconnection cannot be detected by this function.**

31.5.3.1 Diagnostic procedure

1. Set the ADCAnPDCTL1.PDNA or ADCAnPDCTL2.PDNB bit which correspond to analog input pins (ADCAnIm, ADCAnImS) to be diagnosed to enable the pull down resistor.
2. Generate the start trigger of scan group to perform the A/D conversion.
3. Perform the A/D conversion multiple times on the same analog input.
4. Monitor the channel's A/D conversion results and check if any result declines to 0 V.

31.5.4 Diagnosis of T&H Circuit

This function is used to diagnose proper operation of the T&H0 to T&H5 circuits for ADCA0I0 to ADCA0I5.

Virtual channel registers 33 to 35 (ADCA0VCR33 to 35) are used exclusively for comparison of the potential conversion result using the T&H circuit and that obtained without using the T&H circuit to detect a failure of the T&H circuit.

For this diagnosis, the ADCA0THACR.HLDCTE is set to 1 (ADCA0THBCR.HLDCTE = 1) and ADCA0THACR.HLDTE to 0 (ADCA0THBCR.HLDTE = 0) and the A/D conversion trigger is used as the hold start/end trigger. Connect the reference voltage signal (DIAGOUT0, DIAGOUT1, or DIAGOUT2) selected by ADCAnDGCTL0.PSEL[2:0] to the target channels for diagnosis by using the ADCAnDGCTL1 register.

31.5.4.1 Diagnostic Procedure (in case of T&H circuit ch0 diagnosis)

1. Set ADCA0ADCR.DGON = 1 to enable the self-diagnostic voltage circuit.
2. Wait for 500 ns.
3. Set ADCA0DGCTL0.PSEL[2:0] = 001_B to select $1/3AnV_{REF}$ voltage level.
4. Set ADCA0ADCR.DGON = 1 to update the voltage level.
5. Wait for 500 ns.
6. Set ADCA0DGCTL1.CDG0 = 1 to enable DIAGOUT0.
7. Set ADCA0VCR33.GCTRL[5:0] to ADCAVCR35.GCTRL[5:0] to 000000_B to select physical ch0.
8. Set ADCA0VCR33.CNVCLS and ADCA0VCR34.CNVCLS to1 to select normal conversion.
9. Set ADCA0VCR35.CNVCLS = 0 to select hold value conversion.
10. Set ADCA0THACR.SGS[1:0] = 01_B to select SG1 to “T&H group A”.
11. Set ADCA0THER.TH0E = 1 to enable T&H circuit ch0.
12. Set ADCA0THGSR.TH0GS = 0 to select T&H circuit ch0 to “T&H group A”.
13. Set ADCA0SGVCSP1.VCSP[5:0] =100001_B to select SG1 start pointer to VCR33.
14. Set ADCA0SGVCEP1.VCEP[5:0] = 100011_B to select SG1 end pointer to VCR35.
15. Set ADCA0DGCTL0.PSEL[2:0] = 011_B to select $2/3AnV_{REF}$ voltage level.
16. Set ADCA0THSMPSTCR.SMPST = 1 to execute T&H sampling.
17. Wait for 500 ns.
18. Set ADCA0SGSTCR1.SGST = 1 to execute SG1 A/D conversion.
19. Read ADCA0DIR33 to ADCA0DIR35, and check A/D conversion result to see if SG1 A/D conversion has finished.

NOTES

To clear ADCAnADCR.DGON, follow the procedure below:

1. Confirm that SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.
2. Use the ADCAnTHER register to disable the diagnosed T&Hk.
3. Clear ADCAnDGCTL0.PSEL[2:0].
4. Clear ADCAnADCR.DGON.

31.5.4.2 Diagnosis Mechanism

(1) A reference voltage “A” is applied to one of the reference voltage signals DIAGOUT0 to DIAGOUT2. The T&H circuit holds the voltage “A” and an A/D conversion is performed without using the T&H circuit.

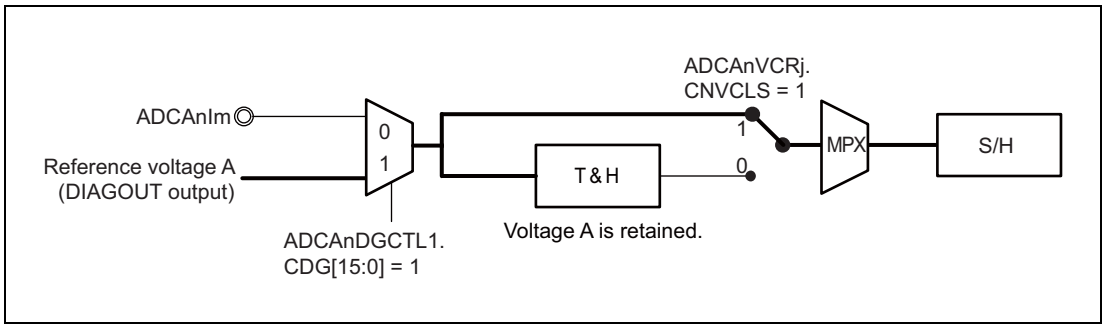


Figure 31.34 T&H Circuit Diagnostic Mechanisms (1)

(2) A reference voltage “B” is applied to one of the reference voltage signals DIAGOUT0 to DIAGOUT2. The T&H circuit still holds the voltage “A” and an A/D conversion is performed without using the T&H circuit.
Note that since the reference voltage “A” is being held, the reference voltage “B” is not held.

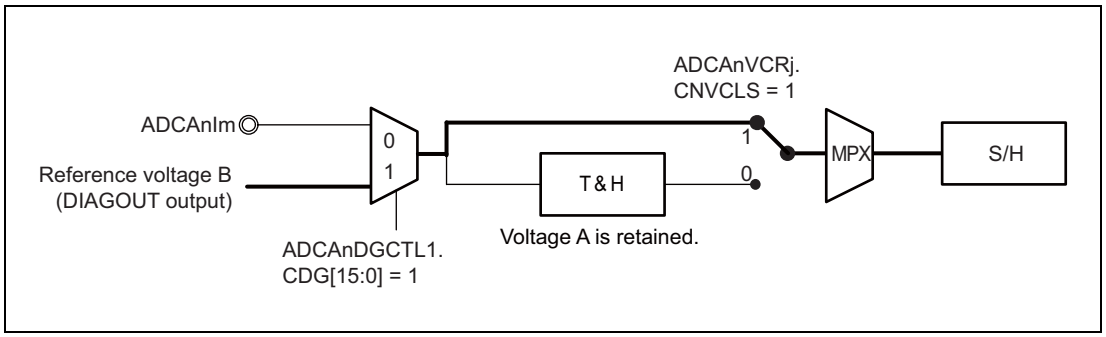


Figure 31.35 T&H Circuit Diagnostic Mechanisms (2)

(3) An A/D conversion is performed using the T&H circuit. The T&H circuit continues to hold the voltage A.

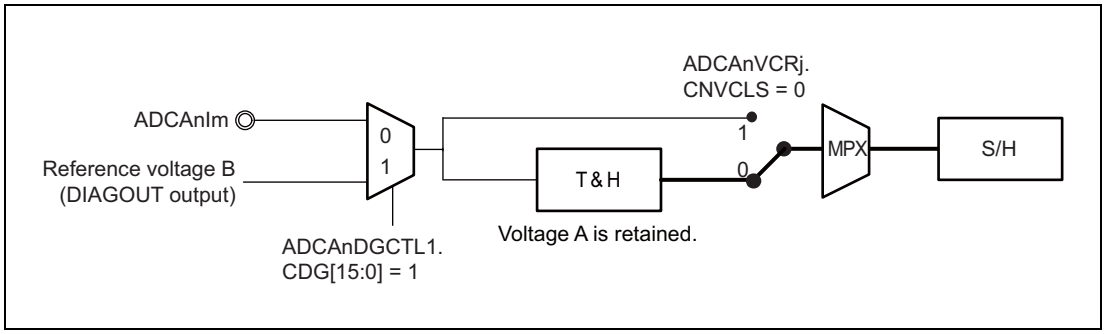


Figure 31.36 T&H Circuit Diagnostic Mechanisms (3)

(4) The diagnosis of T&H circuit is successful if the following results are obtained:
1. The first result (step 1) is voltage “A”.
2. The second result (step 2) is voltage “B”.
3. The last result (step 3) is voltage “A” again.

31.6 Definition of A/D Conversion Accuracy

A/D conversion accuracy is defined as follows:

- Resolution
  Digital output code value from the A/D converter
- Quantization error
  An error essentially contained in the A/D converter, which is assumed as 1/2 LSB (**Figure 31.37**).
- Offset error
  Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from the minimum voltage value 000_H to 001_H. However, the quantization error is not included.
- Full scale error
  Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from FFE_H to FFF_H. However, the quantization error is not included.
- DNL (Differential nonlinear error)
  Deviation between the ideal digital output code width (Vq) and the actual digital output code width (Va), which is assumed as (Va – Vq)/Vq. However, the offset error, the full scale error, and the quantization error are not included.
- INL (Integral nonlinear error)
  Deviation of the actual value from the ideal A/D conversion characteristics between the zero voltage and the full scale voltage, which is assumed as an integral of DNL from 000_H to a digital output code. However, the offset error, the full scale error, and the quantization error are not included.
- Absolute accuracy
  Deviation between the digital value and the analog input value. The offset error, the full scale error, the quantization error, DNL, and INL are included.

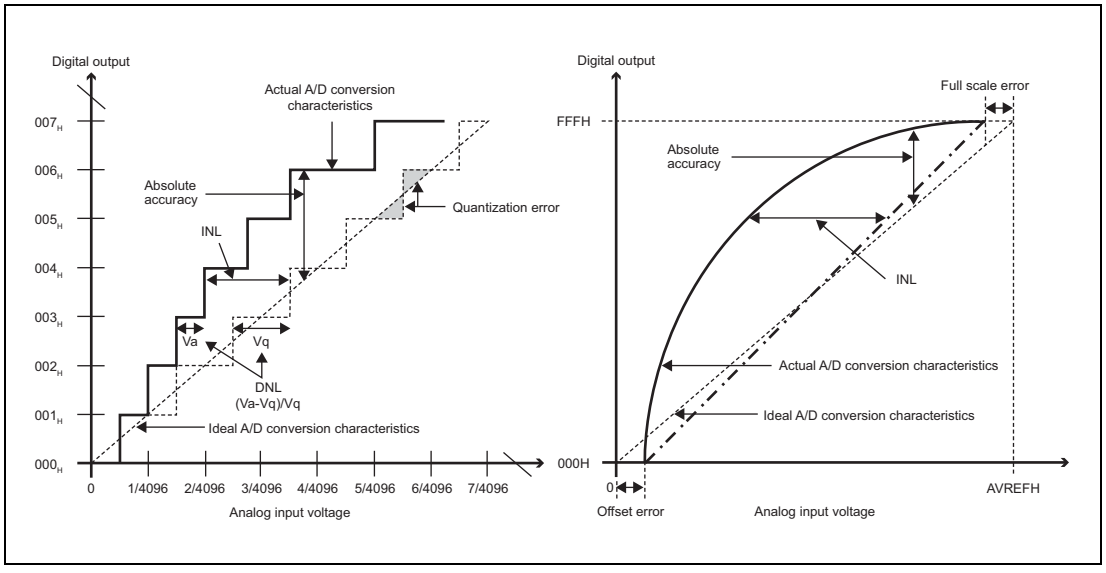


Figure 31.37 Definition of A/D Conversion Accuracy

31.7 Usage Notes

31.7.1 Range of Channel Input Voltage

CAUTION

ADCAnIm and ADCAnImS input voltages should be used within the specification range. If the channel input voltage exceeds AnVREF or falls below AnVSS, the converted value of the channel is saturated and may influence electric characteristics of other channels.

When an over-voltage is applied to a pin ADCAnIm and at the same pin the ADC self-diagnosis (diagnosis of channel multiplexer) is executed an offset voltage to the diagnosis voltage can be measured. According to the diagnosis setting, adjacent pins are affected from the pin which have over-voltaged. There are two advices for avoiding a problem. The group which have no injected current can be measure with no influence.

Case1 : Injected current is applied at one pin where the ADC self-diagnosis is executed

First ADC self-diagnosis conversion cycle:

→ only ADCA0I0 is selected (CDG0=1, CDG1-15=0) ← injected current

Second ADC self-diagnosis conversion cycle:

→ other ADCA0I1-15 are selected (CDG0=0, CDG1-15=1) ← not injected current

Case2: Injected current is applied at multiple pins where the ADC self-diagnosis is executed

First ADC self-diagnosis conversion cycle:

→ only ADCA0I0 is selected (CDG0=1, CDG1-15=0) ← injected current

Second ADC self-diagnosis conversion cycle:

→ only ADCA0I1 is selected (CDG0=0, CDG1=1, CDG2-15=0) ← injected current

Third ADC self-diagnosis conversion cycle:

→ only ADCA0I2 is selected (CDG0=0, CDG1=0, CDG2=1, CDG3-15=0) ← injected current

n ADC self-diagnosis conversion cycle:

→ other ADCA0In to ADCA0I15 are selected (CDG_{n-1}=0, CDGn to CDG15=1) ← no injected current

31.7.2 Notes on Application Design

(1) Analog input pins (ADCAnIm, ADCAnImS)

- Ensure that the input voltages on the ADCAnIm and ADCAnImS pins are within the specified ranges. We recommend using diodes with V_F of 0.3V or below to form a clamp to avoid the input of voltages at or above AnVREF and at or below AnVSS. The results of conversion for input voltages at or above AnVREF and at or below AnVSS are undefined and so are not guaranteed. Input of such voltages can also affect the results of conversion on other channels.
- Reduce noise on the analog input pins (ADCAnIm and ADCAnImS) by connecting a resistor Re between the pins and the external sources of analog input signals for conversion and capacitor Ce to the AnVSS pins.

- Avoid analog signal lines crossing digital signal lines and vice versa, since this can introduce noise and reduce performance in A/D conversion.
- We recommend avoiding the driving of large currents through input and output pins near the ADCAnIm and ADCAnImS pins and toggled signals in particular should be kept away from these pins.
- If you are using the standby functions, confirm the stop of A/D conversion. Then, set the ADCAnSGCRx.TRGMD, and ADCAnTHER.THkE bits, which are to be effective on standby, to 0.
- If you are using the LPS on ADCA0 (also when standby function is used), set the ADCA0SGCRx.TRGMD bit to 1 and the ADCA0SGTSELx.TxSEL bits to SEQTRG (LPS). For details, see **Section 13, Low-Power Sampler (LPS)**.
- Do not connect a channel to be used with the T&H function to an external analog multiplexer.
- Changes to physical and virtual channels during operation while the T&H function is in use is prohibited.
- Writing to PWM-Diag-related registers while PWM-Diag is not in use is prohibited.

(2) Power Wiring

The following methods are recommended to minimize the influence of switching noise from digital circuit on A/D converter accuracy.

- Connect markedly thick wiring patterns to the mesh pattern or connect solid patterns to the power-supply lines.
- Insert bypass capacitors between power-supply pins (EVCC, BVCC, and AnVREF) and ground pins (EVSS, BVSS, and AnVSS).
- We recommend separating the analog power supply (AnVREF) from the digital power supplies (EVCC and BVSS) and providing the voltages from a series regulator. If the analog power supply is to come from the same source as that of the digital power supplies, wire the analog and digital power supplies to an electrolytic capacitor, and provide separate wiring patterns on the board.
  We also recommend inserting a chip inductor in the input for the analog power supply.
  Furthermore, earth the analog and digital grounds to the same point on an electrolytic capacitor, and provide separate wiring patterns for the grounds on the board.
  The analog power supply also serves as the analog reference voltage for this product.

(3) Variation in A/D converted data

The effects of noise and variations in the power supply voltages lead to dispersal of the results of A/D conversion. Furthermore, noise on the analog input pins (ADCAnIm and ADCAnImS) or on the reference voltage input pins (AnVREF and AnVSS) can lead to the results of A/D conversion being incorrect.

Apply software processing to avoid ill effects on the system of fluctuations in or incorrectness of the results of A/D conversion.

Examples of software handling are described below.

- Use averaged values from several rounds of A/D conversion
- Execute A/D conversion for several time and omit extreme results
- Repeat the processing for abnormalities to check for repeated abnormalities in the case of results of A/D conversion which will cause malfunctions of the system.

There is a possibility that A/D conversion accuracy of high priority SG become worse when following both conditions are applicable.

(1) During the A/D conversion of low priority SG (e.g. SG1), conversion trigger of the high priority SG (e.g. SG3) occur.

(2) The channel T&H function of high priority SG (e.g. SG3) is enabled.

The above case is one of an example of SG combination. The SG priority is SG4 > SG3 > SG2 > SG1. The fluctuation of conversion error depends on the external circuit and devices mounted on the customer board.

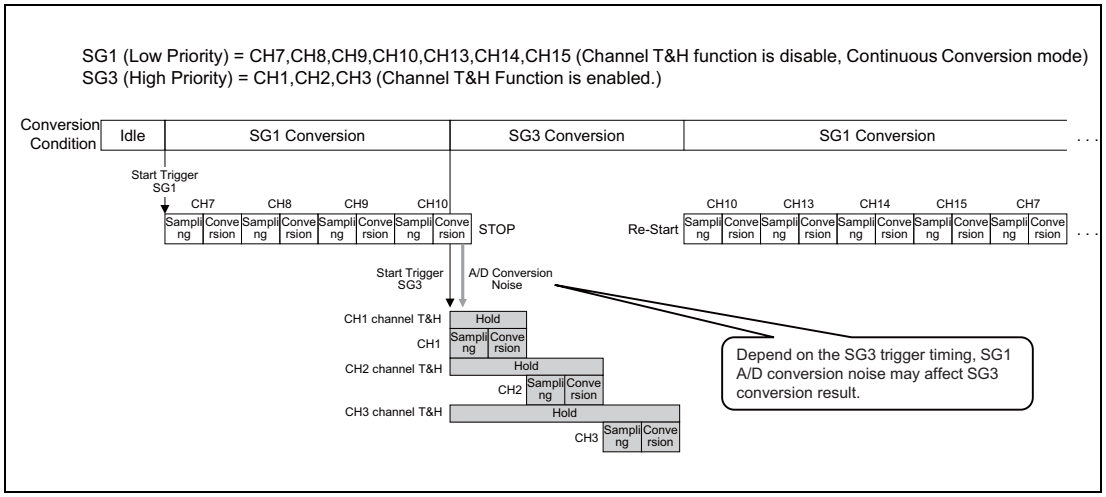


Figure 31.38 SG priority

Examples of software handling are described below.

- Low priority SG (e.g. SG1) A/D conversion have to be finished 3 ADCLK before the conversion trigger of high priority SG (e.g. SG3), if the SG3 contains the channels which channel T&H is available.
- If there is a case that high priority SG (e.g. SG3) conversion trigger occur during the conversion of low priority SG (e.g. SG1), disable the high priority SG channel T&H function.
- If the both conditions mentioned in previous page need to be used, adjust high priority SG (e.g. SG3) conversion trigger timing to synchronize with the following timing (the period shown by arrowed line in the following figure) during A/D conversion of low priority SG (e.g. SG1).

Even if trigger timing is adjusted above recommendation time, conversion error specified in Data Sheet cannot be removed.

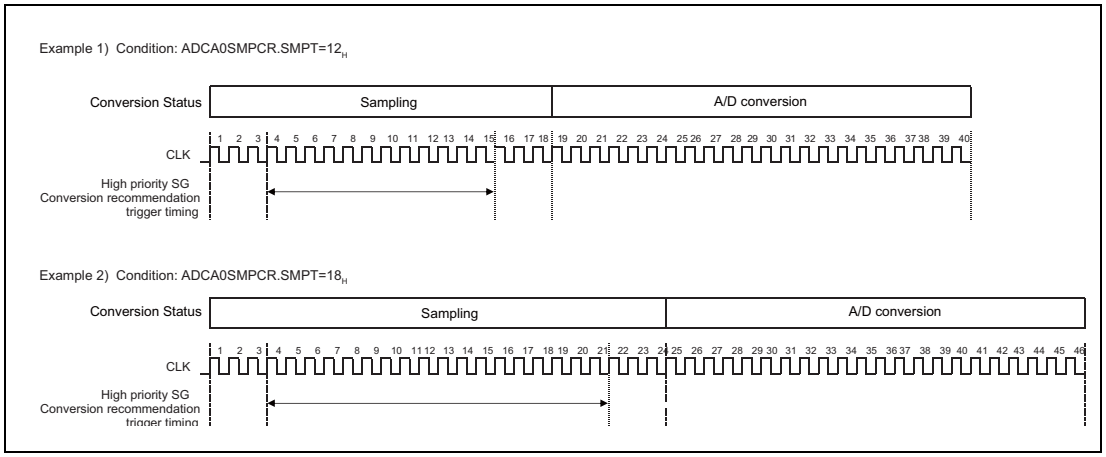


Figure 31.39 SG conversion trigger timing

When it is not possible to use the above 3 software handlings, following process is recommended.

- Doing A/D conversion several times and using average of several A/D conversion results.

- Doing continuous A/D conversion several times, remove abnormal conversion result and use only the other results.

- When abnormal A/D conversion result is detected, not to proceed abnormal operation immediately, doing one more A/D conversion before proceeding abnormal operation.

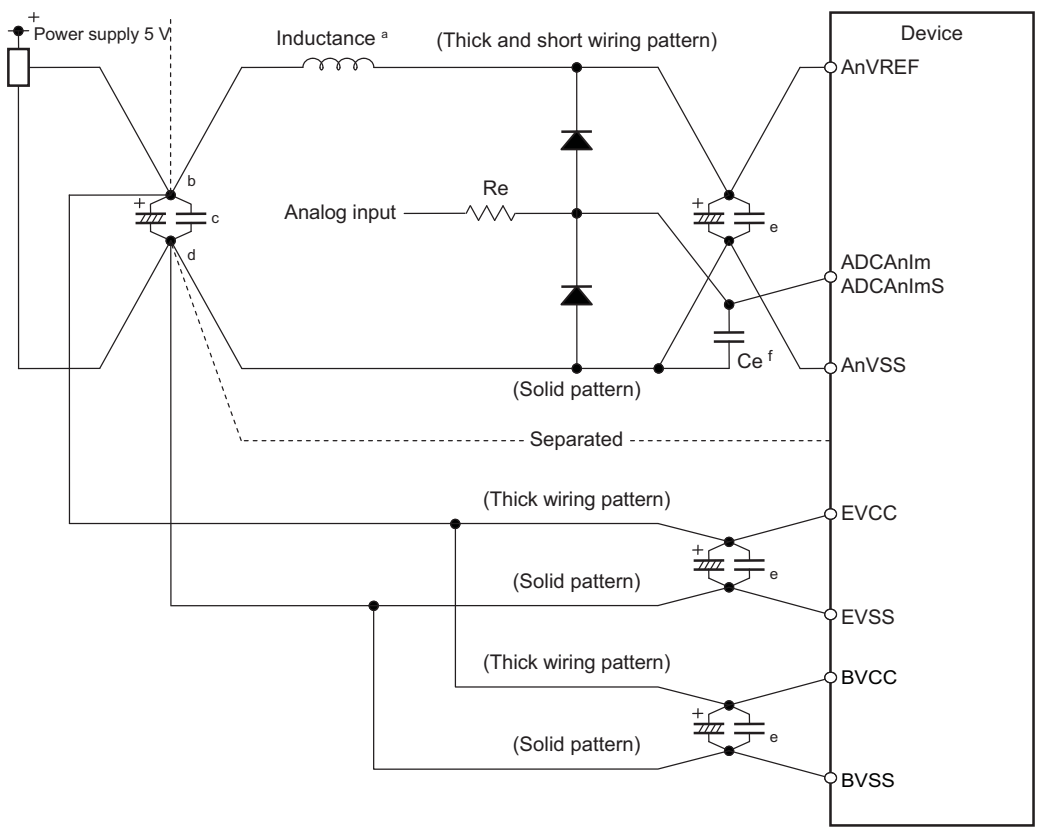
The effect of above process is depend on the external circuit and devices mounted on the customer board. Sufficient evaluation of the system is recommended.

(4) Alternative Input/Output

Analog input (ADCAnIm, ADCAnImS) pins can be used as port pins.

Do not read from input port pins or write to output port pins while an ADCAnIm or ADCAnImS pin function is selected and handling A/D conversion. Doing so may reduce the accuracy of conversion.

Fluctuations in output current from output port pins due to the effects of an external circuit connected to a port pin while A/D conversion is in progress may also reduce the accuracy of conversion. If digital pulses are applied to or digital pulses are output through a pin adjacent to a pin for which A/D conversion is in progress, the A/D converted value may not be as expected due to coupling noise. Accordingly, do not apply pulses to or output pulses from a pin adjacent to a pin for which A/D conversion is in progress.



a) Insert an inductor for the analog power input.
b) Wire the analog and digital supplies to a single point near the 5V power supply and provide separate wiring patterns on the board.
c) Place a bypass capacitor (electrolytic or laminated ceramic) near the common power supply.
d) Wire the analog and digital grounds to a single point near the power supplies, and provide separate wiring patterns for each on the board.
e) Place bypass capacitors (electrolytic or laminated ceramic) near the respective power supply pins.
f) Place capacitor Ce near the analog input pins of the device.

Figure 31.40 Example of Power Wiring

Section 32 Key Return (KR)

This section contains a generic description of the Key Return (KR) function.

The first part in this section describes the RH850/F1M specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the KR.

32.1 Features of RH850/F1M KR

32.1.1 Number of Units and Channels

This microcontroller has the following number of KR units.

Table 32.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	1		
Name	KRn (n = 0)		

The KR unit has the Key Return function of the following number of channels.

Table 32.2 KRn Unit Configurations and Channels

Unit Name (Channel Name) KRn	No. of Channels	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
KR0	8	√	√	√

Table 32.3 Indices

Index	Description
n	Throughout this section, individual KR units are identified by the index “n” (n = 0); for example, KRnKRM indicates the key return mode register.
m	Throughout this section, individual KR channels are identified by the index “m” (m = 0 to 7); for example, KRnKRMm indicates the key input enable bit of KRnKRM (key return mode register).

32.1.2 Register Base Address

KRn base address is listed in the following table.

KRn register addresses are given as offsets from the base address.

Table 32.4 Register Base Address

Base Address Name	Base Address
<KR0_base>	FFF7 8000_H

32.1.3 Clock Supply

The KRn clock supply is shown in the following table.

Table 32.5 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
KR0	PCLK	CPUCLK4
	Register access clock	CPUCLK4

32.1.4 Interrupt Requests

KRn interrupt requests are listed in the following table:

Table 32.6 Interrupt Requests

Unit Interrupt Signal	Description	Interrupt Number	DMA Trigger Number
KR0			
INTKRn	Key interrupt	90	—

32.1.5 Reset Sources

KRn reset sources are listed in the following table. KRn is initialized by these reset sources.

Table 32.7 Reset Sources

Unit Name	Reset Source
KR0	All reset sources (ISORES)

32.1.6 External Input/Output Signals

External input/output signals of KRn are listed below.

Table 32.8 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal
KR0		
KRnTPKR7 to KRnTPKR0	Key input signal	KR0I7 to KR0I0

32.2 Overview

32.2.1 Functional Overview

The Key Return function has the following features:

A key interrupt request signal (INTKRn) can be generated by inputting a falling signal, that goes from high to low, to any of the eight key input pins (KRnTPKR7 to KRnTPKR0).

32.2.2 Block Diagram

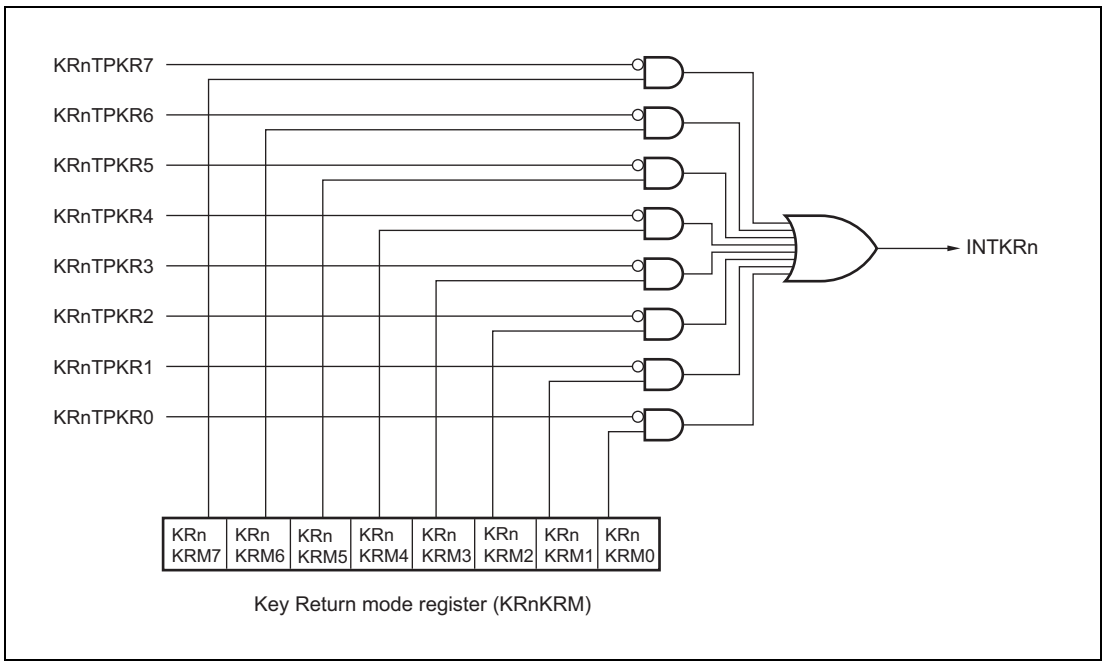


Figure 32.1 Block Diagram of the Key Return Function

32.3 Registers

32.3.1 List of Registers

KR registers are listed in the following table.

For details about <KRn_base>, see **Section 32.1.2, Register Base Address**.

Table 32.9 List of Registers

Module Name	Register Name	Symbol	Address
KRn	Key return mode register	KRnKRM	<KRn_base>

32.3.2 KRnKRM — Key Return Mode Register

This register enables/disables the key input signal detection.

Access: This register can be read or written in 8-bit or 1-bit units.

Address: <KRn_base>

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	KRnKRM7	KRnKRM6	KRnKRM5	KRnKRM4	KRnKRM3	KRnKRM2	KRnKRM1	KRnKRM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.10 KRnKRM - Key Return Mode Register Contents

Bit position	Bit name	Function
7 to 0	KRnKRMm	Enables/disables the key input signal detection. 0: Disabled 1: Enabled

32.4 Operation

32.4.1 Interrupt Request INTKRn

The interrupt request INTKRn is generated when the level of the corresponding key input pin KRnTPKRm is changed from high to low while input to the key input pin KRnTPKRm is enabled (KRnKRM.KRnKRMm = 1).

Figure 32.2 shows how the interrupt request is generated:

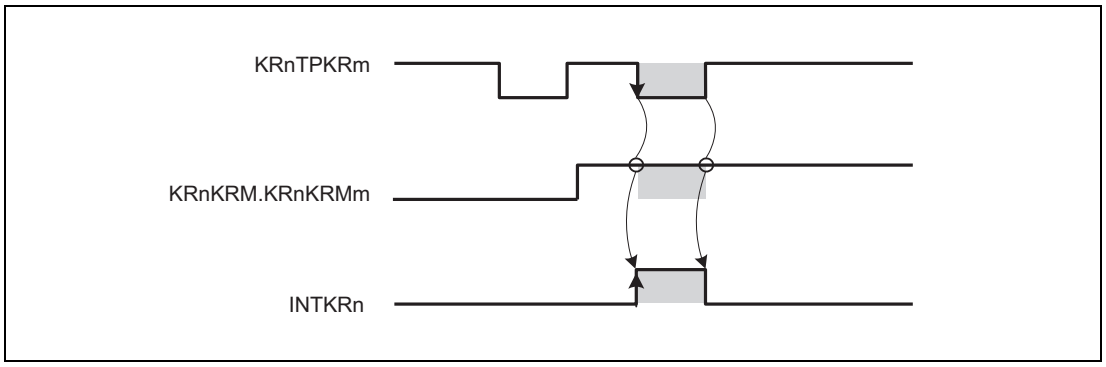


Figure 32.2 Interrupt Request Generation

CAUTIONS

1. **The change of a key input pin (KRnTPKRm) level from high to low does not trigger another INTKRn if any of the key return input pins are already low. The next INTKRn is only triggered by a key input pin level changing from high to low if all other key input pins are high.**
2. **If the key input value changes at the same time the setting of KRnKRM. KRnKRMm is changed, an unintended key interrupt request INTKRn might be generated.**
   Therefore, mask (disable) INTKRn of the interrupt controller before changing KRnKRM.KRnKRMm from 0 to 1, or from 1 to 0.

Section 33 Functional Safety

This section provides an overview of the safety mechanisms included in the RH850/F1M Series.

This microcontroller has been developed as a Safety Element out of Context (SEooC) in accordance with ISO26262.

For more information about the development process and safety mechanisms, please contact our sales office.

The following are the failure detection functions provided by this microcontroller.

33.1 Overview

ECC

Detect failures of memories and data transfer paths and correct some types of failures.

Memory Protection

Detects erroneous access to memories and peripheral circuits to protect the data in these elements against erroneous access.

Clock Monitor

Monitors the clock operation to detect an abnormal operation.

For details, see **Section 11.7, Clock Monitor A (CLMA)**.

Data CRC

Generates CRC to detect data errors.

For details, see **Section 34, Data CRC (DCRA)**.

Write-Protected Registers

The write-protected registers are protected from inadvertent write access due to erroneous program execution.

For details, see **Section 5, Write-Protected Registers**.

33.2 ECC and EDC

33.2.1 Overview

This product incorporates an ECC for the following memories. The ECC enables detection and correction of errors of the data retained in the memories. The ECC also enables detection and correction of errors produced between the ECC encoder and memories; and memories and ECC decoder.

Table 33.1 ECC Overview

Applicable Memory		Applicable Data Width [bits]	Operation upon Error Detection					Failure Insertion
			Detection/ Correction	SYSERR	Interrupt Notice	Error Status	Address Capture	
Code flash		128	SEC-DED	DED*2	SEC	Possible	Possible	Possible
Data flash		32	SEC-DED	—	DED	Possible	—	Possible
Local RAM (CPU1)		32	SEC-DED	DED*2	SEC	Possible	—	Possible
Retention RAM		32	SEC-DED	DED*2	SEC	Possible	Possible	Possible
Instruction cache (data)		64	SEC-DED*3	—	SEC	Possible	Possible	Possible
Instruction cache (tag)		32	SED-DED	—	SED*4	Possible	Possible	Possible
Peripheral RAM*1	CSIH	32	SEC-DED	—	DED	Possible	—	Possible
	RSCAN	32	SEC-DED	—	DED	Possible	—	Possible
	FlexRay	32	SEC-DED	—	DED	Possible	—	Possible

Note 1. For details of ECC for each peripheral IP, see the corresponding section below.
- RS-CAN RAM: **Section 20.11, Detection and Correction of Errors in RS-CAN RAM**
- CSIHnRAM: **Section 16.7, Detection and Correction of Errors in CSIHn RAM**
- FlexRay: **Section 21.4, Detection and Correction of Errors in FlexRay RAM**

Note 2. When the DMA makes an access, SYSERR is not generated. Confirm that no ECC 2-bit error has occurred
- in the following bits after the DMA transfer:
- CF1STERSTR_VCI.DEDF0 (Code Flash)
- LR1STERSTR_PE1.DEDFn (Local RAM CPU1)
- GR1STERSTR_VCI.DEDFn (Retention RAM)

Note 3. ECC 1-bit error correction is selectable by ICCTRL.D1EIV.

Note 4. When single error is occurred, the entry is cleared and fetching is repeated.

Applicable Data Width

This is the data width to be ECC encoded.

To write data with a smaller data width than shown, the following processing is required. ECC is also performed for a read in (1).

(1) Reading data to be ECC-encoded including data to be rewritten

(2) Replacing data to be rewritten

(3) Writing back data generated in (2)

Detection/Correction

SEC-DED: 1-bit errors can be detected and corrected, and 2-bit errors can only be detected.

SED-DED: 1-bit errors and 2-bit errors can only be detected.

SYSERR

SYSERR can be generated upon error detection.

Interrupt Notice

An interrupt can be generated upon error detection.

Error Status

The status of a detected error is retained.

Address Capture

The address of a detected error is retained.

Failure Insertion

An ECC error can be intentionally caused to enable self-diagnosis of the ECC decoder operation.

33.2.2 Address Parity

This product incorporates address EDC (parity) for the following memories. The address EDC enables detection of errors during address decoding. The EDC also enables detection of errors produced at addresses between the parity encoder and memories.

Table 33.2 Address Parity Overview

Applicable Memory	Parity Bit	Error Status	Address Capture	Failure Insertion
Code flash	1 bit	Possible	Possible	Possible

CAUTION

When the DMA makes an access, SYSERR is not generated. Confirm that no parity error has occurred in the following bits after the DMA transfer:

- **CF1STERSTR_VCI.APEF0**

33.2.3 Code Flash ECC and Address Parity

33.2.3.1 Overview

The code flash ECC is summarized in the table below.

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.
Address parity	Address parity check can be either enabled or disabled. Address parity is checked during address decoding. In the initial state, this function is enabled.
Error notification	A notification is sent when an ECC error occurs. **ECC Error:** • Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 2-bit errors is selectable. Selectable by setting the SEGCONT.ROME bit (initial value: notification is disabled) and SEGCONT.VCIE bit (initial value: notification is disabled). When a DMA transfer is executed, an error is not notified. For details, refer to **Table 33.1, ECC Overview**. • Enabling or disabling of error (INTECCSCFLI0 interrupt) notification in the case of detection of a 1-bit ECC error is selectable. This can be selected by using the FEINTFMSK.ECCSCFLI0FEIFMSK (INTECCSCFLI0 interrupt mask) bit (initial value: interrupt masked) and the CFERRINT.SEDIE (1-bit ECC error notification control) bit (initial value: notification disabled). **Parity Error:** • Enabling or disabling of error (SYSERR exception) notification in the case of detection of address parity errors is selectable. Selectable by setting the SEGCONT.ROME bit (initial value: notification is disabled) and SEGCONT.VCIE bit (initial value: notification is disabled). When a DMA transfer is executed, an error is not notified. For details, refer to **Table 33.2, Address Parity Overview**. In the initial state, error notification is disabled upon detection of a 2-bit ECC error, and error notification is disabled upon detection of a 1-bit ECC error.
Error status	The detection of ECC 2-bit errors, ECC 1-bit errors, and address parity errors can be monitored. The ECC 1-bit error status is set only when no error status has been set. The ECC 2-bit error status is set even when the ECC 1-bit error status is set. A register for clearing the error status is provided.
Address capture	When no error status has been set, the address at which the first error occurred is captured. In addition, when the retained address source is a 1-bit ECC error, the address of the 2-bit ECC error or address parity error is also captured.
Self-diagnosis	ROM data, the ECC bit, and the address parity bit can be read directly. Arbitrary data can be written to ROM data, the ECC bit, and the address parity bit.
Instruction execution suppression	Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching prevents the execution of incorrect instructions.

An ECC decoder is provided for each read port (CPU1 and interconnect) connected to the code flash interface. See **Figure 33.1**.

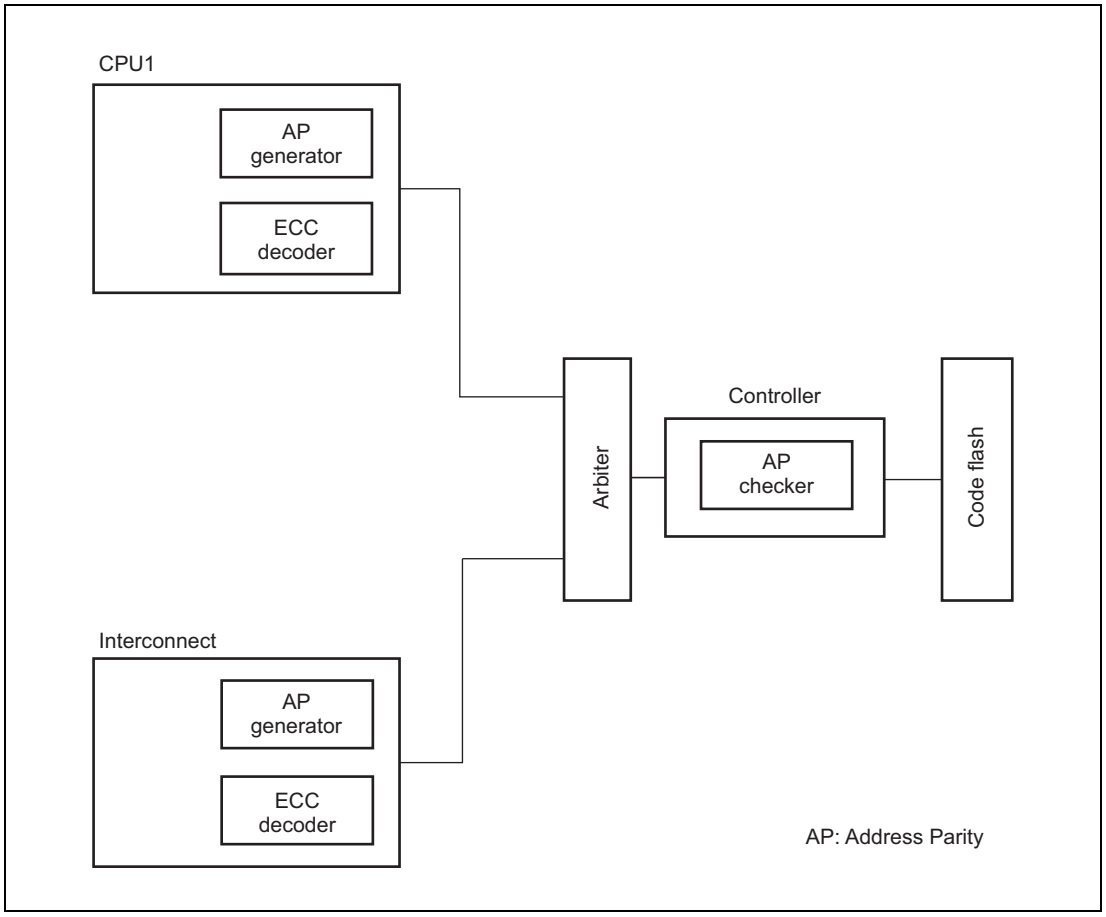


Figure 33.1 Code Flash ECC

33.2.3.2 Interrupt Requests

Interrupt requests for code flash ECC are listed below.

Table 33.3 Code flash ECC Interrupt requests (during CPU read access)

Unit interrupt signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of code flash	INTECCSCFLI0	—
—	ECC 2-bit error interrupt of code flash	SYSERR	—

33.2.3.3 List of Registers

Table 33.4 List of Registers

Register Name	Symbol*[1]	Address
Code flash address parity control register	CFAPCTL	FFC6 2000_H
Code flash ECC control register (VCI)	CFECCCTL_VCI	FFC6 2200_H
Code flash error information control register (VCI)	CFERRINT_VCI	FFC6 2204_H
Code flash status clear register (VCI)	CFSTCLR_VCI	FFC6 2208_H
Code flash error count overflow status register (VCI)	CFOVFSTR_VCI	FFC6 $220C_H$
Code flash 1st error status register (VCI)	CF1STERSTR_VCI	FFC6 2210_H
Code flash 1st error address register (VCI)	CF1STEADR0_VCI	FFC6 2250_H
Code flash ECC control register (PE1)	CFECCCTL_PE1	FFC6 2400_H
Code flash error information control register (PE1)	CFERRINT_PE1	FFC6 2404_H
Code flash status clear register (PE1)	CFSTCLR_PE1	FFC6 2408_H
Code flash error count overflow status register (PE1)	CFOVFSTR_PE1	FFC6 $240C_H$
Code flash 1st error status register (PE1)	CF1STERSTR_PE1	FFC6 2410_H
Code flash 1st error address register (PE1)	CF1STEADR0_PE1	FFC6 2450_H
Code Flash sub-test control register (VCI)	CFSTSTCTL_VCI	FFC6 2350_H
Code Flash sub-test control register (PE1)	CFSTSTCTL_PE1	FFC6 2550_H

Note 1. The registers with symbols "_VCI" and "_PE1" as suffixes are provided to the particular ECC controllers: the registers with "_VCI" are provided to the ECC controller for access from the system interconnect 1 to the Code Flash, the registers with "_PE1" are provided to the ECC controller for access from the CPU1.

33.2.3.4 Details of Registers

(1) CFAPCTL — Code Flash Address Parity Control Register

CFAPCTL enables or disables address parity check. Set the PROT1 and PROT0 bits to 01_B when writing to CFAPCTL.

Access: CFAPCTL can be read or written in 32-bit units.
CFAPCTLL can be read or written in 16-bit units.

Address: CFAPCTL: FFC6 2000_H
CFAPCTLL: FFC6 2000_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT[1:0]		—	—	—	—	—	—	—	—	—	—	—	—	APTESTA	APARIDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.5 CFAPCTL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT[1:0]	Enable or disable modification of the APARIDIS and APTESTA bits. The value written is not retained. These bits are always read as 0. Set PROT1 and PROT0 to 01 when writing to this register.
14		
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	APTESTA	Address Parity Checker Test Sets the address parity checker to test mode. When APTESTA = 1, the parity generated by the address parity generator is inverted. When writing to this bit, write 01 to PROT1 and PROT0 at the same time.
0	APARIDIS	Address Parity Check Disable Enables or disables address parity check by the address parity circuit. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: Enables address parity check. 1: Disables address parity check.

(2) CFECCCTL_VCI/PE1 — Code Flash ECC Control Register

CFECCCTL enables or disables ECC error detection and correction and 1-bit error correction. Set the PROT1 and PROT0 bits to 01_B when writing to CFECCCTL.

Access: CFECCCTL_VCI and CFECCCTL_PE1 can be read or written in 32-bit units.
CFECCCTL_VCIL and CFECCCTL_PE1L can be read or written in 16-bit units.

Address: CFECCCTL_VCI: FFC6 2200_H
CFECCCTL_VCIL: FFC6 2200_H
CFECCCTL_PE1: FFC6 2400_H
CFECCCTL_PE1L: FFC6 2400_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.6 CFECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	These bits enable or disable modification of the ECCDIS and SECDIS bits. The written data is not retained. These bits are always read as 0. Set PROT1 and PROT0 to 01 when writing to this register.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable This bit enables or disables 1-bit error correction when the ECC error detection/correction is enabled. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: 1-bit error correction is performed when a 1-bit error is detected. 1: 1-bit error correction is not performed when a 1-bit error is detected.
0	ECCDIS	ECC Disable This bit enables or disables ECC error detection/correction. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: ECC error detection/correction is enabled. 1: ECC error detection/correction is disabled.

(3) CFERRINT_VCI/PE1 — Code Flash Error Information Control Register

CFERRINT enables or disables generation of the error notification signal upon detection of an 1-bit ECC error.

Access: CFERRINT_VCI and CFERRINT_PE1 can be read or written in 32-bit units.
CFERRINT_VCIL and CFERRINT_PE1L can be read or written in 16-bit units.
CFERRINT_VCILL and CFERRINT_PE1LL can be read or written in 8-bit units.

Address: CFERRINT_VCI: FFC6 2204$_H$
CFERRINT_VCIL: FFC6 2204$_H$
CFERRINT_VCILL: FFC6 2204$_H$
CFERRINT_PE1: FFC6 2404$_H$
CFERRINT_PE1L: FFC6 2404$_H$
CFERRINT_PE1LL: FFC6 2404$_H$

Value after reset: 0000 0006$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 33.7 CFERRINT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SEDIE	ECC 1-Bit Error Notification Enable Enables or disables generation of the error notification signal (FE-level maskable interrupt request) upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(4) CFSTCLR_VCI/PE1 — Code Flash Status Clear Register

CFSTCLR clears the error flags in the error status register (CF1STERSTR), the overflow flag in the error overflow status register (CFOVFSTR), and the error address register (CF1STEADR).

Access: CFSTCLR_VCI and CFSTCLR_PE1 are the write-only registers that can be written in 32-bit units.
CFSTCLR_VCIL and CFSTCLR_PE1L are the write-only registers that can be written in 16-bit units.
CFSTCLR_VCILL and CFSTCLR_PE1LL are the write-only registers that can be written in 8-bit units.

Address: CFSTCLR_VCI: FFC6 2208_H
CFSTCLR_VCIL: FFC6 2208_H
CFSTCLR_VCILL: FFC6 2208_H
CFSTCLR_PE1: FFC6 2408_H
CFSTCLR_PE1L: FFC6 2408_H
CFSTCLR_PE1LL: FFC6 2408_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 33.8 CFSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	STCLR0	Error Status Flag Clear 0: No effect (Setting 0 does not affect the APEF0, DEDF0, and SEDF0 flags in CF1STERSTR; ERROVF0 flag in CFOVFSTR; and CF1STEADR0.) 1: Writing 1 to this bit clears the APEF0, DEDF0, and SEDF0 flags in CF1STERSTR; ERROVF0 flag in CFOVFSTR; and CF1STEADR0.

(5) CFOVFSTR_VCI/PE1 — Code Flash Error Count Overflow Status Register

CFOVFSTR monitors occurrence of error overflow. If a second error occurs after the first error (while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set.

Access: CFOVFSTR_VCI and CFOVFSTR_PE1 are read-only registers that can be read in 32-bit units.
CFOVFSTR_VCIL and CFOVFSTR_PE1L are read-only registers that can be read in 16-bit units.
CFOVFSTR_VCILL and CFOVFSTR_PE1LL are read-only registers that can be read in 8-bit units.

Address: CFOVFSTR_VCI: FFC6 220C$_H$
CFOVFSTR_VCIL: FFC6 220C$_H$
CFOVFSTR_VCILL: FFC6 220C$_H$
CFOVFSTR_PE1: FFC6 240C$_H$
CFOVFSTR_PE1L: FFC6 240C$_H$
CFOVFSTR_PE1LL: FFC6 240C$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.9 CFOVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	Error Overflow Flag ERROVF0 is set if a second error occurs while any of the error flags (APEF0, DEDF0, and SEDF0) in the error status register is set. ERROVF0 is not set when the second error occurs at the same address or source like the first error. 0: Did not occur 1: Occurred

(6) CF1STERSTR_VCI/PE1 — Code Flash 1st Error Status Register

CF1STERSTR monitors occurrence of the first error. The error status is set if an error occurs while the error flag is 0. If an ECC 2-bit error or an address parity error occurs while the ECC 1-bit error monitor flag is set, the error flag for the ECC 2-bit error or an address parity error flag is set with the error flag for the ECC 1-bit error held.

The CF1STERSTR register is cleared by an internal reset, external reset, or by setting the STCLR bit in the CFSTCLR register to 1.

Access: CF1STERSTR_VCI and CF1STERSTR_PE1 are read-only registers that can be read in 32-bit units.
CF1STERSTR_VCIL and CF1STERSTR_PE1L are read-only registers that can be read in 16-bit units.
CF1STERSTR_VCILL and CF1STERSTR_PE1LL are read-only registers that can be read in 8-bit units.

Address: CF1STERSTR_VCI: FFC6 2210$_H$
CF1STERSTR_VCIL: FFC6 2210$_H$
CF1STERSTR_VCILL: FFC6 2210$_H$
CF1STERSTR_PE1: FFC6 2410$_H$
CF1STERSTR_PE1L: FFC6 2410$_H$
CF1STERSTR_PE1LL: FFC6 2410$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	APEF0	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.10 CF1STERSTR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	APEF0	Address Parity Error Monitor Flag 0: Address parity error is not detected. 1: Address parity error is detected. Clearing condition: Set the STCLR0 bit in CFSTCLR to 1. Setting condition: Address parity error is detected when both DEDF0 and APEF0 are 0.
1	DEDF0	ECC 2-bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. Clearing condition: Set the STCLR0 bit in CFSTCLR to 1. Setting condition: ECC 2-bit error is detected when both DEDF0 and APEF0 are 0.
0	SEDF0	ECC 1-bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. Clearing condition: Set the STCLR0 bit in CFSTCLR to 1. Setting condition: ECC 1-bit error is detected when DEDF0, SEDF0 and APEF0 are 0.

(7) CF1STEADR0_VCI/PE1 — Code Flash 1st Error Address Register

CF1STEADR holds the address at which an error has occurred.

The error address is updated if an error occurs while all the error flags are 0 in CF1STERSTR. The address is updated if an ECC 2-bit error or an address parity error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error or an address parity error occurs, the address is not updated.

In addition, the EADR[24:4] bits in this register correspond to the [24:4] bits of the real address. The real address can be calculated by adding the upper address [31:25] bits, to which code flash is mapped, as a base address.

The CF1STERSTR register is cleared by an internal reset, external reset, or by setting the STCLR bit in the CFSTCLR register to 1.

Access: CF1STEADR0_VCI and CF1STEADR0_PE1 are read-only registers that can be read in 32-bit units.
CF1STEADR0_VCIL, CF1STEADR0_VCIH, CF1STEADR0_PE1L, and CF1STEADR0_PE1H are read-only registers that can be read in 16-bit units.
CF1STEADR0_VCILL, CF1STEADR0_VCILH, CF1STEADR0_VCIHL, CF1STEADR0_VCIHH, CF1STEADR0_PE1LL, CF1STEADR0_PE1LH, CF1STEADR0_PE1HL, and CF1STEADR0_PE1HH are read-only registers that can be read in 8-bit units.

Address: CF1STEADR0_VCI: FFC6 2250$_H$
CF1STEADR0_VCIL: FFC6 2250$_H$
CF1STEADR0_VCIH: FFC6 2252$_H$
CF1STEADR0_VCILL: FFC6 2250$_H$
CF1STEADR0_VCILH: FFC6 2251$_H$
CF1STEADR0_VCIHL: FFC6 2252$_H$
CF1STEADR0_VCIHH: FFC6 2253$_H$
CF1STEADR0_PE1: FFC6 2450$_H$
CF1STEADR0_PE1L: FFC6 2450$_H$
CF1STEADR0_PE1H: FFC6 2452$_H$
CF1STEADR0_PE1LL: FFC6 2450$_H$
CF1STEADR0_PE1LH: FFC6 2451$_H$
CF1STEADR0_PE1HL: FFC6 2452$_H$
CF1STEADR0_PE1HH: FFC6 2453$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	EADR[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.11 CF1STEADR0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24 to 4	EADR[24:4]	1st Error Address Monitors the address of the first error. The error address is updated if an error occurs while all the error flags are 0 in CF1STERSTR. The address is updated if an ECC 2-bit error or an address parity error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error or an address parity error occurs, the address is not updated. Clearing condition: Set the STCLR0 bit in CFSTCLR to 1.

Table 33.11 CF1STEADR0 Register Contents (2/2)

Bit Position	Bit Name	Function
3 to 0	Reserved	When read, the value after reset is returned.

(8) CFSTSTCTL_VCI/PE1 — Code Flash Sub-test Control Register

The CFSTSTCTL register is used for the ECC test (self-diagnosis). This register is dedicated to code flash. After ECC test mode is enabled by setting ECCTST to 1, the ECC bits and address parity bit can be read directly.

Set PROT1 to 0 and PROT0 to 1 when writing to this register.

Access: CFSTSTCTL_VCI and CFSTSTCTL_PE1 can be read or written in 32-bit units.
CFSTSTCTL_VCIL and CFSTSTCTL_PE1L can be read or written in 16-bit units.

Address: CFSTSTCTL_VCI: FFC6 2350$_H$
CFSTSTCTL_VCIL: FFC6 2350$_H$
CFSTSTCTL_PE1: FFC6 2550$_H$
CFSTSTCTL_PE1L: FFC6 2550$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 33.12 CFSTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCTST bit. The value written is not retained. These bits are always read as 0. Set PROT1 to 0 and PROT0 to 1 when writing to this register.
14	PROT0	
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test After ECC test mode is enabled by setting ECCTST to 1, the data in the ECC bits and the address parity bit can be read directly.

Correctly reading instructions from the code flash access port is not possible while ECC test mode is selected (ECCTST = 1). While the access port for the CPU is set to test mode (including during changes to the value of the ECCTST bit), the CPU must run a program from the local RAM or retention RAM and must not fetch instructions from the code flash memory.

The CPU has a small data buffer. If an old value remains in this buffer, the correct value cannot be read even when the ECCTST bit is switched. When switching the ECCTST bit, be sure to clear the data buffer. For how to clear the data buffer, see **Section 3.2.2, Instruction Cache and Data Buffer**.

From the code flash access port with ECC test mode selected, access must be made by reading 4 bytes aligned to 16n address. The results of reading code flash are as follows:

Table 33.13 Results of Reading Code Flash

Bit Number	Description
31 to 10	These bits are always 0.
9	Address parity bit
8 to 0	ECC bits

33.2.4 Data Flash ECC

33.2.4.1 Overview

The data flash ECC is summarized in the table below.

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit errors are detected and corrected, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. • Enabling or disabling of error notification (INTECCDEEP0 interrupt) in the case of detection of ECC 2-bit errors is selectable. Selectable by setting the FEINTFMSK.ECCDEEP0FEIFMSK (INTECCDEEP0 interrupt mask) bit (initial value: interrupt is masked) and DFERRINT.DEDIE (2-bit ECC error notification control) bit (initial value: notification is enabled). • An error is not notified when a 1-bit ECC error is detected. In the initial state, error notification is disabled upon detection of a 2-bit ECC error.
Error status	The detection of ECC 2-bit errors and ECC 1-bit errors is can be monitored. The function is set only while no error status is set. A register for clearing the error status is provided.
Self-diagnosis	ROM data and the ECC bit can be read directly. Arbitrary data can be written to ROM data and the ECC bit.

33.2.4.2 Interrupt Requests

The interrupt requests for data flash ECC are shown below.

Table 33.14 Data flash ECC interrupt requests (during read access)

Unit interrupt signal	Description	Name	DMA Trigger Number
—	ECC 2-bit error interrupt of data flash	INTECCDEEP0	—

33.2.4.3 List of Registers

Table 33.15 List of Registers

Register Name	Symbol	Address
Data Flash ECC control register	DFECCCTL	FFC6 2A00$_H$
Data Flash error status register	DFERSTR	FFC6 2A04$_H$
Data Flash error status clear register	DFERSTC	FFC6 2A08$_H$
Data Flash error notification control register	DFERRINT	FFC6 2A14$_H$
Data Flash test control register	DFTSTCTL	FFC6 2A1C$_H$

33.2.4.4 Details of Registers

(1) DFECCCTL — Data Flash ECC Control Register

DFECCCTL enables or disables ECC error detection and correction and 1-bit error correction.

Set the PROT1 and PROT0 bits to 01_B when writing to DFECCCTL.

Access: DFECCCTL can be read or written in 16-bit units.

Address: DFECCCTL: FFC6 $2A00_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SEDDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.16 DFECCCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	These bits enable or disable modification of the SEDDIS and ECCDIS bits. The written data is not retained. These bits are always read as 0. Set PROT1 and PROT0 to 01 when writing to this register.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SEDDIS	1-Bit Error Correction Disable This bit enables or disables 1-bit error correction when the ECC error detection/correction is enabled. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: 1-bit error correction is performed when a 1-bit error is detected. 1: 1-bit error correction is not performed when a 1-bit error is detected.
0	ECCDIS	ECC Disable This bit enables or disables ECC error detection/correction. When writing to thisbit, write 01 to PROT1 and PROT0 at the same time. ECC error detection/correction is enabled in the initial state. 0: ECC error detection/correction is enabled. 1: ECC error detection/correction is disabled.

(2) DFERSTR — Data Flash Error Status Register

DFERSTR monitors occurrence of errors.

The SEDF bit is set if an ECC 1-bit error is detected while ECC error detection and correction are enabled, and the DEDF bit is set if an ECC 2-bit error is detected.

Access: DFERSTR is a read-only register that can be read in 8-bit units.

Address: DFERSTR: FFC6 2A04$_H$

Value after reset: 00$_H$

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

Table 33.17 DFERSTR Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	DEDF	ECC 2-bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. Clearing condition: Set the ERRCLR bit in DFERSTC to 1. Setting condition: ECC 2-bit error is detected when both SEDF and DEDF are 0.
0	SEDF	ECC 1-bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. Clearing condition: Set the ERRCLR bit in DFERSTC to 1. Setting condition: ECC 1-bit error is detected when both SEDF and DEDF are 0.

(3) DFERSTC — Data Flash Error Status Clear Register

DFERSTC clears the error flags in the data flash error status register.

Access: DFERSTC is a write-only register that can be written in 8-bit units.

Address: DFERSTC: FFC6 2A08$_H$

Value after reset: 00$_H$

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

Table 33.18 DFERSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing, write the value after reset.
0	ERRCLR	SEDF and DEDF Flag Clear (for the DFERSTR register) Writing 1 to this bit clears the SEDF and DEDF flags of the DFERSTR register.

(4) DFERRINT — Data Flash Error Notification Control Register

DFERRINT enables or disables generation of the error notification signal upon detection of an ECC 2-bit error.

Access: DFERRINT can be read or written in 8-bit units.

Address: DFERRINT: FFC6 2A14$_H$

Value after reset: 02$_H$

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

Table 33.19 DFERRINT Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	DEDIE	ECC 2-bit error Notification Control Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

(5) DFTSTCTL — Data Flash Test Control Register

The DFTSTCTL register is used for the ECC test.

After ECC test mode is enabled by setting ECCTST to 1, the ECC bits can be read.

Set PROT1 to 0 and PROT0 to 1 when writing to the DFTSTCTL register.

Access: DFTSTCTL can be read or written in 16-bit units.

Address: FFC6 2A1C$_H$

Value after reset: 0000$_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 33.20 DFTSTCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	Enables or disables modification of the ECCTST bit. The value written is not retained. These bits are always read as 0. Set PROT1 to 0 and PROT0 to 1 when writing to the DFTSTCTL register.
14	PROT0	
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCTST	ECC Test Mode Sets ECC test mode. 0: Normal mode 1: ECC test mode

33.2.5 Local RAM (CPU1) ECC

33.2.5.1 Overview

Local RAM ECC of CPU1 is summarized in the table below.

Item	Description
ECC error detection and correction	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled, and 1-bit errors are detected and corrected, 2-bit errors detected.
Error notification	A notification is sent when an ECC error occurs. **ECC Error:** • Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 2-bit errors is selectable. Selectable by setting the SEGCONT.TCME bit (initial value: notification is disabled). • Enabling or disabling of error notification (INTECCRAM interrupt) in the case of detection of ECC 1-bit errors is selectable. Selectable by setting the FEINTFMSK.ECCRAMFEIFMSK (INTECCRAM interrupt mask) bit (initial value: interrupt is masked) and LRERRINT.SEDIE (1-bit ECC error notification control) bit (initial value: notification is disabled). In the initial state, notification of the 2-bit error is disabled and notification of the 1-bit error is disabled.
Error status	A status register is provided to indicate the statuses of 2-bit ECC error detection and 1-bit ECC error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Self-diagnosis	Arbitrary data can be written to RAM data and the ECC bit. RAM data and the ECC bit can be read directly.
Others	Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching prevents the execution of incorrect instructions.

CAUTION

When ECC error detection/correction for the local RAM is enabled, initialize the RAM using a write instruction with the 32-bit length of RAM access before the RAM is used. If the RAM before initialization is read, an FE-level maskable interrupt or SYSERR exceptional processing may be generated.

Moreover, if the RAM is not initialized with the 32-bit length (and initialized with 8- or 16-bit length), an FE-level maskable interrupt or SYSERR exceptional processing may be generated.

The local RAM in CPU1 is configured so that up to 128 bits can be read or written at the same time. On the other hand, an ECC circuit is provided for every 32-bits of data with each 32-bit data called banks 0 to 3. Data with smaller address values (LSB side) is called bank 0 and data with larger address values (MSB side) is called bank 3. The table below shows the relationship between addresses and bank numbers.

Table 33.21 Relationship between Addresses and Bank Numbers

Lower 4 bits of address (hexadecimal)	F_H to C_H	B_H to 8_H	7_H to 4_H	3_H to 0_H
Bank No.	Bank 3	Bank 2	Bank 1	Bank 0

33.2.5.2 Interrupt Requests

Local RAM ECC interrupt requests are listed below.

Table 33.22 Local RAM ECC Interrupt Requests

Unit interrupt signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	INTECCRAM	—
—	ECC 2-bit error interrupt of RAM	SYSERR*1	—

Note 1. When the DMA makes an access, SYSERR is not generated. Confirm that no ECC 2-bit error has occurred in the following bits after the DMA transfer:
- LR1STERSTR_PE1.DEDFn

33.2.5.3 List of Registers

Table 33.23 List of Registers

Register Name	Symbol	Address
Local RAM test control register (PE1)	LRTSTCTL_PE1	FFC6 5004_H
Local RAM test data read buffer 0 (PE1)	LRTDATBF0_PE1	FFC6 5008_H
Local RAM test data read buffer 1 (PE1)	LRTDATBF1_PE1	FFC6 $500C_H$
Local RAM ECC control register (PE1)	LRECCCTL_PE1	FFC6 5400_H
Local RAM error information control register (PE1)	LRERRINT_PE1	FFC6 5404_H
Local RAM status clear register (PE1)	LRSTCLR_PE1	FFC6 5408_H
Local RAM 1st error status register (PE1)	LR1STERSTR_PE1	FFC6 5410_H

33.2.5.4 Details of Registers

(1) LRTSTCTL_PE1 — Local RAM Test Control Register

This register is used for the ECC test (self-diagnosis). After ECC test mode is enabled by setting ECCTST to 1, any data can be written to the ECC bits. The DATSEL bit is used to select RAM data or the ECC bits.

Set PROT1 to 0 and PROT0 to 1 when writing to the LRTSTCTL register.

Access: LRTSTCTL_PE1 can be read or written in 32-bit units.
LRTSTCTL_PE1L can be read or written in 16-bit units.

Address: LRTSTCTL_PE1: FFC6 5004$_H$
LRTSTCTL_PE1L: FFC6 5004$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	ECCTST	DATSEL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.24 LRTSTCTL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCTST and DATSEL bits. The value written is not retained. These bits are always read as 0. Set PROT1 to 0 and PROT0 to 1 when writing to this register.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ECCTST	ECC Test After ECC test mode is enabled by setting ECCTST to 1, the ECC bits can be read directly.
0	DATSEL	Data Select This bit is valid when ECCTST is 1. This bit selects the RAM bit which can be accessed when writing. 0: RAM data is selected. 1: The ECC bits are selected.

CAUTION

When ECC test mode for the local RAM is enabled (ECCTST = 1), the local RAM should be accessed in 4-byte units.

(2) LRTDATBFn_PE1 — Local RAM Test Data Read Buffer n (n = 0, 1)

In ECC test mode (self-diagnosis), the ECC bits can be read. If the local RAM is read while ECCTST = 1 in the local RAM test control register LRTSTCTL, reading from the local RAM reads out the ECC bits, and these bits are stored in this buffer.

Access: LRTDATBFn_PE1 is a read-only register that can be read in 32-bit units.

Address: LRTDATBFn_PE1: FFC6 5008_H + n × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	LRDATABF[22:16]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	LRDATABF[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.25 LRTDATBFn Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24, 23	Reserved	These bits are read as an undefined value.
22 to 16	LRDATABF [22:16]	These bits are valid when ECCTST = 1 (selecting test mode) in the local RAM test control register. When reading from the corresponding bank in the local RAM, the ECC bits for the local RAM (bank (2n + 1)) are stored in LRTDATABF[22:16].
15 to 9	Reserved	When read, the value after reset is returned.
8, 7	Reserved	These bits are read as an undefined value.
6 to 0	LRDATABF [6:0]	These bits are valid when ECCTST = 1 (selecting test mode) in the local RAM test control register. When reading from the corresponding bank in the local RAM, the ECC bits for the local RAM (bank (2n)) are stored in LRTDATABF[6:0].

(3) LRECCCTL_PE1 — Local RAM ECC Control Register

LRECCCTL enables or disables ECC error detection and correction and 1-bit error correction. Set the PROT1 and PROT0 bits to 01_B when writing to LRECCCTL.

Access: LRECCCTL_PE1 can be read or written in 32-bit units.
LRECCCTL_PE1L can be read or written in 16-bit units.

Address: LRECCCTL_PE1: FFC6 5400_H
LRECCCTL_PE1L: FFC6 5400_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.26 LRECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set PROT1 to 0 and PROT0 to 1 when writing to this register.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: 1-bit error correction is performed when a 1-bit error is detected. 1: 1-bit error correction is not performed when a 1-bit error is detected.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(4) LRERRINT_PE1 — Local RAM Error Information Control Register

LRERRINT enables or disables generation of the error notification signal upon detection of a 1-bit ECC error.

Access: LRERRINT_PE1 can be read or written in 32-bit units.
LRERRINT_PE1L can be read or written in 16-bit units.
LRERRINT_PE1LL can be read or written in 8-bit units.

Address: LRERRINT_PE1: FFC6 5404$_H$
LRERRINT_PE1L: FFC6 5404$_H$
LRERRINT_PE1LL: FFC6 5404$_H$

Value after reset: 0000 0006$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 33.27 LRERRINT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(5) LRSTCLR_PE1 — Local RAM Status Clear Register

LRSTCLR clears the error flags in the error status register (LR1STERSTR). LRSTCLR is a write-only register and is always read as 0.

Access: LRSTCLR_PE1 is a write-only register that can be written in 32-bit units.
LRSTCLR_PE1L is a write-only register that can be written in 16-bit units.
LRSTCLR_PE1LL is a write-only register that can be written in 8-bit units.

Address: LRSTCLR_PE1: FFC6 5408$_H$
LRSTCLR_PE1L: FFC6 5408$_H$
LRSTCLR_PE1LL: FFC6 5408$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	STCLR3	STCLR2	STCLR1	STCLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 33.28 LRSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing, write the value after reset.
3	STCLR3	Error Status Flag Clear (for bank 3) Writing 1 to this bit clears the DEDF3, and SEDF3 flags in LR1STERSTR.
2	STCLR2	Error Status Flag Clear (for bank 2) Writing 1 to this bit clears the DEDF2, and SEDF2 flags in LR1STERSTR.
1	STCLR1	Error Status Flag Clear (for bank 1) Writing 1 to this bit clears the DEDF1, and SEDF1 flags in LR1STERSTR.
0	STCLR0	Error Status Flag Clear (for bank 0) Writing 1 to this bit clears the DEDF0, and SEDF0 flags in LR1STERSTR.

(6) LR1STERSTR_PE1 — Local RAM 1st Error Status Register

LR1STERSTR monitors occurrence of the first error. The error status is set if an error occurs while the error flag is 0.

For each bank, the flag status is not updated if another error occurs in the same bank while any of the error flags is set. However, if the ECC 2-bit error flag is not set, the corresponding error flag is set if an ECC 2-bit error occurs.

If more than one error occurs simultaneously, all the corresponding error flags are set. LR1STERSTR is cleared by an internal reset, an external reset, or setting 1 to the STCLR bit in LRSTCLR.

Access: LR1STERSTR_PE1 is a read-only register that can be read in 32-bit units.
LR1STERSTR_PE1L and LR1STERSTR_PE1H are the read-only registers that can be read in 16-bit units.
LR1STERSTR_PE1LL, LR1STERSTR_PE1LH, LR1STERSTR_PE1HL, and LR1STERSTR_PE1HH are the read-only registers that can be read in 8-bit units.

Address: LR1STERSTR_PE1: FFC6 5410$_H$
LR1STERSTR_PE1L: FFC6 5410$_H$
LR1STERSTR_PE1LL: FFC6 5410$_H$
LR1STERSTR_PE1LH: FFC6 5411$_H$
LR1STERSTR_PE1H: FFC6 5412$_H$
LR1STERSTR_PE1HL: FFC6 5412$_H$
LR1STERSTR_PE1HH: FFC6 5413$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	DEDF3	SEDF3	—	—	—	—	—	—	DEDF2	SEDF2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DEDF1	SEDF1	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.29 LR1STERSTR Register Contents

Bit Position	Bit Name	Function
7+8n:2+8n	Reserved	When read, the value after reset is returned.
1+8n	DEDFn	ECC 2-bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. Clearing condition: Set the STCLRn bit in LRSTCLR to 1. Setting condition: ECC 2-bit error is detected when DEDFn is 0.
0+8n	SEDFn	ECC 1-bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. Clearing condition: Set the STCLRn bit in LRSTCLR to 1. Setting condition: ECC 1-bit error is detected when both DEDFn and SEDFn are 0.

Note: n = 0 to 3, where “n” denotes a bank number.

33.2.6 Retention RAM ECC

CAUTION

Unlike the RH850/F1H, the RH850/F1M is not equipped with the global RAM, and the global RAM area is used as the retention RAM area. Therefore, use the ECC for the global RAM as the ECC for the retention RAM. Descriptions of register names and functions for the global RAM should be treated as those of the retention RAM.

33.2.6.1 Overview

The retention RAM ECC is summarized in the table below.

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled, and 1-bit errors area detected and corrected, 2-bit errors are detected.
Error notification	Notifies an ECC error upon occurrence. • Enabling or disabling of error (SYSERR exception) notification in the case of detection of ECC 2-bit errors is selectable. Selectable by setting the SEGCONT.VCIE bit (initial value: notification is disabled). • Enabling or disabling of error (INTECCRAM interrupt) notification in the case of detection of a 1-bit ECC error is selectable. This can be selected by using the FEINTFMSK.ECCRAMFEIFMSK (INTECCRAM interrupt mask) bit (initial value: interrupt masked) and the GRERRINT.SEDIE (an ECC 1-bit error notification control) bit (initial value: notification disabled). In the initial state, error notification is disabled upon detection of an ECC 2-bit error, and error notification is disabled upon detection of an ECC 1-bit error.
Error status	A status register is provided to indicate the statuses of 2-bit ECC error detection and 1-bit ECC error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Self-diagnosis	Arbitrary data can be written to RAM data and the ECC bit. RAM data and the ECC bit can be read directly.
Address capture	If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured. In addition, if the address source retained is 1-bit ECC error, it is captured when a 2-bit ECC error is detected.
Others	Generating a SYSERR exception in response to the detection of a 2-bit ECC error during instruction fetching prevents the execution of incorrect instructions.

An ECC encoder and an ECC decoder are provided for each access port (CPU1 and interconnect) connected to the retention RAM. An ECC decoder and an ECC encoder for RMW processing are also provided. See **Figure 33.2**.

- RMW processing

A bit-manipulation instruction, 2-byte write or 1-byte write is executed in the following three steps:

(1) Reading 32-bit data

(2) Generating write data by replacing the specified data (modify)

(3) Writing 32-bit data

This process is called a read-modify-write (RMW) operation in this section.

The RMW processing in retention RAM is performed in the controller.

The RMW processing performs ECC decoding for a read in (1) and ECC encoding for a write in (3).

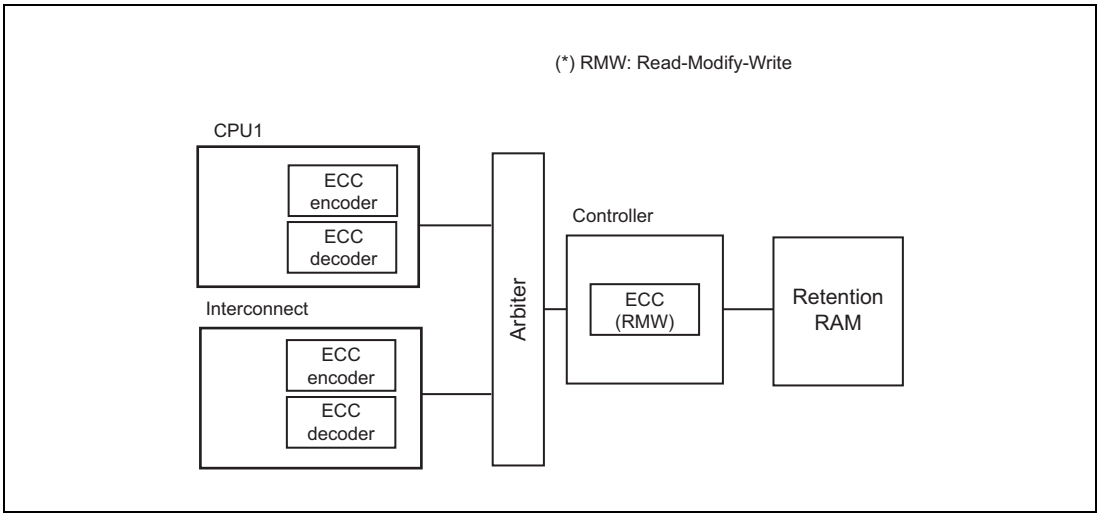


Figure 33.2 Retention RAM ECC

The retention RAM is configured so that up to 64 bits can be read or written at the same time.

An ECC circuit is provided for every 32 bits of data. That is, two ECC decoders, ECC encoders and ECC circuits (for RMW) in **Figure 33.2** are provided for each of the upper 32 bits of data and the lower 32 bits of data.

Table 33.30 Addresses and Corresponding ECC Circuits

Lower 3 bits of address	7_H to 4_H	3_H to 0_H
Corresponding ECC circuit	upper 32 bits	lower 32 bits

CAUTION

In local RAM (CPU1), ECC circuit is provided for each 32 bit data called banks 0 to 3 for 128 bits.

In Retention RAM, ECC circuit is provided for each 32 bits of data called upper 32 bits and lower 32 bits for 64 bits.

33.2.6.2 Interrupt Requests

Retention RAM ECC interrupt requests are listed below.

Table 33.31 Retention RAM ECC Interrupt Requests

Unit interrupt signal	Description	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of RAM	INTECCRAM	—
—	ECC 2-bit error interrupt of RAM	SYSERR	—

33.2.6.3 List of Registers

CAUTION

Please read “global RAM” described in Table 33.32 as “retention RAM”.

Table 33.32 List of Registers

Register Name	Symbol*1	Address
Global RAM ECC control register (GRAMC)	GRECCCTL_GRAMC	FFC6 4000_H
Global RAM test control register	GRTSTCTL	FFC6 4004_H
Global RAM test data read buffer 2	GRTDATBF2	FFC6 4010_H
Global RAM test data read buffer 3	GRTDATBF3	FFC6 4014_H
Global RAM ECC decoder input data buffer 0	GRDECINBF0	FFC6 4018_H
Global RAM ECC decoder input data buffer 1	GRDECINBF1	FFC6 $401C_H$
Global RAM ECC control register (VCI)	GRECCCTL_VCI	FFC6 4200_H
Global RAM error information control register (VCI)	GRERRINT_VCI	FFC6 4204_H
Global RAM status clear register (VCI)	GRSTCLR_VCI	FFC6 4208_H
Global RAM error count overflow status register (VCI)	GROVFSTR_VCI	FFC6 $420C_H$
Global RAM 1st error status register (VCI)	GR1STERSTR_VCI	FFC6 4210_H
Global RAM 1st error (lower 32 bits data) address register (VCI)	GR1STEADR0_VCI	FFC6 4250_H
Global RAM 1st error (upper 32 bits data) address register (VCI)	GR1STEADR1_VCI	FFC6 4254_H
Global RAM ECC control register (PE1)	GRECCCTL_PE1	FFC6 4400_H
Global RAM error information control register (PE1)	GRERRINT_PE1	FFC6 4404_H
Global RAM status clear register (PE1)	GRSTCLR_PE1	FFC6 4408_H
Global RAM error count overflow status register (PE1)	GROVFSTR_PE1	FFC6 $440C_H$
Global RAM 1st error status register (PE1)	GR1STERSTR_PE1	FFC6 4410_H
Global RAM 1st error (lower 32 bits data) address register (PE1)	GR1STEADR0_PE1	FFC6 4450_H
Global RAM 1st error (upper 32 bits data) address register (PE1)	GR1STEADR1_PE1	FFC6 4454_H

Note 1. The registers with Symbol “_GRAMC”, “_VCI”, and “_PE1” as suffixes are provided to the particular ECC controllers: the registers with “_GRAMC” are provided to the ECC controller to perform read-accesses for read-modify-write processing when 8-bit or 16-bit data is written to retention RAM, the registers with “_VCI” are provided to the ECC controller for access from the system interconnect 1 to retention RAM, and the registers with “_PE1” are provided to the ECC controller for access from the CPU1 to retention RAM.

33.2.6.4 Details of Registers

CAUTION

Please read “global RAM” described in this section as “retention RAM”.

(1) GRECCCTL_GRAMC — Global RAM ECC Control Register

GRECCCTL is the ECC control register shared by Global RAMs. GRECCCTL specifies ECC processing for Read-Modify-Write (RMW) processing. Set the PROT1 and PROT0 bits to 01_B when writing to GRECCCTL_GRAMC.

Access: GRECCCTL_GRAMC can be read or written in 32-bit units.
GRECCCTL_GRAMCL can be read or written in 16-bit units.

Address: GRECCCTL_GRAMC: FFC6 4000_H
GRECCCTL_GRAMCL: FFC6 4000_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.33 GRECCCTL_GRAMC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables updating ECCDIS and SECDIS. The value written is not retained. These bits are always read as 0. Set PROT1 to 0 and PROT0 to 1 when writing to this register.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable for RMW Enables or disables 1-bit error correction when ECC error detection and correction are enabled. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: 1-bit error correction is performed when a 1-bit error is detected. 1: 1-bit error correction is not performed when a 1-bit error is detected.
0	ECCDIS	ECC Disable for RMW Enables or disables ECC error detection and correction. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. In the initial state, ECC error detection and correction are enabled. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction. **CAUTION** The encoding function is enabled even though the error detection and correction are disabled.

(2) GRTSTCTL — Global RAM Test Control Register

This register is used for the ECC test (self-diagnosis). After ECC test mode is enabled by setting ECCTST to 1, any data can be written to the ECC bits. The DATSEL0 or DATSEL1 bit is used to select RAM data or the ECC bits. Also, input and output by the ECC decoder in the global RAM controller can be controlled for testing (self-diagnosis).

Set PROT1 to 0 and PROT0 to 1 when writing to this register.

Access: GRTSTCTL can be read or written in 32-bit units.
GRTSTCTLL can be read or written in 16-bit units.

Address: GRTSTCTL: FFC6 4004$_H$
GRTSTCTLL: FFC6 4004$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	ECCTST	DECINEN	DATSEL[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.34 GRTSTCTL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCTST, DECINEN and DATSEL [1:0] bits. The value written is not retained. These bits are always read as 0. Set PROT1 to 0 and PROT0 to 1 when writing to this register.
14	PROT0	
13 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	ECCTST	ECC Test After ECC test mode is enabled by setting ECCTST to 1, the ECC bits can be read directly.
2	DECINEN	GRAMC ECC Decoder Error Injection Enable This bit is valid when ECCTST is 1. This bit enables input of the value in the ECC decoder input buffer to the ECC decoder for use in updating of data. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: The value in the ECC decoder input buffer is not selected. 1: The value in the ECC decoder input buffer is selected.

Table 33.34 GRTSTCTL Register Contents (2/2)

Bit Position	Bit Name	Function
1 to 0	DATSEL[1:0]	Read Buffer Storage Data Select 0 and 1 This bit is valid when ECCTST is 1. This bit selects the value to be stored in the read buffer GRTDATBFn, the value to be written to each field of RAM, and the value to be input to the ECC decoder for use in updating of data at the time of RMW access. When writing to this bit, write 01 to PROT1 and PROT0 at the same time.

DATSEL1	DATSEL0	Operation
0	0	• GRTDATBFn: When reading involves an RMW cycle, the ECC bits are stored. • Global RAM: When writing involves an RMW cycle, the data area is updated according to the size of write access and the address. The ECC bits are not updated.
0	1	• GRTDATBFn: When reading involves an RMW cycle, the ECC bits are stored. • Global RAM: When writing involves an RMW cycle, only the ECC bits are updated. The data area is not updated.
1	0	• GRTDATBFn: When access is RMW, this register stores the ECC decoding results for reading data during RMW access. The value is not updated for non-RMW access. • Global RAM: Operates in the same way as in normal operating mode (i.e. when ECCTST = 0).
1	1	• GRTDATBFn: When access is RMW, this register stores the ECC decoding results for updating data during RMW access. The value is not updated for non-RMW access. • Global RAM: Operates in the same way as in normal operating mode (i.e. when ECCTST = 0). • When and only when these settings are the case, the value of the GRDECINBF0 or 1 register is input to the ECC decoder for use in updating of data at the time of RMW access instead of the write data being sent from the CPU etc.

In any case, the result read to the CPU and DMAC is the same value as would normally be read out.

(3) GRTDATBFn — Global RAM Test Data Read Buffer n (n = 2, 3)

In test mode (ECCTST = 1), data in the RAM, the ECC bits, and ECC decoder output can all be read. When the RAM is read, the value selected by the DATSEL1 or DATSEL0 bit of the global RAM test control register is stored in this buffer.

Access: GRTDATBFn is a read-only register that can be read in 32-bit units.

Address: GRTDATBFn: FFC6 4008_H + n × 4_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GRTDATBF[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GRTDATBF[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.35 GRTDATBFn Register Contents

Bit Position	Bit Name	Function
31 to 0	GRTDATBF [31:0]	These bits are valid when while ECCTST = 1 (selecting test mode) in the global RAM test control register.

DATSEL1	DATSEL0	GRTDATBF[31:0]
0	0	When reading from the RAM, the ECC bits are stored in GRTDATBF[6:0].An undefined value is GRTDATBF[7]. 0 is stored in GRTDATBF[31:8].
0	1	
1	0	When access is RMW, the output data from the ECC decoder for reading (after updating) are stored in GRTDATBF[31:0].
1	1	When access is RMW, the output data from the ECC decoder for use in updating of data (after updating) are stored in GRTDATBF[31:0].

Note: n = 2, 3, where “n = 2” denotes the lower 32 bits and “n = 3” denotes the upper 32 bits.

(4) GRDECINBF0 — Global RAM ECC Decoder Input Data Buffer 0

This register (GRDECINBF0) holds input data for the ECC decoder to use in updating the data at the time of RMW access. The data of this register is input to the ECC decoder as 32-bit RAM data.

Access: GRDECINBF0 can be read or written in 32-bit units.

Address: GRDECINBF0: FFC6 4018$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GRDECINBF0[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GRDECINBF0[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.36 GRDECINBF0 Register Contents

Bit Position	Bit Name	Function
31 to 0	GRDECINBF0 [31:0]	These bits are valid when ECCTST = 1 (selecting test mode) in the global RAM test control register. When DECINEN = 1, the value of this register is input to the ECC decoder as data for use in updating in response to the execution of an RMW instruction. The value is treated as if it were 32 bits of data from RAM. This register provides the values for both the 32 higher-order bits and the 32 lower-order bits.

(5) GRDECINBF1 — Global RAM ECC Decoder Input Data Buffer 1

This register (GRDECINBF1) holds input data for the ECC decoder to use in updating the data at the time of RMW access. The data of this register is input to the ECC decoder as 7-bit ECC data.

Access: GRDECINBF1 can be read or written in 32-bit units.

Address: GRDECINBF1: FFC6 401C$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	GRDECINBF1[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.37 GRDECINBF1 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6 to 0	GRDECINBF1 [6:0]	These bits are valid when ECCTST = 1 (selecting test mode) in the global RAM test control register. When DECINEN = 1, the value of this register is input to the ECC decoder as data for use in updating in response to the execution of an RMW instruction. The value is treated as if it were 7 bits of data from the ECC. This register provides the values for both the 32 higher-order bits and the 32 lower-order bits.

(6) GRECCCTL_VCI/PE1 — Global RAM ECC Control Register

GRECCCTL enables or disables ECC error detection and correction and 1-bit error correction. Set the PROT1 and PROT0 bits to 01_B when writing to GRECCCTL.

The setting of this register is used for accesses through the respective access port.

Access: GRECCCTL_VCI and GRECCCTL_PE1 can be read or written in 32-bit units.
GRECCCTL_VCIL and GRECCCTL_PE1L can be read or written in 16-bit units.

Address: GRECCCTL_VCI: FFC6 4200_H
GRECCCTL_VCIL: FFC6 4200_H
GRECCCTL_PE1: FFC6 4400_H
GRECCCTL_PE1L: FFC6 4400_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.38 GRECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set PROT1 to 0 and PROT0 to 1 when writing to this register.
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: 1-bit error correction is performed when a 1-bit error is detected. 1: 1-bit error correction is not performed when a 1-bit error is detected.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

(7) GRERRINT_VCI/PE1 — Global RAM Error Information Control Register

GRERRINT enables or disables generation of the error notification signal upon detection of an ECC 1-bit error.

The setting of this register is used for accesses through the respective access port.

Access: GRERRINT_VCI and GRERRINT_PE1 can be read or written in 32-bit units.
GRERRINT_VCIL and GRERRINT_PE1L can be read or written in 16-bit units.
GRERRINT_VCILL and GRERRINT_PE1LL can be read or written in 8-bit units.

Address: GRERRINT_VCI: FFC6 4204$_H$
GRERRINT_VCIL: FFC6 4204$_H$
GRERRINT_VCILL: FFC6 4204$_H$
GRERRINT_PE1: FFC6 4404$_H$
GRERRINT_PE1L: FFC6 4404$_H$
GRERRINT_PE1LL: FFC6 4404$_H$

Value after reset: 0000 0006$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 33.39 GRERRINT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(8) GRSTCLR_VCI/PE1 — Global RAM Status Clear Register

GRSTCLR clears the error flags in the error status register (GR1STERSTR), the overflow flag in the error count overflow status register (GROVFSTR), and the error address register (GR1STEADR). GRSTCLR is a write-only register and is always read as 0.

The setting of this register is used for accesses through the respective access port.

Access: GRSTCLR_VCI and GRSTCLR_PE1 are the write-only registers that can be written in 32-bit units.
GRSTCLR_VCIL and GRSTCLR_PE1L are the write-only registers that can be written in 16-bit units.
GRSTCLR_VCILL and GRSTCLR_PE1LL are the write-only registers that can be written in 8-bit units.

Address: GRSTCLR_VCI: FFC6 4208_H
GRSTCLR_VCIL: FFC6 4208_H
GRSTCLR_VCILL: FFC6 4208_H
GRSTCLR_PE1: FFC6 4408_H
GRSTCLR_PE1L: FFC6 4408_H
GRSTCLR_PE1LL: FFC6 4408_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR1	STCLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 33.40 GRSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	STCLR1	Error Status Flag Clear (for the 32 higher-order bit data) Writing 1 to this bit clears the EXDEDF1, EXSEDF1, DEDF1, and SEDF1 flags in GR1STERSTR; ERROVF1 flag in GROVFSTR; and GR1STEADR1.
0	STCLR0	Error Status Flag Clear (for the 32 lower-order bit data) Writing 1 to this bit clears the EXDEDF0, EXSEDF0, DEDF0, and SEDF0 flags in GR1STERSTR; ERROVF0 flag in GROVFSTR; and GR1STEADR0.

(9) GROVFSTR_VCI/PE1 — Global RAM Error Count Overflow Status Register

GROVFSTR monitors occurrence of error overflow. If a second error occurs after the first error (while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set. ERROVF is cleared by an internal reset, an external reset, or setting the STCLR bit to 1 in GRSTCLR.

The setting of this register is used for accesses through the respective access port.

Access: GROVFSTR_VCI and GROVFSTR_PE1 are the read-only registers that can be read in 32-bit units.
GROVFSTR_VCIL and GROVFSTR_PE1L are the read-only registers that can be read in 16-bit units.
GROVFSTR_VCILL and GROVFSTR_PE1LL are the read-only registers that can be read in 8-bit units.

Address: GROVFSTR_VCI: FFC6 420C$_H$
GROVFSTR_VCIL: FFC6 420C$_H$
GROVFSTR_VCILL: FFC6 420C$_H$
GROVFSTR_PE1: FFC6 440C$_H$
GROVFSTR_PE1L: FFC6 440C$_H$
GROVFSTR_PE1LL: FFC6 440C$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF1	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.41 GROVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	ERROVF1	Error Overflow Flag (for the 32 higher-order bit data) ERROVF1 is set if a second error occurs while any of the error flags (EXDEDF1, EXSEDF1, DEDF1, and SEDF1) in the error status registers set. ERROVF1 is not set when the second error occurs at the same address or source like the first error.
0	ERROVF0	Error Overflow Flag (for the 32 lower-order bit data) ERROVF0 is set if a second error occurs while any of the error flags (EXDEDF0, EXSEDF0, DEDF0, and SEDF0) in the error status register is set. ERROVF0 is not set when the second error occurs at the same address or source like the first error.

(10) GR1STERSTR_VCI/PE1 — Global RAM 1st Error Status Register

GR1STERSTR monitors occurrence of the first error. The error status is set if an error occurs while the error flag is 0. The upper 32 bits and lower 32 bits operate independently in terms of error detection and status updating. The flag status is not updated if another error occurs while any of the error flags is set in the upper 32 bits and lower 32 bits, respectively. However, the corresponding error flag is set if an ECC 2-bit error occurs only when either (or both) the EXSEDFn or SEDFn bit is set.

If more than one error occurs simultaneously, all the corresponding error flags are set. GR1STERSTR is cleared by an internal reset, an external reset, or setting the STCLR bit to 1 in GRSTCLR.

The setting of this register is used for accesses through the respective access port.

Access: GR1STERSTR_VCI and GR1STERSTR_PE1 are the read-only registers that can be read in 32-bit units.
GR1STERSTR_VCIL and GR1STERSTR_PE1L are the read-only registers that can be read in 16-bit units.
GR1STERSTR_VCILL, GR1STERSTR_VCILH, GR1STERSTR_PE1LL, and GR1STERSTR_PE1LH are the read-only registers that can be read in 8-bit units.

Address: GR1STERSTR_VCI: FFC6 4210$_H$
GR1STERSTR_VCIL: FFC6 4210$_H$
GR1STERSTR_VCILL: FFC6 4210$_H$
GR1STERSTR_VCILH: FFC6 4211$_H$
GR1STERSTR_PE1: FFC6 4410$_H$
GR1STERSTR_PE1L: FFC6 4410$_H$
GR1STERSTR_PE1LL: FFC6 4410$_H$
GR1STERSTR_PE1LH: FFC6 4411$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	EXDEDF1	EXSEDF1	—	—	DEDF1	SEDF1	—	—	EXDEDF0	EXSEDF0	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.42 GR1STERSTR Register Contents (1/2)

Bit Position	Bit Name	Function
7+8×n 6+8×n	Reserved	When read, the value after reset is returned.
5+8×n	EXDEDFn	ECC 2-bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. Clearing condition: Set the STCLRn bit in GRSTCLR to 1. Setting condition: ECC 2-bit error is detected when both EXDEDFn and DEDFn are 0 (during RMW processing for GRAM).
4+8×n	EXSEDFn	ECC 1-bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. Clearing condition: Set the STCLRn bit in GRSTCLR to 1. Setting condition: ECC 1-bit error is detected when EXDEDFn, EXSEDFn, DEDFn, and SEDFn are 0 (during RMW processing for GRAM).
3+8×n 2+8×n	Reserved	When read, the value after reset is returned.

Table 33.42 GR1STERSTR Register Contents (2/2)

Bit Position	Bit Name	Function
1+8×n	DEDFn	ECC 2-bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. Clearing condition: Set the STCLRn bit in GRSTCLR to 1. Setting condition: ECC 2-bit error is detected when both EXDEDFn and DEDFn are 0.
0+8×n	SEDFn	ECC 1-bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. Clearing condition: Set the STCLRn bit in GRSTCLR to 1. Setting condition: ECC 1-bit error is detected when EXDEDFn, EXSEDFn, DEDFn, and SEDFn are 0.

Note: n = 0, 1, where “n = 0” denotes the 32 lower-order bits and “n = 1” denotes the 32 upper-order bits.

(11) GR1STEADRn_VCI/PE1 — Global RAM 1st Error Address Register n (n = 0, 1)

GR1STEADRn holds the address at which an error has occurred.

The error address is updated if an error occurs while all the error flags are 0 in GR1STERSTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error occurs, the address is not updated.

In addition, the EADR[20:0] bits in this register correspond to the [24:4] bits of the real address. The real address can be calculated by adding the upper address [31:21] bits, to which global RAM is mapped, as a base address.

GR1STEADR is cleared by an internal reset, an external reset, or setting the STCLR bit to 1 in GRSTCLR. During accesses to the lower 32-bit data, the address is stored in GR1STEADR0. During accesses to the upper 32-bit data, the address is stored in GR1STEADR1.

The setting of this register is used for accesses through the respective access port.

Access: GR1STEADRn_VCI and GR1STEADRn_PE1 are the read-only registers that can be read in 32-bit units.
GR1STEADRn_VCIL, GR1STEADRn_VCIH, GR1STEADRn_PE1L, and GR1STEADRn_PE1H are the read-only registers that can be read in 16-bit units.
GR1STEADRn_VCILL, GR1STEADRn_VCILH, GR1STEADRn_VCIHL, GR1STEADRn_PE1LL, GR1STEADRn_PE1LH, and GR1STEADRn_PE1HL are the read-only registers that can be read in 8-bit units.

Address: GR1STEADRn_VCI: FFC6 4250_H + n × 4_H
GR1STEADRn_VCIL: FFC6 4250_H + n × 4_H
GR1STEADRn_VCILL: FFC6 4250_H + n × 4_H
GR1STEADRn_VCILH: FFC6 4251_H + n × 4_H
GR1STEADRn_VCIH: FFC6 4252_H + n × 4_H
GR1STEADRn_VCIHL: FFC6 4252_H + n × 4_H
GR1STEADRn_PE1: FFC6 4450_H + n × 4_H
GR1STEADRn_PE1L: FFC6 4450_H + n × 4_H
GR1STEADRn_PE1LL: FFC6 4450_H + n × 4_H
GR1STEADRn_PE1LH: FFC6 4451_H + n × 4_H
GR1STEADRn_PE1H: FFC6 4452H + n × 4_H
GR1STEADRn_PE1HL: FFC6 4452H + n × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	EADR[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.43 GR1STEADRn Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned.
20 to 0	EADR[20:0]	1st Error Address Monitors the address of the first error. The error address is updated if an error occurs while all the error flags are 0 in GR1STERSTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

33.2.7 Instruction Cache ECC

33.2.7.1 Overview

The instruction cache ECC is summarized in the table below.

Item	Description
ECC error detection and correction	ECC error detection and correction can be enabled or disabled. When enabled, either of the following settings can be selected. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function enabled, and 1-bit errors are detected and corrected*, 2-bit errors are detected.
Error notification	A notification is sent when an ECC error occurs. • Enabling or disabling of error (INTECCRAM interrupt) notification in the case of detection of a 1-bit ECC error is selectable. This can be selected by using the FEINTFMSK.ECCRAMFEIFMSK (INTECCRAM interrupt mask) bit (initial value: interrupt masked) and the IDERRINT_PE1(ITERRINT_PE1).SEDIE (ECC 1-bit error notification control) bit (initial value: notification disabled). In the initial state, notification of the 2-bit ECC error is disabled, and notification of the ECC 1-bit error is disabled.
Error status	A status register is provided to indicate the statuses of 2-bit ECC error detection and 1-bit ECC error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Self-diagnosis	Arbitrary data can be written to the RAM data and the ECC bit by using a cache instruction. Similarly, RAM data and the ECC bit can be read directly. Since this instruction passes along the same encoding/decoding path as the general cache fill or instruction fetch, an error can be injected and it can also be verified.
Address capture	If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured. The address is captured when a 2-bit or 1-bit ECC error is detected.

Note: The ECC function for Instruction cache tag does not have the ECC 1-bit error correction function.

33.2.7.2 List of Registers

Table 33.44 List of Registers

Register Name	Symbol	Address
Instruction cache data RAM ECC control register (PE1)	IDECCCTL_PE1	FFC6 0400$_H$
Instruction cache data RAM error information control register (PE1)	IDERRINT_PE1	FFC6 0404$_H$
Instruction cache data RAM error status clear register (PE1)	IDSTCLR_PE1	FFC6 0408$_H$
Instruction cache data RAM error count overflow status register (PE1)	IDOVFSTR_PE1	FFC6 040C$_H$
Instruction cache data RAM 1st error status register (PE1)	ID1STERSTR_PE1	FFC6 0410$_H$
Instruction cache data RAM (Bank0) 1st error address register (PE1)	ID1STEADR0_PE1	FFC6 0450$_H$
Instruction cache data RAM (Bank1) 1st error address register (PE1)	ID1STEADR1_PE1	FFC6 0454$_H$
Instruction cache tag RAM ECC control register (PE1)	ITECCCTL_PE1	FFC6 1400$_H$
Instruction cache tag RAM error information control register (PE1)	ITERRINT_PE1	FFC6 1404$_H$
Instruction cache tag RAM error status clear register (PE1)	ITSTCLR_PE1	FFC6 1408$_H$
Instruction cache tag RAM error count overflow status register (PE1)	ITOVFSTR_PE1	FFC6 140C$_H$
Instruction cache tag RAM 1st error status register (PE1)	IT1STERSTR_PE1	FFC6 1410$_H$
Instruction cache tag RAM 1st error address register (PE1)	IT1STEADR0_PE1	FFC6 1450$_H$

33.2.7.3 Details of Registers

(1) IDECCCTL_PE1 — Instruction Cache Data RAM ECC Control Register

IDECCCTL enables or disables ECC error detection and correction for cache data RAM. Set the PROT1 and PROT0 bits to 01_B when writing to IDECCCTL.

Access: IDECCCTL_PE1 can be read or written in 32-bit units.
IDECCCTL_PE1L can be read or written in 16-bit units.

Address: IDECCCTL_PE1: FFC6 0400_H
IDECCCTL_PE1L: FFC6 0400_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R\W

Table 33.45 IDECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set PROT1 to 0 and PROT0 to 1 when writing to this register.
14	PROT0	
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

Note: ECC 1-bit error correction is selectable by ICCTRL.D1EIV.

(2) IDERRINT_PE1 — Instruction Cache Data RAM Error Information Control Register

IDERRINT enables or disables generation of the error notification signal upon detection of a 1-bit ECC error in cache data RAM.

Access: IDERRINT_PE1 can be read or written in 32-bit units.
IDERRINT_PE1L can be read or written in 16-bit units.
IDERRINT_PE1LL can be read or written in 8-bit units.

Address: IDERRINT_PE1: FFC6 0404_H
IDERRINT_PE1L: FFC6 0404_H
IDERRINT_PE1LL: FFC6 0404_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 33.46 IDERRINT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(3) IDSTCLR_PE1 — Instruction Cache Data RAM Error Status Clear Register

IDSTCLR clears the error flags in the error status register (ID1STERSTR), the overflow flag in the error overflow status register (IDOVFSTR), and the error address register (ID1STEADR). IDSTCLR is a write-only register and is always read as 0.

Access: IDSTCLR_PE1 is a write-only register that can be written in 32-bit units.
IDSTCLR_PE1L is a write-only register that can be written in 16-bit units.
IDSTCLR_PE1LL is a write-only register that can be written in 8-bit units.

Address: IDSTCLR_PE1: FFC6 0408$_H$
IDSTCLR_PE1L: FFC6 0408$_H$
IDSTCLR_PE1LL: FFC6 0408$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR1	STCLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 33.47 IDSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing, write the value after reset.
1	STCLR1	Error Status Flag Clear (for bank 1) Writing 1 to this bit clears the DEDF1and SEDF1 flags in ID1STERSTR; ERROVF1 flag in IDOVFSTR; and ID1STEADR1.
0	STCLR0	Error Status Flag Clear (for bank 0) Writing 1 to this bit clears the DEDF0and SEDF0 flags in ID1STERSTR; ERROVF0 flag in IDOVFSTR; and ID1STEADR0.

(4) IDOVFSTR_PE1 — Instruction Cache Data RAM Error Count Overflow Status Register

IDOVFSTR monitors occurrence of error overflow in cache data RAM. If a second error occurs after the first error (while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set. ERROVF is cleared by an internal reset, an external reset, or setting the STCLR bit to 1 in IDSTCLR.

Access: IDOVFSTR_PE1 is a read-only register that can be read in 32-bit units.
IDOVFSTR_PE1L is a read-only register that can be read in 16-bit units.
IDOVFSTR_PE1LL is a read-only register that can be read in 8-bit units.

Address: IDOVFSTR_PE1: FFC6 040C$_H$
IDOVFSTR_PE1L: FFC6 040C$_H$
IDOVFSTR_PE1LL: FFC6 040C$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF1	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.48 IDOVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	ERROVF1	Error Overflow Flag (for bank 1) ERROVF1 is set if a second error occurs while any of the error flags (DEDF1 and SEDF1) in the error status register is set. ERROVF1 is not set when the second error occurs at the same address or source like the first error.
0	ERROVF0	Error Overflow Flag (for bank 0) ERROVF0 is set if a second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register is set. ERROVF0 is not set when the second error occurs at the same address or source like the first error.

(5) ID1STERSTR_PE1 — Instruction Cache Data RAM 1st Error Status Register

ID1STERSTR monitors occurrence of the first error in cache data RAM. The error detection and flag update conditions for bank 0 and bank 1 are independent. The error status is set if an error occurs while both error flags (DEDFn and SEDFn) in the same bank are 0. In addition, DEDFn is set if an ECC 2-bit error occurs while SEDFn is set. In this case, SEDFn is not modified. Note that SEDFn is not set if an ECC 1-bit error occurs while DEDFn is already set in the same bank.

The ID1STERSTR register is cleared to 0 by a system reset or by setting the STCLR bit in the IDSTCLR register to 1.

Note that the Way where an error occurred cannot be identified by using this register.

Access: ID1STERSTR_PE1 is a read-only register that can be read in 32-bit units.
ID1STERSTR_PE1L is a read-only register that can be read in 16-bit units.
ID1STERSTR_PE1LL and ID1STERSTR_PE1LH are the read-only registers that can be read in 8-bit units.

Address: ID1STERSTR_PE1: FFC6 0410$_H$
ID1STERSTR_PE1L: FFC6 0410$_H$
ID1STERSTR_PE1LL: FFC6 0410$_H$
ID1STERSTR_PE1LH: FFC6 0411$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DEDF1	SEDF1	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.49 ID1STERSTR Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, the value after reset is returned.
9	DEDF1	ECC 2-bit Error Monitor Flag (for bank 1) 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. Clearing condition: Set the STCLR1 bit in IDSTCLR to 1. Setting condition: ECC 2-bit error is detected when DEDF1 is 0.
8	SEDF1	ECC 1-bit Error Monitor Flag (for bank 1) 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. Clearing condition: Set the STCLR1 bit in IDSTCLR to 1. Setting condition: ECC 1-bit error is detected when both SEDF1 and DEDF1 are 0.
7 to 2	Reserved	When read, the value after reset is returned.
1	DEDF0	ECC 2-bit Error Monitor Flag (for bank 0) 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. Clearing condition: Set the STCLR0 bit in IDSTCLR to 1. Setting condition: ECC 2-bit error is detected when DEDF0 is 0.
0	SEDF0	ECC 1-bit Error Monitor Flag (for bank 0) 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. Clearing condition: Set the STCLR0 bit in IDSTCLR to 1. Setting condition: ECC 1-bit error is detected when both SEDF0 and DEDF0 are 0.

(6) ID1STEADRn_PE1 — Instruction Cache Data RAM (Bank n) 1st Error Address Register (n = 0, 1)

The ID1STEADR register holds the address at which an error occurred. It holds an error address if an error occurs while all error flags for the same bank are 0 in the ID1STEADR register. In addition, the address information is updated if an ECC 2-bit error occurs while the error flag of the 1st error indicates an ECC 1-bit error. The address information will not be updated after an ECC 2-bit error has occurred. This register holds the physical address of the cache RAM. EADRn[6:0] store the entry address, and EADRn[8] stores the Way group number. Note that EADRn[7] is always 0. The ID1STERSTR register is cleared to 0 by a system reset or by setting the STCLR bit in the IDSTCLR register to 1.

Access: ID1STEADRn_PE1 is a read-only register that can be read in 32-bit units.
ID1STEADRn_PE1L is a read-only register that can be read in 16-bit units.
ID1STEADRn_PE1LL and ID1STEADRn_PE1LH are the read-only registers that can be read in 8-bit units.

Address: ID1STEADRn_PE1: FFC6 0450_H + n × 4_H
ID1STEADRn_PE1L: FFC6 0450_H + n × 4_H
ID1STEADRn_PE1LL: FFC6 0450_H + n × 4_H
ID1STEADRn_PE1LH: FFC6 0451_H + n × 4_H

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	EADRn[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.50 ID1STEADRn Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned.
8 to 0	EADRn[8:0]	1st Error Address (for bank n) Monitors the address of the first error. The error address is updated if an error occurs while all the error flags for bank n are 0 in ID1STERSTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error occurs, the address is not updated.

Note: Regarding the Way group, please refer to Section 3.2.2.2 Instruction Cache Function.

(7) ITECCCTL_PE1 — Instruction Cache Tag RAM ECC Control Register

ITECCCTL enables or disables ECC error detection in cache tag RAM. Set the PROT1 and PROT0 bits to 01_B when writing to ITECCCTL.

Access: ITECCCTL_PE1 can be read or written in 32-bit units.
ITECCCTL_PE1L can be read or written in 16-bit units.

Address: ITECCCTL_PE1: FFC6 1400_H
ITECCCTL_PE1L: FFC6 1400_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECCDIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 33.51 ITECCCTL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15	PROT1	Enables or disables modification of the ECCDIS bit. The value written is not retained. These bits are always read as 0. Set PROT1 to 0 and PROT0 to 1 when writing to this register.
14	PROT0	
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	ECCDIS	ECC Disable Enables or disables ECC error detection. When writing to this bit, write 01 to PROT1 and PROT0 at the same time. 0: Enables ECC error detection. 1: Disables ECC error detection.

(8) ITERRINT_PE1 — Instruction Cache Tag RAM Error Information Control Register

ITERRINT enables or disables generation of the error notification signal upon detection of a 1-bit ECC error in cache tag RAM.

Access: ITERRINT_PE1 can be read or written in 32-bit units.
ITERRINT_PE1L can be read or written in 16-bit units.
ITERRINT_PE1LL can be read or written in 8-bit units.

Address: ITERRINT_PE1: FFC6 1404$_H$
ITERRINT_PE1L: FFC6 1404$_H$
ITERRINT_PE1LL: FFC6 1404$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 33.52 ITERRINT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection is enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

(9) ITSTCLR_PE1 — Instruction Cache Tag RAM Error Status Clear Register

ITSTCLR clears the error flags in the error status register (IT1STERSTR), the overflow flag in the error overflow status register (ITOVFSTR), and the error address register (IT1STEADR). ITSTCLR is a write-only register and is always read as 0.

Access: ITSTCLR_PE1 is a write-only register that can be written in 32-bit units.
ITSTCLR_PE1L is a write-only register that can be written in 16-bit units.
ITSTCLR_PE1LL is a write-only register that can be written in 8-bit units.

Address: ITSTCLR_PE1: FFC6 1408$_H$
ITSTCLR_PE1L: FFC6 1408$_H$
ITSTCLR_PE1LL: FFC6 1408$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 33.53 ITSTCLR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing, write the value after reset.
0	STCLR0	Error Status Flag Clear Writing 1 to this bit clears the DEDF0 and SEDF0 flags in IT1STERSTR; ERROVF0 flag in ITOVFSTR; and IT1STEADR0.

(10) ITOVFSTR_PE1 — Instruction Cache Tag RAM Error Count Overflow Status Register

ITOVFSTR monitors occurrence of error overflow in cache tag RAM. If a second error occurs after the first error (while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set. ERROVF is cleared by an internal reset, an external reset, or setting the STCLR bit to 1 in ITSTCLR.

Access: ITOVFSTR_PE1 is a read-only register that can be read in 32-bit units.
ITOVFSTR_PE1L is a read-only register that can be read in 16-bit units.
ITOVFSTR_PE1LL is a read-only register that can be read in 8-bit units.

Address: ITOVFSTR_PE1: FFC6 140C$_H$
ITOVFSTR_PE1L: FFC6 140C$_H$
ITOVFSTR_PE1LL: FFC6 140C$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.54 ITOVFSTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	Error Overflow Flag ERROVF0 is set if a second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register is set. ERROVF0 is not set when the second error occurs at the same address or source like the first error.

(11) IT1STERSTR_PE1 — Instruction Cache Tag RAM 1st Error Status Register

IT1STERSTR monitors occurrence of the first error in cache tag RAM.

The error flag is overwritten if an ECC 2-bit error occurs while the error flag is 0.

Note that SEDF0 is not set if an ECC 1-bit error occurs when DEDF0 is already set. The IT1STERSTR register is cleared by an internal reset, an external reset, or by setting the STCLR bit in the ITSTCLR register to 1. Note that the Way where an error occurred cannot be identified by using this register.

Access: IT1STERSTR_PE1 is a read-only register that can be read in 32-bit units.
IT1STERSTR_PE1L is a read-only register that can be read in 16-bit units.
IT1STERSTR_PE1LL is a read-only register that can be read in 8-bit units.

Address: IT1STERSTR_PE1: FFC6 1410$_H$
IT1STERSTR_PE1L: FFC6 1410$_H$
IT1STERSTR_PE1LL: FFC6 1410$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.55 IT1STERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF0	ECC 2-bit Error Monitor Flag 0: ECC 2-bit error is not detected. 1: ECC 2-bit error is detected. Clearing condition: Set the STCLR0 bit in ITSTCLR to 1. Setting condition: ECC 2-bit error is detected when DEDF0 is 0.
0	SEDF0	ECC 1-bit Error Monitor Flag 0: ECC 1-bit error is not detected. 1: ECC 1-bit error is detected. Clearing condition: Set the STCLR0 bit in ITSTCLR to 1. Setting condition: ECC 1-bit error is detected when both SEDF0 and DEDF0 are 0.

(12) IT1STEADR0_PE1 — Instruction Cache Tag RAM 1st Error Address Register

The IT1STEADR register holds the address at which an error occurred. It holds the error address if an error occurs while all error flags for the relevant banks are 0 in the IT1STERSTR register. In addition, the address information is updated if an ECC 2-bit error occurs while the error flag of the 1st error indicates an ECC 1-bit error. The address information will not be updated after an ECC 2-bit error has occurred. In this register, EADRn[6:0] store the entry address, and EADRn[8] stores the Way group number. Note that EADR[7] is always 0. The IT1STEADR register is cleared by an internal reset, an external reset, or by setting the STCLR bit in the ITSTCLR register to 1.

Access: IT1STEADR0_PE1 is a read-only register that can be read in 32-bit units.
IT1STEADR0_PE1L is a read-only register that can be read in 16-bit units.
IT1STEADR0_PE1LL and IT1STEADR0_PE1LH are the read-only registers that can be read in 8-bit units.

Address: IT1STEADR0_PE1: FFC6 1450$_H$
IT1STEADR0_PE1L: FFC6 1450$_H$
IT1STEADR0_PE1LL: FFC6 1450$_H$
IT1STEADR0_PE1LH: FFC6 1451$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	EADR[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.56 IT1STEADR0 Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned.
8 to 0	EADR[8:0]	1st Error Address Monitors the address of the first error. The error address is updated if an error occurs while all the error flags are 0 in IT1STERSTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. If an ECC 2-bit error occurs, the address is not updated.

33.3 Memory Protection

33.3.1 Overview

This product incorporates the memory protection function to prevent erroneous accesses to data in memories and control registers of the peripheral circuits.

- MPU
  The CPU protects memory against illegal access by itself. The CPU does not generate the signals for access to addresses where access is prohibited by the MPU. For details, see the *RH850G3M User's Manual: Software*.
- Slave Guard
  A specific memory is protected against illegal accesses from any bus master. Slave guard includes the following guard types. The details of each type are given in the following sections.
  - PEG
    The local RAM is protected against illegal accesses. However, accesses from the CPU incorporating the local RAM itself are excluded. For example, accesses from the CPU1 to local RAM in the CPU1 are not rejected by the PEG.
    For details, refer to **Section 3, CPU System**.
  - GRG
    The retention RAM is protected against illegal accesses.

33.3.1.1 Identifiers for Slave Guard

For the slave guard function, the type of illegal accesses to be rejected can be designated using the following identifiers.

Table 33.57 Indices

Identifier	Function
UM	When the CPU makes an access, this indicates the operating mode of the CPU. 0: Supervisor mode 1: User mode When the DMAC makes an access, the value of this identifier is the value in the channel master setting register. When another master makes an access, the value of this identifier is always 0.
SPID	When the CPU makes an access, this indicates the system protection identifier SPID that is assigned to the CPU. When the DMAC makes an access, the value of this identifier is the value in the channel master setting register. When another master makes an access, the value of this identifier is always 00_B.
PEID	This indicates the access source bus master. 000_B: Reserved 001_B: CPU1 010_B: Reserved 011_B: Reserved 100_B: Other bus master (H-Bus bus master) 101_B: Reserved 110_B: Reserved 111_B: Reserved When the DMAC makes an access, the value of this identifier is the value in the channel master setting register.

33.3.2 GRG (Global RAM Guard)

This product is provided with 4-channel GRG, which is described in detail in the following sections.

CAUTION

Unlike the RH850/F1H, the RH850/F1M is not equipped with the global RAM, and the global RAM area is used as the retention RAM area. Therefore, use the GRG (global RAM guard) as the retention RAM Guard. Descriptions of register names and functions for the global RAM should be treated as those of the retention RAM.

33.3.2.1 List of Registers

Table 33.58 List of Registers

Register Name	Symbol	Address
GRG protection setting register 0	MGDGRPROT0	FFC4 9000$_H$
GRG compare base address register 0	MGDGRBAD0	FFC4 9004$_H$
GRG valid compare address register 0	MGDGRADV0	FFC4 9008$_H$
GRG protection setting register 1	MGDGRPROT1	FFC4 9010$_H$
GRG compare base address register 1	MGDGRBAD1	FFC4 9014$_H$
GRG valid compare address register 1	MGDGRADV1	FFC4 9018$_H$
GRG protection setting register 2	MGDGRPROT2	FFC4 9020$_H$
GRG compare base address register 2	MGDGRBAD2	FFC4 9024$_H$
GRG valid compare address register 2	MGDGRADV2	FFC4 9028$_H$
GRG protection setting register 3	MGDGRPROT3	FFC4 9030$_H$
GRG compare base address register 3	MGDGRBAD3	FFC4 9034$_H$
GRG valid compare address register 3	MGDGRADV3	FFC4 9038$_H$
GRG control register (VCI)	MGDGRSCTL_VCI	FFC4 9100$_H$
GRG error status register (VCI)	MGDGRSSTAT_VCI	FFC4 9104$_H$
GRG error access type register (VCI)	MGDGRSTYPE_VCI	FFC4 910C$_H$
GRG control register (PE1)	MGDGRSCTL_PE1	FFC4 9200$_H$
GRG error status register (PE1)	MGDGRSSTAT_PE1	FFC4 9204$_H$
GRG error access type register (PE1)	MGDGRSTYPE_PE1	FFC4 920C$_H$

- MGDGRPROTn, MGDGRBADn, and MGDGRADVn set the protection specifications for each channel (n: 0 to 3).
- MGDGRSCTL_*, MGDGRSSTAT_*, and MGDGRSTYPE_* indicate error information on each access port: “_VCI” represents access from the system interconnect to the retention RAM, “_PE1” represents access from the CPU1 to the retention RAM.

33.3.2.2 Details of Registers

(1) MGDGRPROTn — GRG Protection Setting Register n (n = 0 to 3)

Access: MGDGRPROTn can be read or written in 32-bit units.
MGDGRPROTnL and MGDGRPROTnH can be read or written in 16-bit units.
MGDGRPROTnLL, MGDGRPROTnHL, and MGDGRPROTnHH can be read or written in 8-bit units.

Address: MGDGRPROTn: FFC4 9000_H + n × 10_H
MGDGRPROTnL: FFC4 9000_H + n × 10_H
MGDGRPROTnLL: FFC4 9000_H + n × 10_H
MGDGRPROTnH: FFC4 9002_H + n × 10_H
MGDGRPROTnHL: FFC4 9002_H + n × 10_H
MGDGRPROTnHH: FFC4 9003_H + n × 10_H

Value after reset: 07FF FFF0$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	EN	—	—	—	—	UM	—	—	—	PEID4	—	—	PEID1	—	—
Value after reset	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R/W	R	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SPID1	SPID0	—	—	—	—	—
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R

Table 33.59 MGDGRPROTn Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
30	EN	Protection Enable 0: Disables protection. 1: Enables protection. Only access permitted by this register is possible.
29 to 26	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
25	UM	User Mode Access 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode.
24 to 22	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
21	PEID4	Access with PEID = 4 (peripheral device connected to H-BUS)*1 0: Disables access with PEID4. 1: Enables access with PEID4.
20, 19	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
18	PEID1	Access with PEID = 1 (CPU1)*1 0: Disables access with PEID1. 1: Enables access with PEID1.
17 to 7	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
6	SPID1	Access with SPID = 1 (CPU1)*2 0: Disables access with SPID1. 1: Enables access with SPID1.
5	SPID0	Access with SPID = 0 (peripheral device connected to H-BUS)*2 0: Disables access with SPID0. 1: Enables access with SPID0.

Table 33.59 MGDGRPROTn Register Contents (2/2)

Bit Position	Bit Name	Function
4 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. Access with PEID
PEID is a bit list with each bit corresponding to a PEID value.
Setting multiple bits enables ID values of multiple bus masters at the same time.

Note 2. Access with SPID
SPID is a bit list with each bit corresponding to a SPID value.
Setting multiple bits enables ID values of multiple masters at the same time.

CAUTION

Global RAM Guard (GRG) only supports write accesses. The guard setting is not enabled for read accesses.

(2) MGDGRBADn — GRG Compare Base Address Register n (n = 0 to 3)

Access: MGDGRBADn can be read or written in 32-bit units.
MGDGRBADnL and MGDGRBADnH can be read or written in 16-bit units.
MGDGRBADnLH and MGDGRBADnHL can be read or written in 8-bit units.

Address: MGDGRBADn: FFC4 9004_H + n × 10_H
MGDGRBADnL: FFC4 9004_H + n × 10_H
MGDGRBADnLH: FFC4 9005_H + n × 10_H
MGDGRBADnH: FFC4 9006_H + n × 10_H
MGDGRBADnHL: FFC4 9006_H + n × 10_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	AD[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD[15:9]							—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R

Table 33.60 MGDGRBADn Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 9	AD[20:9]	Compare Base Address*1 These bits set the base address of protection setting domain specified by the GRG protection setting register n (MGDGRPROTn).
8 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Note 1. For details, refer to **33.3.2.2 (3), MGDGRADVn — GRG Valid Compare Address Register n (n = 0 to 3) Setting example:** .

(3) MGDGRADVn — GRG Valid Compare Address Register n (n = 0 to 3)

Access: MGDGRADVn can be read or written in 32-bit units.
MGDGRADVnL and MGDGRADVnH can be read or written in 16-bit units.
MGDGRADVnLH and MGDGRADVnHL can be read or written in 8-bit units.

Address: MGDGRADVn: FFC4 9008_H + n × 10_H
MGDGRADVnL: FFC4 9008_H + n × 10_H
MGDGRADVnLH: FFC4 9009_H + n × 10_H
MGDGRADVnH: FFC4 $900A_H$ + n × 10_H
MGDGRADVnHL: FFC4 $900A_H$ + n × 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	ADV[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADV[15:9]							—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R

Table 33.61 MGDGRADVn Register Contents

Bit Position	Bit Name	Function
31 to 21	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
20 to 9	ADV[20:9]	Valid Compare Address Setting MGDGRADVn[20:9] to 1 executes address compare. If all the bits of MGDGRADVn[20:9] are 1, 512 bytes , which is the minimum unit based on the address specified by MGDGRBADn, are protected. However, if all the bits of MGDGRADVn[20:9] are 0, the all areas of global RAM are protected.
8 to 0	Reserved	When read, the value after reset is returned. When writing, write the value after reset.

Setting example:

If MGDGRBADn[20:9] is 800_H and MGDGRADVn[20:9] is $FF7_H$, global RAM guard protection area n is from FEF0 0000_H to FEF0 $01FF_H$ and FEF0 1000_H to FEF0 $11FF_H$.

Concept:

When MGDGRBADn[20:9] is 800_H, the base address is FEF0 0000_H. The settable range is shown by an underline as follows:

1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B
F				E				F				0				0				0				0				H

The settable range is as follows when MGDGRADVn[20:9] is $FF7_H$:

0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	B

The area to which protection applies is as shown below because the bits set to 0 and the lower nine bits are not applicable:

1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	×	0	0	0	×	×	×	×	×	×	×	×	×	B

Therefore the protection range is as follows:

1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B

to

1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	B

and

1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	B

to

1	1	1	1	1	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	B

The following 512 bytes each (a total of 1 Kbyte) are protected (in hexadecimal):

- FEF0 0000_H to FEF0 $01FF_H$
- FEF0 1000_H to FEF0 $11FF_H$

(4) MGDGRSCTL_VCI/PE1 — GRG Control Register

Access: MGDGRSCTL_VCI and MGDGRSCTL_PE1 can be read or written in 32-bit units.
MGDGRSCTL_VCIL and MGDGRSCTL_PE1L can be read or written in 16-bit units.
MGDGRSCTL_VCILL and MGDGRSCTL_PE1LL can be read or written in 8-bit units.

Address: MGDGRSCTL_VCI: FFC4 9100H
MGDGRSCTL_VCIL: FFC4 9100_H
MGDGRSCTL_VCILL: FFC4 9100_H
MGDGRSCTL_PE1: FFC4 9200_H
MGDGRSCTL_PE1L: FFC4 9200_H
MGDGRSCTL_PE1LL: FFC4 9200_H

Value after reset: 0000 0000_H

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERRCLO	ERRCLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 33.62 MGDGRSCTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1	ERRCLO	Error Entry Overflow Flag Clear 0: No operation 1: Clears the overflow flag.
0	ERRCLE	GRG protection violation Error Flag Clear 0: No operation 1: Clears the error flag. Set this bit at the same time as ERRCLO as shown in **Table 33.63**.

Table 33.63 Combinations of ERRCLO and ERRCLE

ERRCLO	ERRCLE	Function
0	0	Clears neither of the bits.
0	1	Setting prohibited
1	0	Clears the OVF bit.
1	1	Clears the OVF and ERR bits.

(5) MGDGRSSTAT_VCI/PE1 — GRG Error Status Register

Access: MGDGRSSTAT_VCI and MGDGRSSTAT_PE1 are the read-only registers that can be read in 32-bit units.
MGDGRSSTAT_VCIL and MGDGRSSTAT_PE1L are the read-only registers that can be read in 16-bit units.
MGDGRSSTAT_VCILL and MGDGRSSTAT_PE1LL are the read-only registers that can be read in 8-bit units.

Address: MGDGRSSTAT_VCI: FFC4 9104$_H$
MGDGRSSTAT_VCIL: FFC4 9104$_H$
MGDGRSSTAT_VCILL: FFC4 9104$_H$
MGDGRSSTAT_PE1: FFC4 9204$_H$
MGDGRSSTAT_PE1L: FFC4 9204$_H$
MGDGRSSTAT_PE1LL: FFC4 9204$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.64 MGDGRSSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	OVF	Error Entry Overflow Flag 0: No overflow 1: An overflow occurred. If a second guard violation occurs with the error detection flag being set after the first guard violation occurs, the error entry overflows and this flag is set because the number of GRG error entry stages is 1. In addition, it is not possible to determine whether overflow has occurred in INTGURD. Check the value of this bit. The error information of the guard violation when an overflow occurs are not captured.
0	ERR	GRG protection violation Error Flag 0: No GRG protection violation has occurred. 1: A GRG protection violation has occurred.

(6) MGDGRSTYPE_VCI/PE1 — GRG Error Access Type Register

Access: MGDGRSTYPE_VCI and MGDGRSTYPE_PE1 are the read-only registers that can be read in 32-bit units. MGDGRSTYPE_VCIL and MGDGRSTYPE_PE1L are the read-only registers that can be read in 16-bit units. MGDGRSTYPE_VCILL, MGDGRSTYPE_VCILH, MGDGRSTYPE_PE1LL, and MGDGRSTYPE_PE1LH are the read-only registers that can be read in 8-bit units.

Address: MGDGRSTYPE_VCI: FFC4 $910C_H$
MGDGRSTYPE_VCIL: FFC4 $910C_H$
MGDGRSTYPE_VCILL: FFC4 $910C_H$
MGDGRSTYPE_VCILH: FFC4 $910D_H$
MGDGRSTYPE_PE1: FFC4 $920C_H$
MGDGRSTYPE_PE1L: FFC4 $920C_H$
MGDGRSTYPE_PE1LL: FFC4 $920C_H$
MGDGRSTYPE_PE1LH: FFC4 $920D_H$

Value after reset: $0000\ 0000_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEID[2:0]			—	—	—	SPID[1:0]		—	UM	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.65 MGDGRSTYPE Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as an undefined value.
15 to 13	PEID[2:0]	The PEID of the originator of the access which led to a GRG protection violation
12 to 10	Reserved	When read, the value after reset is returned.
9 and 8	SPID[1:0]	The SPID of the originator of the access which led to a GRG protection violation
7	Reserved	When read, the value after reset is returned.
6	UM	The UM of the originator of the access which led to a GRG protection violation
5 to 0	Reserved	These bits are read as an undefined value.

Section 34 Data CRC (DCRA)

This section contains a generic description of the data CRC function A (DCRA).

The first part in this section describes the RH850/F1M specific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the DCRA.

34.1 Features of RH850/F1M DCRA

34.1.1 Number of Units

This microcontroller has the following number of DCRA units.

Table 34.1 Number of Units

Product Name	RH850/F1M 144 pins	RH850/F1M 176 pins	RH850/F1M 233 pins
Number of Units	4		
Name	DCRAn (n = 0 to 3)		

Table 34.2 Index

Index	Description
n	Throughout this section, the individual data CRC function A units are identified by the index “n” (n = 0 to 3); for example, DCRAnCTL indicates the DCRAn control register.

34.1.2 Register Base Address

DCRAn base addresses are listed in the following table.

DCRAn register addresses are given as offsets from the base addresses.

Table 34.3 Register Base Addresses

Base Address Name	Base Address
<DCRA0_base>	FFF7 0000_H
<DCRA1_base>	FFF7 1000_H
<DCRA2_base>	FFF7 2000_H
<DCRA3_base>	FFF7 3000_H

34.1.3 Clock Supply

The DCRAn clock supply is shown in the following table.

Table 34.4 Clock Supply

Unit Name	Unit Clock Name	Clock Supply Name
DCRAn	PCLK	CPUCLK4
	Register access clock	CPUCLK4

34.1.4 Reset Sources

DCRAn reset sources are listed in the following table. DCRAn is initialized by these reset sources.

Table 34.5 Reset Sources

Unit Name	Reset Source
DCRAn	All reset sources (ISORES)

34.2 Overview

34.2.1 Functional Overview

The data CRC function A can be used to verify or generate CRC protected data streams of arbitrary length and different bit widths.

- 32-bit Ethernet CRC
  $(X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X^{1}+1)$
- 16-bit CCITT CRC
  $(X^{16}+X^{12}+X^{5}+1)$
- CRC of an arbitrary data block length can be generated.
- After initialization of the CRC data register, every write access to the CRC input register generates a new CRC according to the selected polynomial, and the result is stored in the CRC data register.

34.2.2 Block Diagram

The following picture shows the block diagram of the data CRC function A.

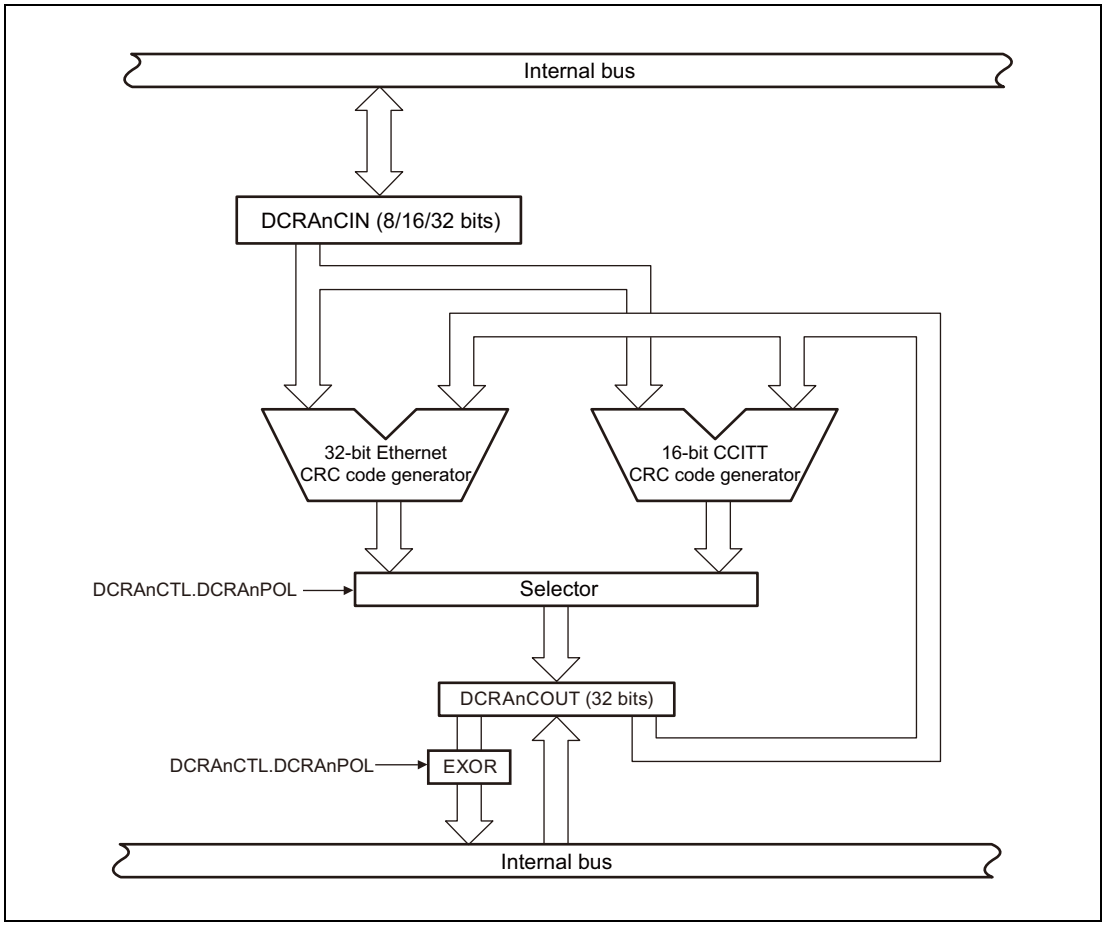
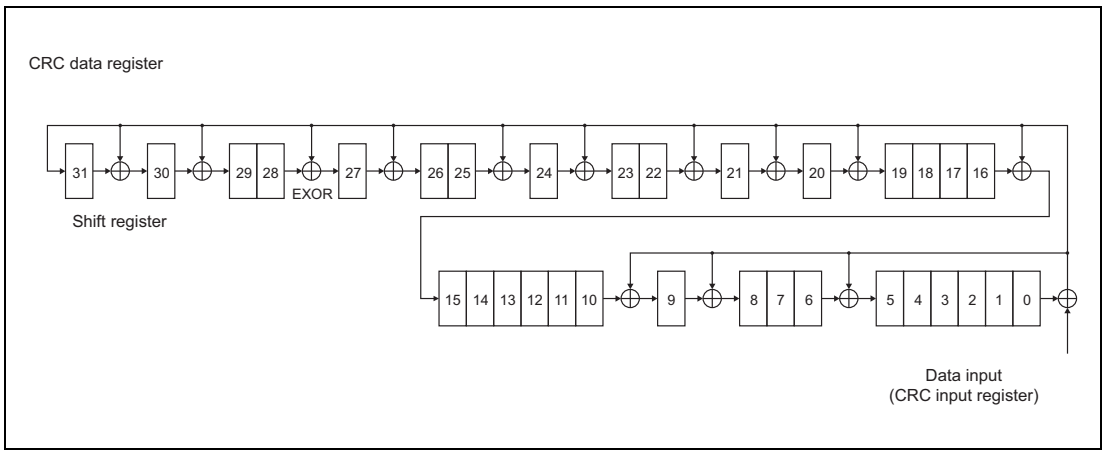


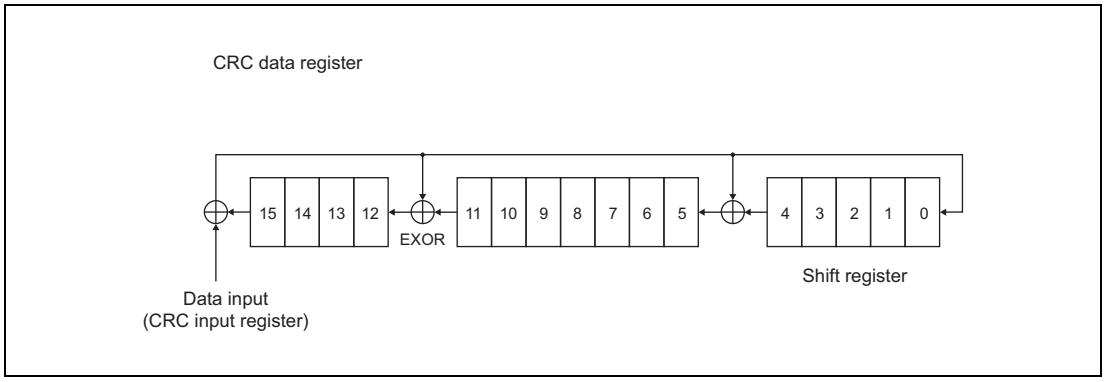
Figure 34.1 Block Diagram of Data CRC Function A

34.2.3 Operational Circuit

- 32-bit Ethernet



- 16-bit CCITT



34.3 Registers

34.3.1 List of Registers

DCRA registers are listed in the following table.

For details about <DCRAn_base>, see **Section 34.1.2, Register Base Address**.

Table 34.6 List of Registers

Module Name	Register Name	Symbol	Address
DCRAn	CRC input register	DCRAnCIN	<DCRAn_base> + 00_H
DCRAn	CRC data register	DCRAnCOUT	<DCRAn_base> + 04_H
DCRAn	CRC control register	DCRAnCTL	<DCRAn_base> + 20_H

34.3.2 DCRAnCIN — CRC Input Register

This register holds the input data for CRC calculation. The effective bit width used for CRC calculation must be set by DCRAnCTL.DCRAnISZ[1:0].

When data is written to this register, the CRC code is generated.

The CRC calculation is immediately started after the DCRAnCIN register is written. The DCRAnCOUT register must be initialized with the initial starting value, before the first data of the data block is written to DCRAnCIN register.

Access: This register can be read or written in 32-bit units.

Address: <DCRAn_base>

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCRAnCIN[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCRAnCIN[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.7 DCRAnCIN Register Contents

Bit Position	Bit Name	Function
31 to 0	DCRAnCIN [31:0]	Input Data for CRC Calculation The valid bits are: • For 32 bit effective bit width: DCRAnCIN[31:0] • For 16 bit effective bit width: DCRAnCIN[15:0] • For 8 bit effective bit width: DCRAnCIN[7:0]

34.3.3 DCRAnCOUT — CRC Data Register

This register stores the result of the CRC code generated by the 32-bit Ethernet polynomial or the 16-bit CCITT polynomial.

Access: This register can be read or written in 32-bit units.

Address: <DCRAn_base> + 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCRAnCOUT[31:16]															
Value after reset *1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCRAnCOUT[15:0]															
Value after reset *1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The read value after reset is 0000 0000_H since the 32-bit Ethernet CRC polynomial is selected as the CRC generating function after reset.

Table 34.8 DCRAnCOUT Register Contents

Bit Position	Bit Name	Function
31 to 0	DCRAnCOUT [31:0]	Result of the CRC Code Generation When the 16-bit CCITT polynomial is enabled, the bits 15 to 0 show the CRC result. The bits 31 to 16 are undefined. The read value of this register is a value obtained by performing EXOR calculation for the following value: • For 32-bit Ethernet polynomial: FFFF $FFFF_H$ • For 16-bit CCITT polynomial: 0000_H For example, when DCRAnCOUT = 5555 5555_H for the 32-bit Ethernet polynomial, AAAA $AAAA_H$ is read.

CAUTION

This register must initialized by setting the initial start value before the first data of the data block is written to DCRAnCIN register.

34.3.4 DCRAnCTL — CRC Control Register

This register controls the CRC generation process.

Access: This register can be read or written in 8-bit units.

Address: <DCRAn_base> + 20_H

Value after reset: 00_H

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

Table 34.9 DCRAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
2, 1	DCRAnISZ[1:0]	Specify the CRC input bit width. 00: 32 bits (DCRAnCIN[31:0]) 01: 16 bits (DCRAnCIN[15:0]) 10: 8 bits (DCRAnCIN[7:0]) 11: Setting prohibited
0	DCRAnPOL	Specifies the CRC generation method. 0: 32-bit Ethernet CRC polynomial generation. The byte order of the DCRAnCIN register is LSB (least significant bit) first. This means that, if the input bit width is 8 bits (DCRAnISZ[1:0] = 10_B), bit positions 7 to 0 of the DCRAnCIN register contain the input data and bit position 0 (LSB) is the start bit of the input data. 1: 16-bit CCITT CRC polynomial generation. The byte order of the DCRAnCIN register is MSB (most significant bit) first. This means that, if the input bit width is 8 bits (DCRAnISZ[1:0] = 10_B), bit positions 7 to 0 of the DCRAnCIN register contain the input data and bit position 7 (MSB) is the start bit of the input data.

CAUTION

- If the CRC generation method (DCRAnCTL.DCRAnPOL) is changed, the DCRAnCOUT register must be initialized by setting the initial start value.
- The CRC input bit width (DCRAnCTL.DCRAnISZ[1:0]) must be set according to the data block bit width. Changing the CRC input bit width is not allowed during processing of a data block (a data block consists of N bytes, half-words or one word). After the final CRC result is read from DCRAnCOUT register, the bit width can be changed. In that case, the DCRAnCOUT register must be initialized with the initial start value.

34.4 Operation

The data CRC function A generates a CRC (cyclic redundancy check) of an arbitrary data block length. The data is forwarded to the data CRC function in 8-, 16- or 32-bit units. The CRC polynomial can either be selected for 32-bit Ethernet or 16-bit CCITT. The initial starting value must be set at the DCRAnCOUT register before the first write access to the CRC input register (DCRAnCIN) is performed.

The flowchart below shows the CRC generating procedure.

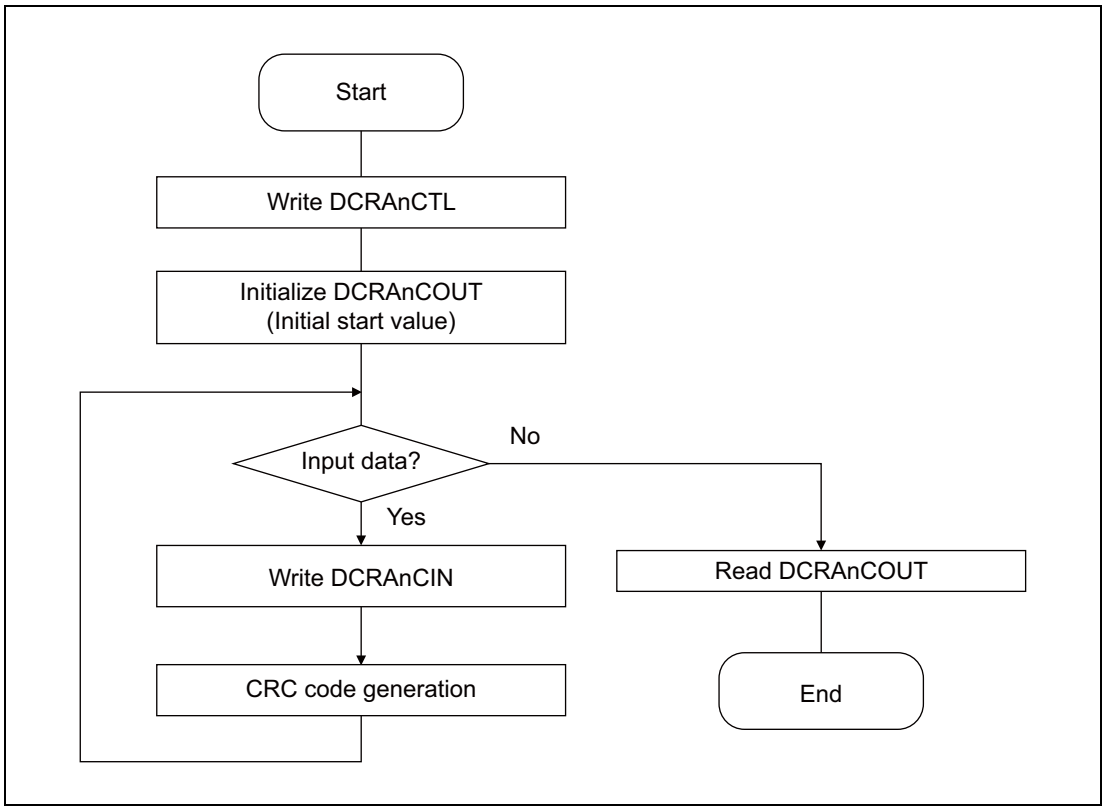


Figure 34.2 Flowchart of Data CRC Function A

NOTES

1. Before writing the first data to DCRAnCIN, the CRC output register DCRAnCOUT must be initialized with the initial start value.
2. DCRAnCOUT must be re-initialized by setting the initial start value when the polynomial is changed by changing DCRAnCTL.DCRAnPOL.
3. Setting example of the initial start values of the respective polynomials
   The following is the example of setting values.

Table 34.10 Setting Example of Initial Start Values (When Read at a Reset)

	Initial Start Value	EXOR Value	DCRAnCOUT Read Value
16-bit CCITT	XXXX $FFFF_H$	XXXX 0000_H	XXXX $FFFF_H$
32-bit Ethernet	$FFFF\ FFFF_H$	$FFFF\ FFFF_H$	$0000\ 0000_H$

Note: X: undefined

Section 35 On-Chip Debug Unit (OCD)

This microcontroller has an on-chip debug function. By using the on-chip debug emulator, programs can be debugged with the microcontroller mounted in the target system.

The debug functions incorporated in this microcontroller conform to IEEE-ISTO 5001™-2003 Class 3*1, a Nexus debug interface standard.

Note 1. This function is supported only by products with an $\overline{\text{EVTO}}$ pin.

CAUTION

The debug functions described in this section are supported by the microcontroller but whether they are usable depends on the debugger. For details on debugging, see the user's manual of the debugger.

35.1 Overview of RH850/F1M OCD

35.1.1 Functional Overview

The on-chip debug functions described below are supported by the microcontroller.

(1) Debug interface

This microcontroller supports the following as debug interfaces: Nexus Interface, Low Pin Debug Interface (1-pin) - hereinafter called “LPD (1-pin)”, and Low Pin Debug Interface (4-pin) - hereinafter called “LPD (4-pin)”.

On-chip debug can be performed using these debug interfaces.

(2) Debug monitoring function

Debug-dedicated monitor program space is mounted and is used during debugging.

The basic debug functions below can be used by running a monitoring program.

- Downloading the user-created program
- Reading and writing the memory and registers
- Running the user-created program starting at any address

(3) On-chip break

A maximum of 12 breakpoints can be specified at any execution address. Of the 12 breakpoints, a maximum of four breakpoints can be specified for any access (access address, access data).

(4) Software break

Software break points can be specified at any execution address.

(5) Peripheral break

The peripheral break function generates a stop request to the peripheral modules of the microcontroller if the user-created program is stopped, for instance upon a breakpoint hit.

(6) Forced break

Execution of the user-created program can be interrupted forcibly.

(7) Forced reset

This device (microcontroller) can be forcibly reset.

(8) Real time RAM monitoring (RRM)

The memory can be read during program execution. Because this read access uses debug-dedicated DMA, it has minimal effect on program execution.

(9) Dynamic memory modification (DMM)

The memory can be written during program execution. Because this write access uses debug-dedicated DMA, it has minimal effect on program execution.

(10) Timer function

Using a 32-bit counter, the time for running the user-created program can be measured based on the clock for debug.

For the measurement accuracy, see the user's manual of the debugger.

(11) Mask function

Masking the following factors is possible.

- All reset sources except for a POC reset and a wakeup reset

(12) Hot plug-in function

Debugging can be started in normal operating mode without external reset input.

NOTE

When the hot plug-in function is used in power save mode, the INTDCUTDI interrupt is required to return from power save mode as the wake-up process.

(13) Security function

To prevent the contents of the flash memory from being read by an unauthorized person, a 128-bit ID code can be written to the microcontroller. If the code the user inputs when starting a debugger does not match the ID code written to the microcontroller, the flash memory cannot be accessed.

For details on how to set the ID code, see the user's manual of the debugger.

(14) Trace function

Execution history, data changes, etc. of the user-created program can be obtained.

NOTE

The trace function is only available in devices with 4-MB memory.

35.2 Peripheral Break Control

The peripheral break function generates a stop request to the peripheral modules of the microcontroller if the user-created program is stopped, for instance upon a breakpoint hit.

During peripheral break, the peripheral modules operate as follows.

a. Modules that stop unconditionally regardless of the EPC.SVSTOP setting

Table 35.1 Modules that Stop Unconditionally Regardless of the EPC.SVSTOP Setting

Module
Window watchdog timer (WDTA)

b. Modules that continue to operate by the setting of emulation registers even when EPC.SVSTOP = 1

Table 35.2 Modules that Continue to Operate by the Setting of Emulation Registers even when EPC.SVSTOP = 1

Module	Emulation Register	n
OS timer (OSTM)	OSTMnEMU.OSTMnSVSDIS 0: Stops during break 1: Continues during break	0 to 9
Timer array unit D (TAUD)	TAUDnEMU.TAUDnSVSDIS 0: Stops during break 1: Continues during break	0
Timer Array Unit B (TAUB)	TAUBnEMU.TAUBnSVSDIS 0: Stops during break 1: Continues during break	0, 1
Timer array unit J (TAUJ)	TAUJnEMU.TAUJnSVSDIS 0: Stops during break 1: Continues during break	0, 1
Real-Time Clock (RTCA)	RTCAnEMU.RTCAnSVSDIS 0: Stops during break 1: Continues during break	0
Clocked serial interface G (CSIG)	CSIGnEMU.CSIGnSVSDIS 0: Stops during break 1: Continues during break	0 to 3
Clocked serial interface H (CSIH)	CSIHnEMU.CSIHnSVSDIS 0: Stops during break 1: Continues during break	0 to 3
Timer Motor Control Function (TAPA)	TAPAnEMU.TAPAnSVSDIS 0: Stops during break 1: Continues during break	0
Encoder Timer (ENCA)	ENCAnEMU.ENCAnSVSDIS 0: Stops during break 1: Continues during break	0
PWM Output/Diagnostic (PWM-Diag)	PWBAnEMU.PWBAnSVSDIS 0: Stops during break 1: Continues during break	0
	PWSAnEMU.PWSAnSVSDIS 0: Stops during break 1: Continues during break	0
A/D converter (ADCA)	ADCAnEMU.ADCAnSVSDIS 0: Stops during break 1: Continues during break	0, 1

CAUTION

For details on the registers, see the register description of the corresponding section.

c. Modules that stop when EPC.SVSTOP = 1

Table 35.3 Modules that Stop when EPC.SVSTOP = 1

Module
LIN/UART interface (RLIN3)
Low-Power Sampler (LPS)

35.3 Registers

35.3.1 EPC — Emulation Peripheral Control Register

This register stops operation of peripheral functions (timer, serial interface, and A/D converter) in debug mode (SVSTOP).

Access: Accessing from the user program is prohibited.

Address: —

Value after reset: 00_H

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

Table 35.4 EPC Register Contents

Bit Position	Bit Name	Function
7	Reserved	—
6	SVSTOP	Stops operation of peripheral functions (timer, serial interface, and A/D converter) during debugging. 0: Does not stop operation 1: Stops operation
5 to 0	Reserved	—

NOTE

EPC is set by the debugger. Setting by the user program is prohibited. For the setting of the debugger, see the user's manual of the debugger.

35.4 Cautions on Using On-Chip Debugging

35.4.1 Treatment of Devices Used for Debugging

Do not install a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during system debugging and thus the write/erase endurance of the flash memory cannot be guaranteed.

35.4.2 Reset Assertion When a Debugger is connected

If a program in which a reset is asserted at the start of program execution is executed when a debugger is being used, the microcontroller is reset before preparation for communications between the OCD emulator and microcontroller is complete. For this reason, communications may not proceed correctly.

The period of preparation for communications depends on the host PC environment of the OCD emulator and the operating frequency of the microcontroller. To ensure that the debugger operates properly when debugging a program in which a reset is asserted at the start of program execution, insert a wait between the start of program and reset assertion.

35.4.3 Transition to DeepSTOP Mode When a Debugger is connected

If a program in which a transition to DeepSTOP mode occurs at the start of program execution is executed when a debugger is being used, the microcontroller stops supplying power to the isolated area before preparation for communications between the OCD emulator and microcontroller is complete. For this reason, communications may not proceed correctly.

The period of preparation for communications depends on the host PC environment of the OCD emulator and the operating frequency of the microcontroller. To ensure that the debugger operates properly when debugging a program in which a reset is asserted at the start of program execution, insert a wait between reset deassertion and transition to DeepSTOP mode.

Section 36 Flash Memory

This section describes the flash memory mounted on RH850/F1M.

The first part in this section describes the characteristics of the mounted flash memory and the characteristics specific to RH850/F1M, such as the memory map, flash memory programming, and ECC.

36.1 Features

- Includes code flash memory and data flash memory
  The code flash memory can store program codes and data and has the user area and the extended user area.
  The data flash memory is used for storing data.
- Method of flash memory programming
  Flash memory programming via a serial interface and programming of flash memory by a user program (self-programming) are supported.
- Support for BGO (Back Ground Operation)
  The BGO function allows programs to be executed in the code flash memory while the data flash memory is being programmed/erased.
- Flash memory data security
  - Support for security functions to protect against illicit tampering with or reading out of data in the flash memory
  - Support for protection functions to protect against erroneous overwriting of the flash memory
- Option byte function
  Sets the operation after releasing reset for ports, WDTA, and CVM.
- Support for the error detection/correction function (ECC) in the code flash memory and data flash memory
  Built-in ECC function can detect 2-bit errors and detect/correct 1-bit errors.
- Interrupts can be acknowledged in self-programming mode.

For details of the error detection/correction function of the code flash memory and data flash memory, see the following section.

- **Section 33.2.3, Code Flash ECC and Address Parity**
- **Section 33.2.4, Data Flash ECC**

36.2 Structure of Memory

36.2.1 Mapping of Code Flash Memory

Figure 36.1 illustrates the mapping of the code flash memory for the 4 MB device. The user area of the code flash memory of the RH850/F1M is divided into 8-Kbyte and 32-Kbyte blocks, which serve as the units of erasure. A single block of 32-Kbyte extended user area is also incorporated. The user area and extended user area are available as areas for storing the user program.

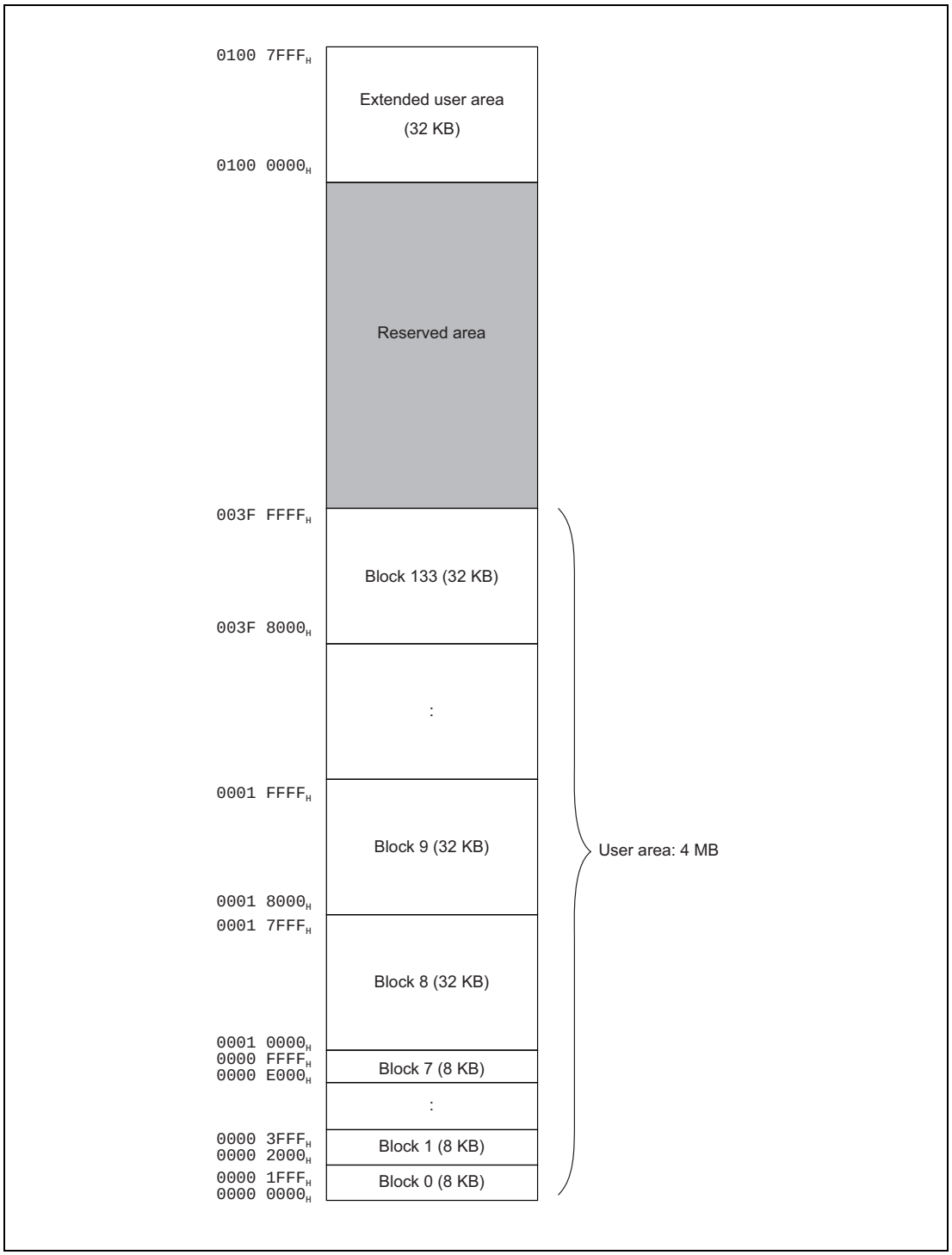


Figure 36.1 Mapping of the 4 MB Code Flash Memory

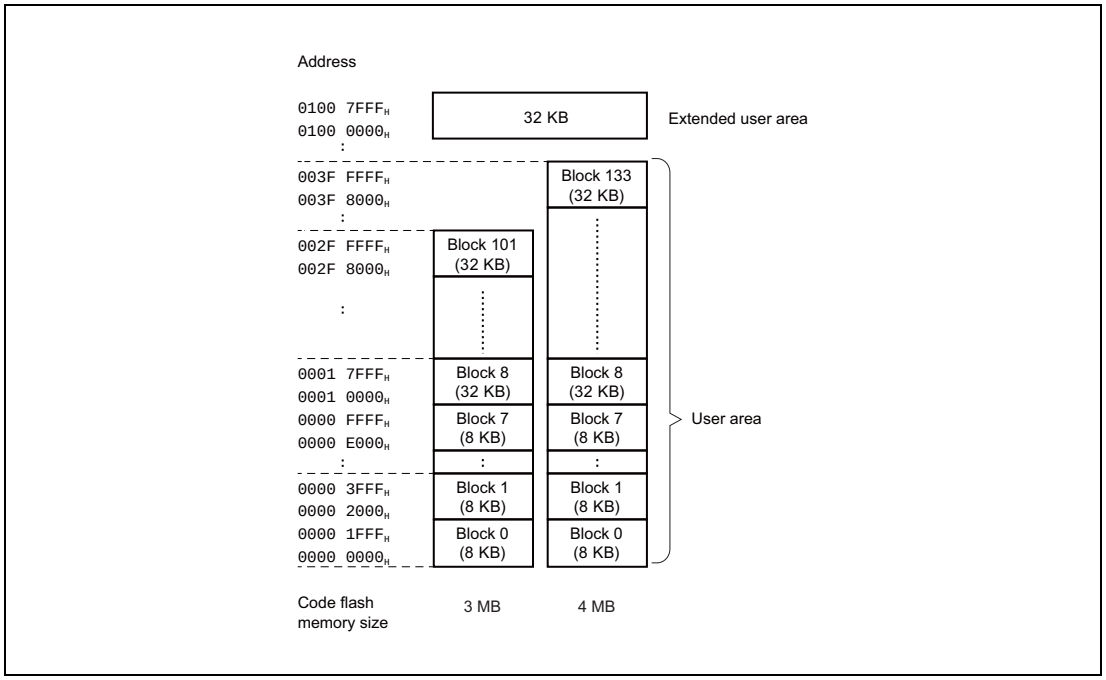


Figure 36.2 Mapping of Code Flash Memory

36.2.2 Mapping of Data Flash Memory

The data area of the data flash memory in the RH850/F1M is divided into 64-byte blocks, with each being a unit for erasure. **Figure 36.3** shows the mapping of the data flash memory.

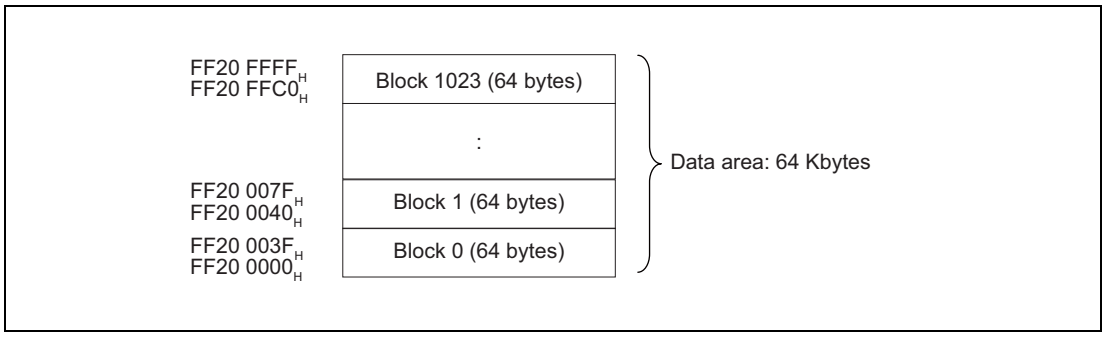


Figure 36.3 Mapping of the Data Flash Memory

36.3 Operating Modes Associated with Flash Memory

Figure 36.4 is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, see **Section 6, Operating Mode**.

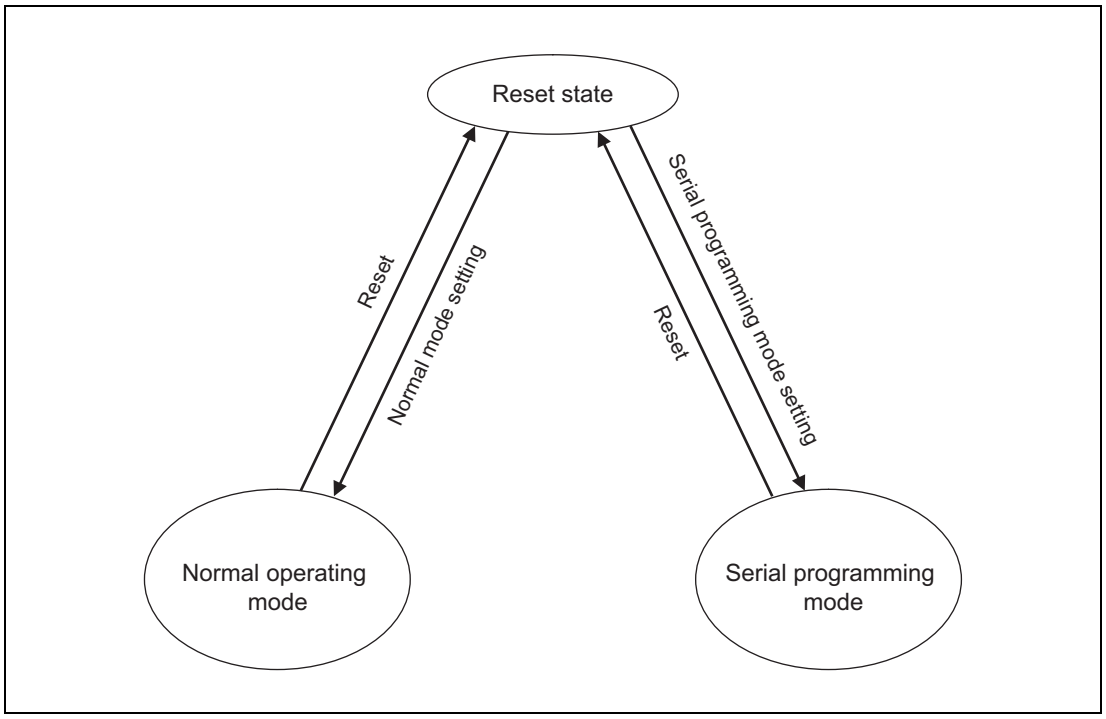


Figure 36.4 Mode Transition Associated with Flash Memory

Table 36.1 shows the flash memory area which is programmable and erasable in each mode and the boot program after reset release.

Table 36.1 Programmable and Erasable Area in Each Mode and the Boot Program after Reset Release

Item	Normal Operating Mode	Serial Programming Mode
Programmable and erasable area	• User area • Extended user area • Data area	• User area • Extended user area • Data area
Boot program after reset release	Program in user area or extended user area (Changeable by using the variable reset vector)	Firmware program for serial programming

36.4 Functions

36.4.1 Functional Overview

The flash memory of the RH850/F1M can be updated via a serial interface by a dedicated flash memory programmer (serial programming), before being mounted on the target system or on a flash adapter system.

Furthermore, security functions to prohibit updating of the user program written in the flash memory are incorporated, and this can prevent tampering by third parties.

Programming by the user program (self-programming) is suited for applications where the target system program may require updating after deployed to the end user. 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 external communications, etc., and this allows programming under various conditions. **Table 36.2** gives an overview of the methods of programming and the corresponding operating modes.

Table 36.2 Methods of Programming

Method of Programming	Overview of Functionality	Operating Mode
Serial programming	A dedicated flash memory programmer allows on-board programming of the flash memory after the device is mounted on the target system.	Serial programming mode
	A dedicated flash memory programmer and dedicated programming adapter board allow off-board programming of the flash memory, i.e. programming of the device before it is mounted on the target system.	
Self-programming	The user program that is written to code flash memory in advance by serial programming also allows updating the flash memory. The background operation capability makes it possible to fetch instructions or otherwise read data in code flash memory while the data flash memory is self-programming. For this reason, it is possible to update the data flash memory by executing a program written to the code flash memory. Instructions in the code flash memory cannot be fetched and data cannot be accessed while the code flash memory is being updated by self-programming. In such cases, a program for updating must be transferred to the local RAM or external memory in advance and executed.	Normal operating mode

Renesas provides a library for self-programming. For details on this library, see the user's manuals for the code flash library and data flash library of this device.

Table 36.3 lists the functions of the flash memory. Dedicated flash memory programmer commands enable serial programming, while reading of the flash memory by a library function or the user program enables self-programming.

Table 36.3 Basic Functions at a Glance

Function	Overview	Level of Support (√: Supported, △: Conditionally Supported, —: Not Supported)	
		Serial programming	Self-programming
Blank checking	This is used to check a specified block to ensure that writing to it has not already proceeded. Results of reading from code flash memory and data flash memory to which nothing has been written after erasure are not guaranteed, so use blank checking to confirm that writing to memory has not proceeded after erasure.	√	△ (Only data flash is supported)
Block erasure	This is for erasing the contents of a specified block of memory.	√	√
Programming	This is for writing to a specified address.	√	√
Verification and checksum	Data that are read out from flash memory are compared with data transferred from the flash memory programmer.	√	— (Reading of data by the user program is possible)
Reading	Data that have been written to the flash memory are read out.	√	√
Setting for OTP (one-time programming)	A specified block of code flash memory is set for OTP (OTP can only be set, that is, it is not possible to release a block's OTP setting).	√	√
Setting an ID	An ID setting is made for use in controlling the connection of a dedicated flash memory programmer for serial programming, controlling of the on-chip debugger, and programming of the code flash memory by self-programming.	√	√
Security settings	Security settings are for use in serial programming.	√	△ (Only when setting is prohibited after being permitted)
Protection settings	Settings for block protection of code flash memory and variable reset vector are provided.	△ (Setting of the reset vector values for variable reset vector function is not supported.)	√
Setting of option bytes	Option bytes are set to change them from the initial values for the RH850/F1M.	√	√
Clearing the configuration	ID setting, security settings, protection settings, and option byte settings are initialized.	√	—

For details on serial programming, see the user’s manual of the flash programmer.

For details on self-programming, see the user’s manuals for the code flash library and data flash library of this device.

The flash memory supports various security functions.

The OTP setting and authentication of the ID code are security functions for use with serial programming and self-programming.

In serial programming, authentication of the ID code, prohibiting connection of a dedicated flash memory programmer, and prohibition of commands (for block erasure, programming, and reading) are available for use as security functions.

The security functions supported by the flash memory are listed in **Table 36.4** and **Table 36.5**.

Table 36.4 Summary of Security Functions

Function	Description
OTP	OTP can be individually set for each block of the user area and the extended user area of code flash memory. When the OTP setting is made for an area, programming by serial programming and by self-programming is prohibited. Once set, the OTP setting cannot be released. Furthermore, since execution of the configuration clearing command is prohibited for any area for which OTP has been set, changing a security setting from “prohibited” to “permitted” is not possible.
ID authentication	The result of ID authentication can be used to control the connection of a dedicated flash memory programmer for serial programming. The result of ID authentication can also be used to control enabling of self-programming.
Prohibition of connection of a dedicated flash memory programmer	The connection of a dedicated flash memory programmer for serial programming is prohibited. Since execution of the configuration clearing command is also prohibited when the connection of a dedicated flash memory programmer is prohibited, changing a security setting from “prohibited” to “permitted” is not possible.
Prohibition of block erasure commands	Block erasure commands at the time of serial programming are prohibited. Since execution of the configuration clearing command is also prohibited when block erasure commands are prohibited, changing a security setting from “prohibited” to “permitted” is not possible.
Prohibition of programming commands	Block erasure commands and programming commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted.
Prohibition of read commands	Read commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted.

Table 36.5 Available Operations and Security Settings

Function	All Security Settings and Erasure, Programming, and Read Operations (√: Executable, —: Not Executable)		Point for Caution Regarding the Security Setting	
	Serial programming	Self-programming	Serial programming	Self-programming
OTP	• Areas for which OTP is set – Block erasure commands: — – Programming commands: — – Read commands: √ • Areas for which OTP is not set – Block erasure commands: √ – Programming commands: √ – Read commands: √	• Areas for which OTP is set – Block erasure: — – Programming: — – Reading: √ • Areas for which OTP is not set – Block erasure: √ – Programming: √ – Reading: √	• The OTP setting cannot be released. • Execution of the configuration clearing command is not possible.	The OTP setting cannot be released.
ID authentication	• When the ID codes do not match – Block erasure commands: — – Programming commands: — – Read commands: — • When the ID codes match – Block erasure commands: √ – Programming commands: √ – Read commands: √	• When the ID codes do not match – Code flash memory - Block erasure: — - Programming: — - Reading: √ – Data flash memory - Block erasure: √ - Programming: √ - Reading: √ • When the ID codes match – Block erasure: √ – Programming: √ – Reading: √	• The configuration clearing command can initialize the setting for prohibition. • The setting for prohibition of block erasure commands is not available. • The setting for prohibition of programming commands is not available. • The setting for prohibition of read commands is not available.	ID authentication is always in effect.
Prohibition of the connection of a dedicated flash memory programmer	• Block erasure commands: — • Programming commands: — • Read commands: —	• Block erasure: √ • Programming: √ • Reading: √	Since execution of the configuration clearing command is prohibited, initialization of the setting for prohibition is not possible.	Since the configuration clearing command is not supported, initialization of the setting for prohibition is not possible.
Prohibition of block erasure commands	• Block erasure commands: — • Programming commands: √ • Read commands: √	• Block erasure: √ • Programming: √ • Reading: √	• Since execution of the configuration clearing command is prohibited, initialization of the setting for prohibition is not possible. • The setting for ID authentication to be effective for serial programming is not available.	
Prohibition of programming commands	• Block erasure commands: — • Programming commands: — • Read commands: √	• Block erasure: √ • Programming: √ • Reading: √	• The configuration clearing command can initialize the setting for prohibition. • The setting for ID authentication to be effective for serial programming is not available.	
Prohibition of read commands	• Block erasure commands: √ • Programming commands: √ • Read commands: —	• Block erasure: √ • Programming: √ • Reading: √		

The flash memory supports various protection functions. The protection functions supported by the flash memory are listed in **Table 36.6**.

Table 36.6 Summary of Protection Functions

Function	Description
Block protection	Lock bit settings can be individually made to enable or disable programming and erasure of each block of the user area and the extended user area of code flash memory. Programming and erasure by self-programming of an area for which the lock bit is set and the lock bit function is enabled are prohibited. Programming or erasure can proceed again when the lock bit function is disabled after having been enabled. When a block of code flash memory is erased, the lock bit for that block is also erased.
Hardware protection	The level on the FLMD0 pin can be set to prohibit programming and erasure of the code flash memory. - FLMD0 = 0: Programming prohibited - FLMD0 = 1: Programming permitted
Variable reset vector	The protection settings include control of the reset vector. As shown in **Figure 36.5**, after programming of a new boot program while leaving the existing boot program in place, changing the reset vector is a safe way to change to the area holding the new boot program. The areas that can be specified by using the reset vector are the user area and extended user area.

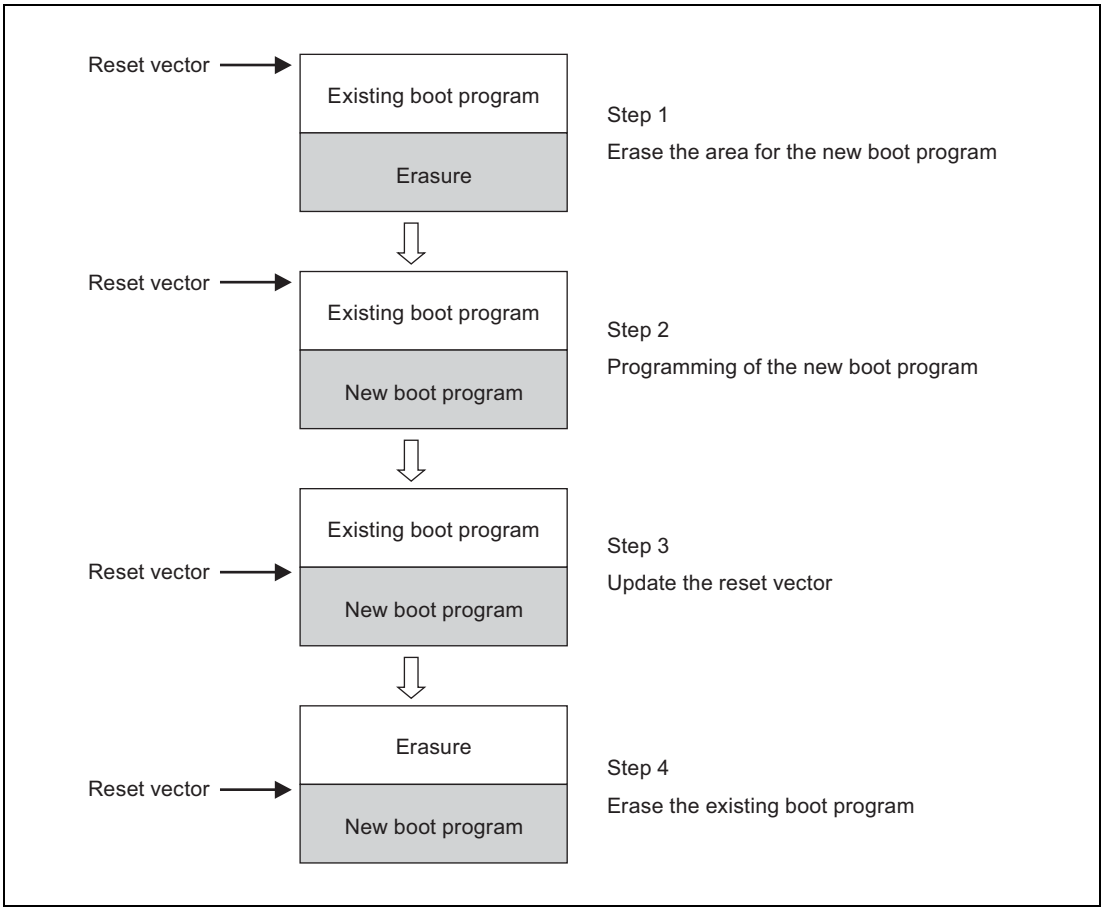


Figure 36.5 Utilizing the Variable Reset Vector Function to Update the Boot Program

NOTE

After step 4, a reset leads to updating of the reset vectors for which RBASE is changed.

36.5 Serial Programming

A dedicated flash memory programmer can be used to handle flash memory in serial programming mode.

Serial programming

The microcontroller is mounted on the system board at the time of serial programming. Providing a connector to the board enables handling of the microcontroller by the flash memory programmer to proceed.

36.5.1 Environments for Programming

The recommended environments for handling the flash memory of the microcontroller with data are described below.

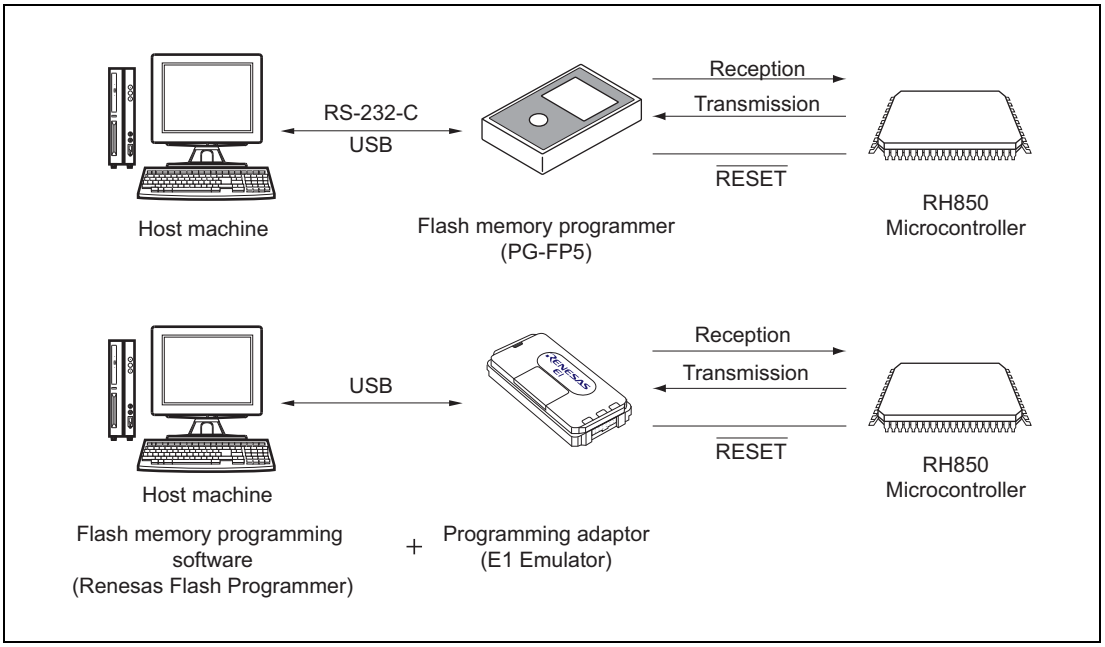


Figure 36.6 Environments for Handling Programs of the Flash Memory

By using the PG-FP5 flash memory programmer or the combination of the Renesas Flash Programmer (software for writing to flash memory) running on the host machine and the E1 emulator as an programming adaptor, the user is easily able to erase, program, and verify the contents of the on-chip memory of flash-memory-equipped microcontrollers from Renesas Electronics.

The PG-FP5 flash memory programmer handles programming from a host machine or programming in stand-alone mode while the Renesas Flash Programmer only handles programming from a host machine.

NOTE

For details on the PG-FP5, see the *PG-FP5 Flash Memory Programmer User's Manual*. For details on the Renesas Flash Programmer of flash programming software, see the *Renesas Flash Programmer Flash Programming Software User's Manual*.

36.6 Communication Modes

36.6.1 Asynchronous Flash Programming Interface - 1-Wire UART

The single-wire asynchronous serial programming interface, 1-wire UART is connected to the flash memory programmer with the following port.

- FPDR(JP0_0): Receive data input/transmit data output

36.6.2 Asynchronous Flash Programming Interface - 2-Wire UART

The double-wire asynchronous serial programming interface, 2-wire UART is connected to the flash memory programmer with the following ports.

- FPDR(JP0_0): Receive data input
- FPDT(JP0_1): Transmit data output

36.6.3 Synchronous Flash Programming Interface CSI

The synchronous serial programming interface CSI is connected to the flash memory programmer with the following ports.

- FPDR(JP0_0): Receive data input
- FPDT(JP0_1): Transmit data output
- FPCK(JP0_2): Serial clock input

The flash memory programmer outputs the serial data clock SCK, and the microcontroller operates as a slave.

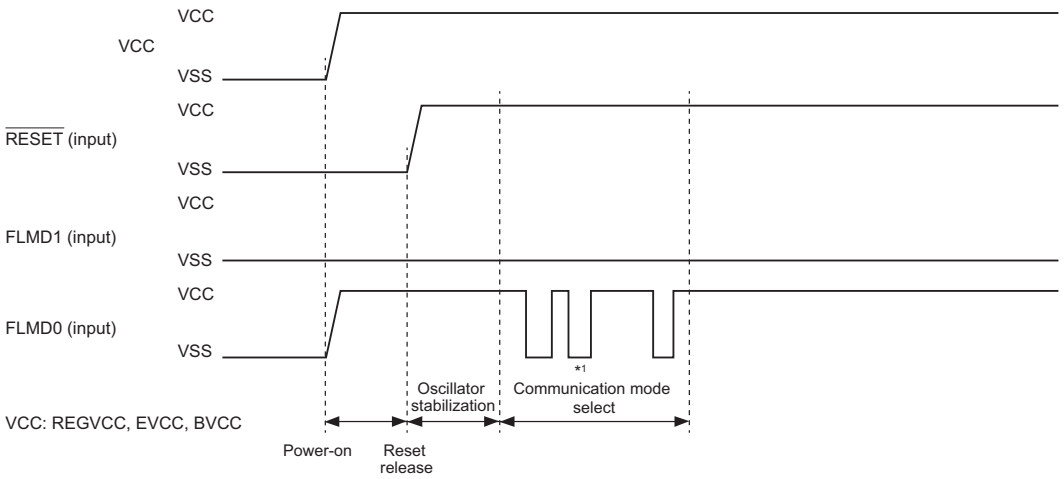
NOTE

For details on Renesas Flash Programmer, see the *Renesas Flash Programmer Flash Programming Software User's Manual*.

36.6.4 Selection of Communication Method

In RH850/F1M, communication method can be selected by pulse input to the FLMD0 pin (up to 7 pulses) after transition to the flash memory programming mode. The FLMD0 pulse is generated by a dedicated flash memory programmer.

Figure 36.7 shows the relation between the number of pulses and communication method.



Note 1. The number of clock cycles to be inserted depends on the communication method as in the following table.

FLMD0 Pulse	Communication Method
0	2-wire UART
7	1-wire UART
3	CSI
Other than above	Setting prohibited

Figure 36.7 Selection of Communication Method

36.7 Self-Programming

36.7.1 Outline

The RH850/F1M supports programming of the flash memory by the user program itself. Renesas Electronics provides a code flash library and a data flash library for use with user programs. These libraries can be used for writing to the code flash memory and to the data flash memory.

When the data flash memory is programmed, the background operation facility makes it possible to execute a programming program from the code flash memory to program the data flash memory. Furthermore, the programming program can be copied to local RAM or external memory in advance of the programming operation, and executed from the given destination to perform the programming.

The programming program can be copied to the local RAM or external memory in advance and executed to program the code flash memory.

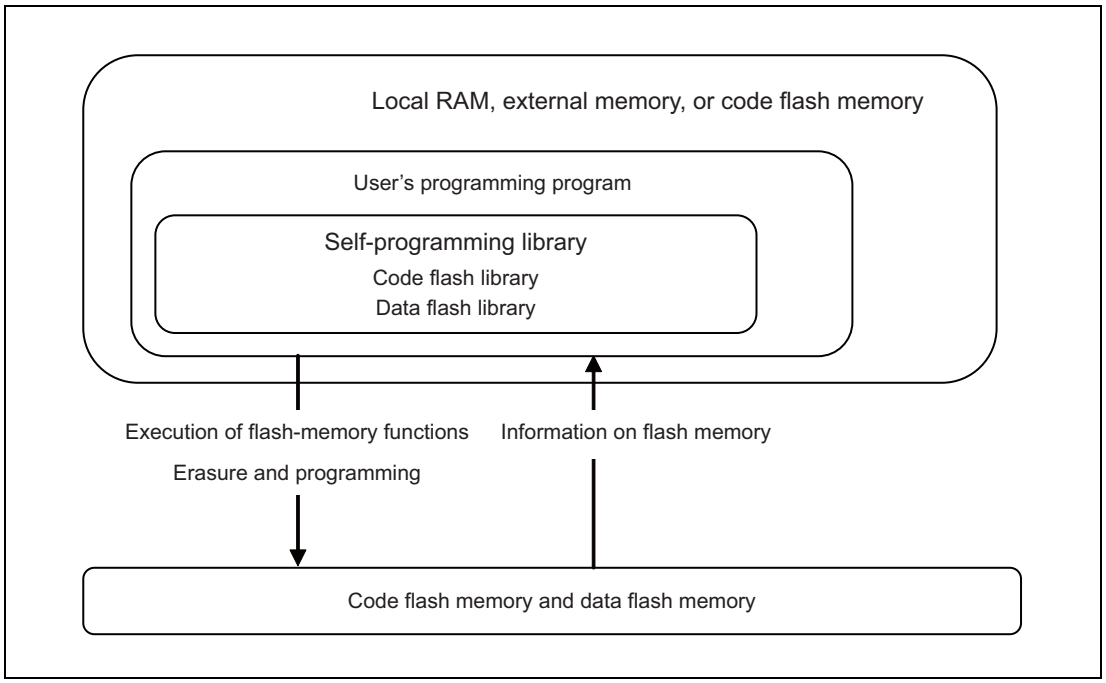


Figure 36.8 Schematic View of Self-Programming

For details on the self-programming of flash memory, see the user's manuals for the code flash and data flash libraries for this device.

36.7.2 Background Operation

Background operations can be used when the combination of the flash memory for writing and the flash memory for reading is any of those listed below.

Table 36.7 Conditions under which Background Operation is Usable

Range for Writing	Range for Reading
Data flash memory	Code flash memory

36.7.3 Enabling Self-Programming

The self-programming function can be activated in normal operating mode.

Erasure and programming of the code flash memory by the self-programming function is enabled by making the FLMD0 pin high level.

This prevents unnecessary overwriting of the program if the device operates incorrectly.

The FLMD0 pin is made high level by using one of the following methods.

- The FLMD0 pin is externally pulled up.
- The FLMD0 pin is pulled up by the FLMDCNT register.

The outline of the FLMDCNT register is described in **Section 36.7.3.1, FLMDCNT Register**.

36.7.3.1 FLMDCNT Register

This register specifies the internal pull-up or pull-down of the FLMD0 pin.

A correct write sequence that uses the FLMDPCMD register is required to update this register. For details, see **Section 5, Write-Protected Registers**.

Access: This register can be read or written in 32-bit units.

Address: FFA0 0000$_H$

Value after reset: 0000 0000$_H$

Bit	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
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FLMDPUP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 36.8 FLMDCNT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
0	FLMDPUP	FLMD0 Pin Software Control 0: Pull-down selected 1: Pull-up selected

36.8 Reading Flash Memory

36.8.1 Reading Code Flash Memory

Special settings are not required to read code flash memory in normal mode. Data can simply be read out through access to addresses in the code flash memory.

Reading from an area of code flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception.

36.8.2 Reading Data Flash Memory

Configure the number of read cycles in the EEPRDCYCL register prior to reading data from data flash memory in normal mode. Once this register is properly configured, data can be read by simply accessing addresses in the data flash memory.

Values read from data flash memory that has been erased but not yet been programmed again are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

36.8.2.1 EEPRDCYCL — Data Flash Wait Cycle Control Register

This register is used to specify the number of wait cycles to be inserted when reading the data in the data flash.

Set the number of wait cycles to be inserted in the clock cycle when reading the data flash according to the operating clock frequency of the CPU (CPUCLK).

Access: This register can be read or written in 8-bit units.

Address: FFC5 9810$_H$

Value after reset: 0F$_H$

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

Table 36.9 EEPRDCYCL Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3 to 0	WAIT[3:0]	Number of Wait Cycles

WAIT[3:0]	Number of Wait Cycles	CPU Operating Frequency		
		FCPUCLK ≤ 40 MHz	40 MHz < FCPUCLK ≤ 80 MHz	80 MHz < FCPUCLK ≤ 120 MHz
0000	1	Setting prohibited	Setting prohibited	Setting prohibited
0001	2	√	Setting prohibited	Setting prohibited
0010	3	√	√	Setting prohibited
0011	4	√	√	√
0100	5	√	√	√
0101	6	√	√	√
0110	7	√	√	√
0111	8	√	√	√
1000	9	√	√	√
Other than above	10	√	√	√

NOTES

1. The read access time to the data flash is calculated by the number of wait cycles.

   Read access time to the data flash = {28 + (Number of wait cycles × 4)} / CPU operating frequency

   However, the time may be changed depending on the combination of instructions before and after the execution.
2. √ indicates the number of wait cycles that can be set.

36.8.2.2 PRDNAMEn – Product Name Storage Register (n = 1 to 3)

This register stores the product name. The product part name is stored in 16-byte ASCII code, and PRDNAME1, PRDNAME2, and PRDNAME3 correspond to the fourth to first bytes, eighth to fifth bytes, and twelfth to ninth bytes of the product part name respectively.

Access: This register is a read-only register that can be read in 32-bit units.

Address: PRDNAME1: FFCD 00D0$_H$
PRDNAME2: FFCD 00D4$_H$
PRDNAME3: FFCD 00D8$_H$

Value after reset: See **Table 36.11**.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PRDNAMEn[31:24]*1								PRDNAMEn[23:16]*1							
Value after reset																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PRDNAMEn[15:8]*1								PRDNAMEn[7:0]*1							
Value after reset																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. n: 1 to 3

Table 36.10 PRDNAMEn Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Product name fourth byte (PRDNAME1), eighth byte (PRDNAME2), twelfth byte (PRDNAME3)
23 to 16	—	Product name third byte (PRDNAME1), seventh byte (PRDNAME2), eleventh byte (PRDNAME3)
15 to 8	—	Product name second byte (PRDNAME1), sixth byte (PRDNAME2), tenth byte (PRDNAME3)
7 to 0	—	Product name first byte (PRDNAME1), fifth byte (PRDNAME2), ninth byte (PRDNAME3)

The following table lists the product information-related registers.

Table 36.11 List of Product Information-Related Registers

Product Part Name	PRDNAME1	PRDNAME2	PRDNAME3
R7F701544	3746 3752	3435 3130	2020 2034
R7F701545	3746 3752	3435 3130	2020 2035
R7F701548	3746 3752	3435 3130	2020 2038
R7F701549	3746 3752	3435 3130	2020 2039
R7F701552	3746 3752	3535 3130	2020 2032
R7F701553	3746 3752	3535 3130	2020 2033
R7F701564	3746 3752	3635 3130	2020 2034
R7F701565	3746 3752	3635 3130	2020 2035
R7F701568	3746 3752	3635 3130	2020 2038
R7F701569	3746 3752	3635 3130	2020 2039
R7F701572	3746 3752	3735 3130	2020 2032
R7F701573	3746 3752	3735 3130	2020 2033

36.9 Option Bytes

The option bytes of the flash memory are an expansion area and hold data specified by the user for a variety of purposes. Initial settings for peripheral modules and so on as specified by the option bytes become effective on release from the reset state.

36.9.1 Option Byte Setting

Be sure to set the option byte area that corresponds to the optional functions listed below, before writing a program to the flash memory.

The optional functions specified by the option bytes are as follows.

- Function of port group JP0
- Activation code method of WDTA1
- Start mode of WDTA1
- Enabling or disabling WDTA1
- Activation code method of WDTA0
- Start mode of WDTA0
- Enabling or disabling WDTA0
- Initial value of the overflow interval time for WDTA0 and WDTA1
- Enabling the high voltage monitor
- Enabling the low voltage monitor

36.9.2 OPBT0 — Option Byte 0

The settings and bit positions of the option bytes are listed below.

For details on how to set an option byte, refer to the flash programming user's manual for serial programming as well as the code flash library for self-programming.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	OPJTAG[1:0]		—	—	WDT 1_3	—	WDT 1_1	WDT 1_0	WDT 0_3	—	WDT 0_1	WDT 0_0	WDT_2	WDT_1	WDT_0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CVM_HD_EN	CVM_LD_EN	—	—	—	—

Table 36.12 Option Byte 0 Settings

Bit Position	Bit Name	Function
31	Reserved	When writing, write “1”.
30, 29	OPJTAG[1:0]	These bits control the function of port group JP0. 00: JP0 is used for general purpose/alternative function port. 01: JP0 is used for LPD (4-pin). 10: JP0 is used for LPD (1-pin). 11: JP0 is used for Nexus I/F.
28, 27	Reserved	When writing, write “1”.
26	WDT1_3	Specifies the activation code method of WDTA1. 0: Fixed activation code 1: Variable activation code
25	Reserved	When writing, write “1”.
24	WDT1_1	Specifies the start mode of WDTA1. 0: Software trigger start mode 1: Default start mode
23	WDT1_0	Enables or disables WDTA1. 0: WDTA1 is disabled 1: WDTA1 is enabled
22	WDT0_3	Specifies the activation code method of WDTA0. 0: Fixed activation code 1: Variable activation code
21	Reserved	When writing, write “1”.
20	WDT0_1	Specifies the start mode of WDTA0. 0: Software trigger start mode 1: Default start mode
19	WDT0_0	Enables or disables WDTA0. 0: WDTA0 is disabled 1: WDTA0 is enabled
18 to 16	WDT_[2:0]	Control of the overflow interval time for WDTA0 and WDTA1 These bits specify the reset value of WDTAnMD.WDTAnOVF[2:0].
15 to 11	Reserved	When writing, write “1”.
10	Reserved	When writing, write “0”.
9 to 6	Reserved	When writing, write “1”.
5	CVM_HD_EN	High Voltage Monitor Enable 0: Disable high voltage detection 1: Enable high voltage detection
4	CVM_LD_EN	Low Voltage Monitor Enable 0: Disable low voltage detection 1: Enable low voltage detection
3 to 0	Reserved	When writing, write “1”.

36.9.3 OPBT1 — Option Byte 1

The settings and bit positions of the option bytes are listed below.

For details on how to set an option byte, refer to the flash programming user's manual for serial programming as well as the code flash library for self-programming.

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

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

Table 36.13 Option Byte 1 Setting

Bit Position	Bit Name	Function
31 to 10	Reserved	When writing, write “1”.
9	Reserved	When writing, write “0”.
8 to 4	Reserved	When writing, write “1”.
3	Reserved	When writing, write “0”.
2	Reserved	When writing, write “1”.
1, 0	Reserved	When writing, write “0”.

36.10 Usage Notes

(1) Reading areas where programming or erasure was interrupted

When programming or erasure of an area of flash memory is interrupted, the data stored in the area become undefined. To avoid the reading out of undefined data, which might cause a malfunction, take care not to fetch instructions or read data from areas where programming or erasure was interrupted.

(2) Reading the code flash memory that has been erased but not yet been programmed again

Note that reading from an area of code flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception. Use blank checking when you need to confirm that an area is in the non-programmed state.

(3) Prohibition of additional writing

Writing to a given area two or more times is not possible. When overwriting data in an area of flash memory after writing to the area has been completed, erase the area first.

(4) Resets during programming and erasure

In the case of an external reset during programming and erasure, wait for at least the minimum value of $\overline{\text{RESET}}$ input low level width once the operating voltage is within the range stipulated in the electrical characteristics after assertion of the reset signal before releasing the device from the reset state.

(5) Allocation of vectors for interrupts and other exceptions during programming and erasure

Generation of an interrupt or other exception during programming or erasure may lead to fetching of the vector from the code flash memory. If this does not satisfy the conditions for using background operation, set the address for vector fetching to an address that is not in the code flash memory.

(6) Abnormal termination of programming and erasure

Even if programming/erasure ends abnormally due to the assertion of a reset by the $\overline{\text{RESET}}$ pin, the programming/erasure state of the flash memory with undefined data cannot be verified or checked. For the area where programming/erasure ends abnormally, the blank check function cannot judge whether the area is erased successfully or not. Erase the area again to ensure that the corresponding area is completely erased before using.

If programming and erasure of code flash memory are not completed normally, the lock bit for the target area may be enabled (locked). In such cases, erase the block to erase the lock bit while the lock bit is in the disabled state (the area is not locked).

(7) Items prohibited during programming and erasure

Do not perform the following operations during programming and erasure.

- Set the operating voltage from the power supply outside the allowed range.
- Change the frequency of the peripheral clock.

Section 37 RAM

This section describes the RAM mounted on RH850/F1M.

37.1 Features

- RH850/F1M includes the following RAMs:
  - Local RAM
    The local RAM is accessible at high speed. Values in the local RAM are not retained in DeepSTOP mode.
  - Retention RAM
    Values in the retention RAM are retained in DeepSTOP mode.
    In addition, even if the power-supply voltage (REGVCC) falls below the POC voltage, data in the retention RAM are retained as long as the voltage does not fall below the RAM retention voltage (V_{VLVI}).

Access time for each RAM is shown in the table below.

Table 37.1 RAM Access Time (Access from CPU1)

Access Type	RAM	1st Access (CPUCLK)	Continue Access (CPUCLK)
Instruction fetch	Local RAM	9	9
	Retention RAM	13	13
Read access	Local RAM	2	1
	Retention RAM	12	1
Write access	Local RAM	2	1
	Retention RAM	12	1

NOTE

There is possibility that number of access clock of above table is changed depending on the combination before and after instructions.
When RAM access is misaligned, these number is increased.

- The local RAM and retention RAM contain error detection/correction function (ECC). The ECC function is included, which can detect 2-bit errors and detect/correct 1-bit errors. For details, see **Section 33, Functional Safety**.

37.2 Memory Configuration

Figure 37.1 shows the memory map of the local RAM and retention RAM.

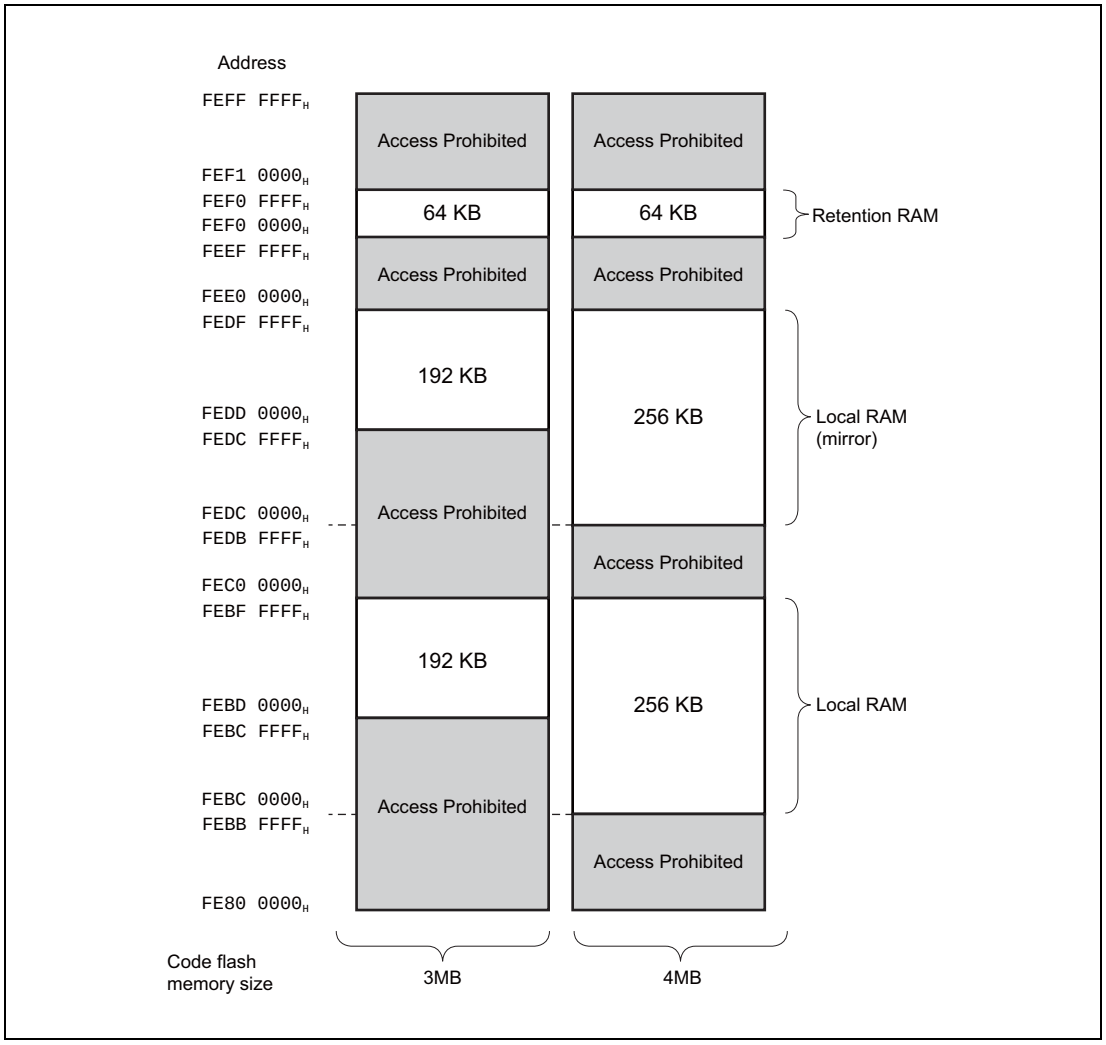


Figure 37.1 Memory Map of the Local RAM and Retention RAM

37.3 Usage Notes

Before accessing the local RAM and retention RAM with ECC error detection and correction enabled, initialize the RAM by setting the access size to the largest bit length.

Accessing the RAM before initializing it may lead to the detection of ECC errors. ECC errors may be detected if initialization is not handled in the maximum unit of access, for example, when 32-bit word RAM is accessed in 8- or 16-bit units.

Section 38 Boundary Scan

This section contains a generic description of boundary scan.

The RH850/F1M has a JTAG interface and provides a boundary scan function.

38.1 Overview

Boundary scan is a test method defined in the IEEE standard 1149.1 that is used to test the connection between the devices mounted on the printed-circuit board. The boundary scan of the RH850/F1M conforms to IEEE Std 1149.1-2001.

38.2 Features

- Five control signals (DCUTCK, DCUTDI, DCUTDO, DCUTMS, and $\overline{\text{DCUTRST}}$)
- TAP controller
- Instruction register
- Bypass register
- Boundary scan register

The JTAG interface has four instruction modes.

- BYPASS
  Test mode conforming to the IEEE 1149.1
- EXTEST
  Test mode conforming to the IEEE 1149.1
- SAMPLE/PRELOAD
  Test mode conforming to the IEEE 1149.1
- IDCODE
  Test mode conforming to the IEEE 1149.1

Figure 38.1 shows a block diagram of the JTAG interface.

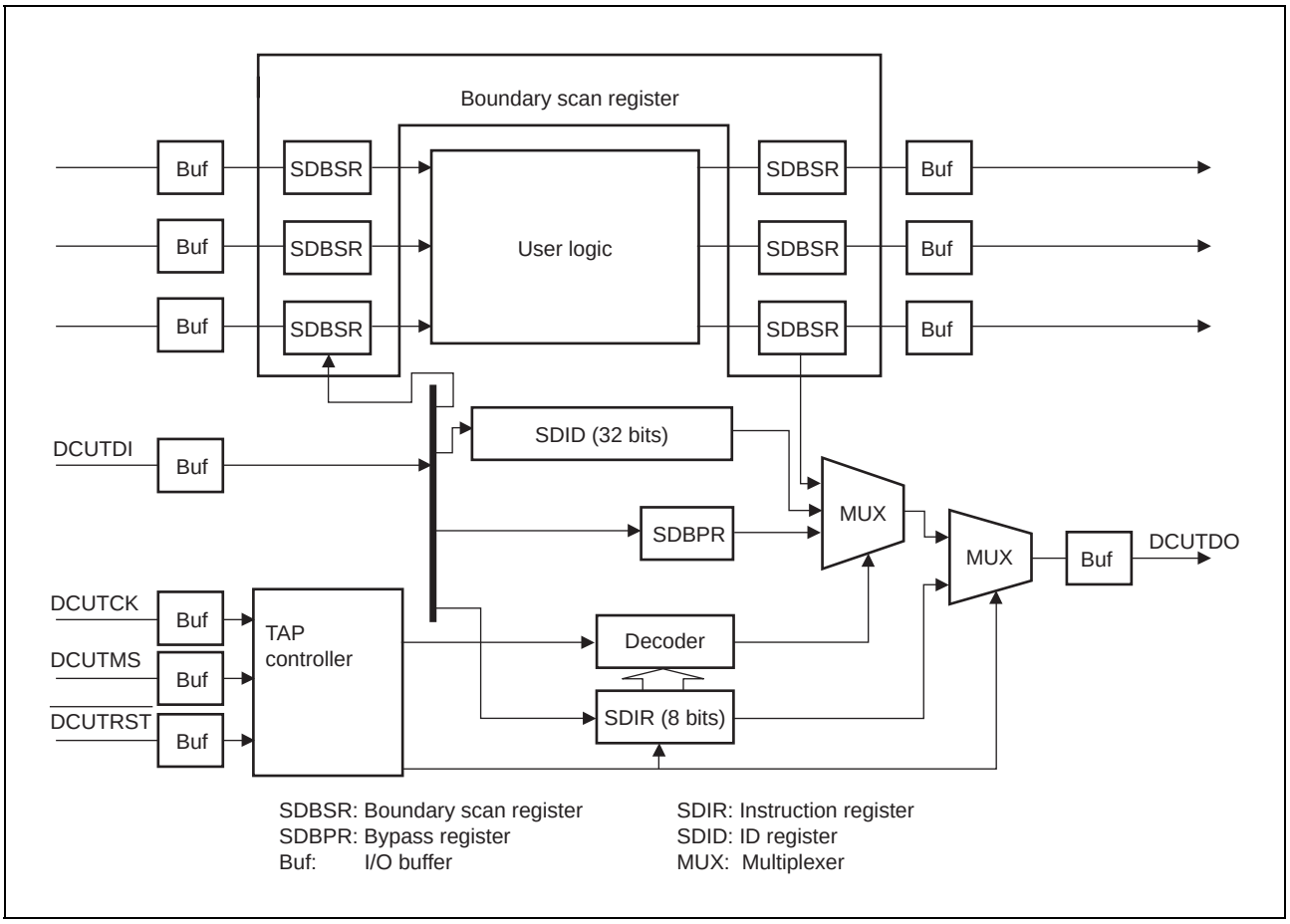


Figure 38.1 Block Diagram of JTAG Interface

38.3 External Input/Output Pins

There are five JTAG control signals: DCUTCK, DCUTDI, DCUTMS, DCUTDO, and $\overline{\text{DCUTRST}}$.

Table 38.1 shows the pin configuration.

Table 38.1 Pin Configuration

Pin Name	Description
DCUTCK	Serial data input/output clock pin Data is input to DCUTDI and is output from DCUTDO in synchronization with this clock signal.
DCUTMS	Mode select input pin Changing the level of this signal in synchronization with DCUTCK changes the state of the TAP controller. For the protocol, see **Figure 38.2, TAP Controller State Transition Diagram**.
$\overline{\text{DCUTRST}}$	Reset input pin A low-level input of this signal resets the JTAG interface. This signal is accepted asynchronously with DCUTCK.
DCUTDI	Serial data input pin Data is input in synchronization with DCUTCK and sent to the JTAG interface.
DCUTDO	Serial data output pin Data to be read from the JTAG interface is output in synchronization with DCUTCK.

38.4 Register Descriptions

The JTAG interface has the following registers. None of the registers can be accessed by the CPU.

- SDIR: Instruction register
- SDID: ID register
- SDBPR: Bypass register
- SDBSR: Boundary scan register

Table 38.2 Register Configuration

Register Name	Symbol	Access Size	Initial Value*1
Instruction register	SDIR	8	55_H
ID register	SDID	32	*2
Bypass register	SDBPR	1	Undefined
Boundary scan register	SDBSR	—	Undefined

Note 1. Registers are initialized when $\overline{\text{DCUTRST}}$ pin is 0 or when TAP is in the Test-Logic-Reset state.
Note 2. The initial value differs depending on the device. Please contact our sales representative for details.

Instructions can be serially transferred from the serial data input pin (DCUTDI) and input to the instruction register (SDIR). The bypass register (SDBPR) is a 1-bit register, to which DCUTDI and DCUTDO are connected in BYPASS mode. The boundary scan register (SDBSR) is connected to DCUTDI and DCUTDO in SAMPLE/PRELOAD mode and EXTEST mode. The ID code register (SDID) is a 32-bit register, from which the ID code is output via DCUTDO in IDCODE mode.

Table 38.3 shows the serial transfer types possible with the JTAG interface registers.

Table 38.3 Serial Transfer Types

Register	Serial Input	Serial Output
SDIR	Possible	Impossible*1
SDBPR	Possible	Possible
SDBSR	Possible	Possible
SDID	Impossible	Possible

Note 1. A fixed value is read out.

38.4.1 Instruction Register (SDIR)

SDIR is an 8-bit register that holds a boundary scan instruction. SDIR is initialized by a low-level input of $\overline{\text{DCUTRST}}$ or in the TAP Test-Logic-Reset state. Operation is not guaranteed if a reserved instruction is set in this register.

Table 38.4 Boundary Scan Instructions

Instruction Code								Description
0	0	0	0	0	0	0	0	JTAG EXTEST
0	1	0	0	0	0	0	0	JTAG SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	JTAG IDCODE (initial value)
1	1	1	1	1	1	1	1	JTAG BYPASS
Other than above								Reserved

38.4.2 ID Register (SDID)

SDID is a 32-bit register with a device specific ID.

SDID can be read from the JTAG interface when the IDCODE instruction is set, but cannot be accessed from the CPU.

For the read values, see **Table 38.2, Register Configuration**.

38.4.3 Bypass Register (SDBPR)

SDBPR is a 1-bit register. When SDIR is set to BYPASS mode, SDBPR is connected to the position between DCUTDI and DCUTDO. The initial value is undefined. SDBPR is not initialized by a power-on reset or by a low level input of $\overline{\text{DCUTRST}}$.

38.4.4 Boundary Scan Register (SDBSR)

SDBSR is a shift register for controlling the external Input/Output pins. When SDIR is set to SAMPLE/PRELOAD or EXTEST mode, SDBSR is connected to the position between DCUTDI and DCUTDO. The initial value is undefined. SDBSR is not initialized by a power-on reset or a low-level input of $\overline{\text{DCUTRST}}$.

38.5 Operation

38.5.1 TAP Controller

Table 38.2 shows the state transition of the TAP controller. Transition is triggered by the DCUTMS value at the rising edge of DCUTCK.

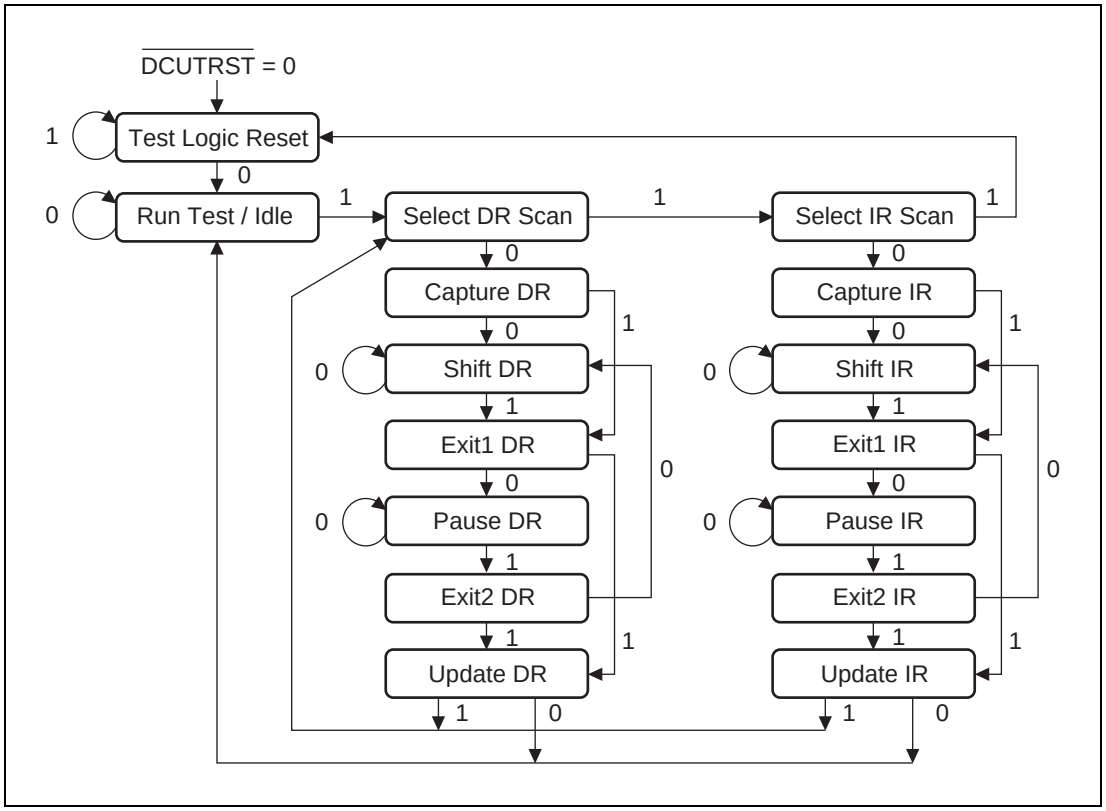


Figure 38.2 TAP Controller State Transition Diagram

NOTE

The DCUTDI value is sampled at the rising edge of DCUTCK and is shifted at the falling edge. DCUTDO is in the high-impedance state in the states other than Shift-DR and Shift-IR. A low-level input of $\overline{\text{DCUTRST}}$ causes transition to Test-Logic-Reset state asynchronously with DCUTCK.

38.5.2 Supported Instructions

38.5.2.1 BYPASS

The BYPASS instruction is a standard instruction indispensable to bypass register operation. This instruction shortens the shift path to achieve high-speed serial data transfer of other devices on the printed-circuit board. During execution of this instruction, the test circuit has no effect on the system circuit.

38.5.2.2 SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input the value to the boundary scan register from the internal circuits of this device; to output the value from the scan path; and to load data onto the scan path. During execution of this instruction, the level of the input pin of this device is sent to the internal circuits as is, and the value of the internal circuits is output to the outside via the output pin as is. Executing this instruction has no effect on the system circuit of this device.

The SAMPLE operation allows taking in the snapshots of the value to be transferred to the internal circuits from the input pin or the value to be transferred to the output pin from the internal circuits to the boundary scan register and allows reading the snapshots from the scan path. Snapshots can be taken in without preventing the normal operation of this device.

The PRELOAD operation allows setting the initial value to the parallel output latch of the boundary scan register from the scan path prior to the EXTEST instruction. If the EXTEST instruction is executed without PRELOAD operation, an undefined value is output from the output pin until the first scan sequence is completed (transfer to the output latch) because the parallel output latch value is always output to the output pin by the EXTEST instruction.

38.5.2.3 EXTEST

The EXTEST instruction is used to test the external circuits when this device is mounted on the printed-circuit board. When this instruction is executed, the output pin is used to output the test data (previously set by the SAMPLE/PRELOAD instruction) from the boundary scan register to the printed-circuit board; whereas the input pin is used to take in the test result from the printed-circuit board to the boundary scan register. When the EXTEST instruction is executed N times for testing, the test data for the Nth execution is scanned in at the (N - 1)th scan-out.

If the data is loaded onto the boundary scan register of the output pin in the Capture-DR state of this instruction, it is not used for testing the external circuits (replaced through shift operation).

38.5.2.4 IDCODE

The IDCODE instruction sets the JTAG interface pins to IDCODE mode, which is defined by the JTAG standard. When the JTAG interface is initialized (by a low-level input of $\overline{\text{DCUTRST}}$ or placing TAP in the Test-Logic-Reset state), IDCODE mode is set.

38.5.3 Pins Subject to Boundary Scan

All pins, excluding pins such as external clock input pins or power supply pins ,are subject to boundary scan.

The pins which are not subjected to boundary scan are listed in **Table 38.5**.

Table 38.5 Pins not Subject to Boundary Scan

Type	Pins
JTAG interface	DCUTCK, DCUTDI, DCUTDO, DCUTMS, $\overline{\text{DCUTRST}}$
Power supply pins	REGVCC, AWOVCL, AWOVSS, ISOVCL, ISOVSS EVCC, BVCC, EVSS, BVSS
Power supply pins (A/D converter)	A0VREF, A1VREF, A0VSS, A1VSS
Clock signals	X1, X2, XT1, IP0_0/XT2
MODE	P10_8/FLMD1, P10_1/MODE0, P10_2/MODE1

The following signals are only sampled in boundary scan mode.

Table 38.6 Pins Subject to Boundary Scan (Sampling Only)

Function	Pin Name
Reset	$\overline{\text{RESET}}$
MODE	FLMD0

The following pins are shared by the analog buffer. Accordingly, boundary scan only applies to general I/O pins.

Table 38.7 Pins Subject to Boundary Scan (Only General I/O Pins)

Function	Pin Name
ADCA0 input	P8_0-12, P9_0-4, AP0_0-15
ADCA1 input	P18_0-15, P19_0-3, AP1_0-15

NOTE

In boundary scan mode, the level of the following pins must be fixed:
P10_1: Low, P10_2: High, P10_8: High

38.6 Usage Notes

1. Once an instruction is set, it is not modified until another instruction is issued. To issue the same instruction twice in a row, insert an instruction that has no effect on chip operation (such as BYPASS) between the instructions.
2. To start the system in boundary scan mode, de-assert $\overline{\text{DCUTRST}}$ while $\overline{\text{RESET}}$ is high. Also be sure to set DCUTMS to high before de-asserting $\overline{\text{DCUTRST}}$ and ensure that DCUTMS remains high for 600 ns + five DCUTCK clock cycles after de-asserting $\overline{\text{DCUTRST}}$.
3. For the maximum clock frequency that can be input to DCUTCK, see the Electrical Characteristics section in the Data Sheet document.
4. If serial transfer is performed exceeding the number of bits of the register connected between DCUTDI and DCUTDO, the data that is input from DCUTDI is output from DCUTDO as is.
5. If the serial transfer sequence is corrupted, be sure to assert $\overline{\text{DCUTRST}}$. In this case, transfer starts again from the beginning regardless of the point of transfer corruption.
6. Data is output via DCUTDO at the falling edge of DCUTCK.
7. To facilitate debugging, route $\overline{\text{DCUTRST}}$ on the board in such a way that patterns can be easily cut.

Section 39 Power Supply and Power Domains

This section describes the power supply and power domains of the RH850/F1M.

39.1 Function

The internal circuits are separated into two independent power domains, the Always-On area (AWO) and the Isolated area (ISO).

The power supply of the Always-On area is always powered in all operating modes and stand-by modes.

The power supply of the Isolated area can be turned off to reduce the overall power consumption depending on the types of stand-by mode.

For each power domain, a dedicated on-chip voltage regulator generates the internal supply voltage.

For operation of the device, the following voltages are required:

- Power supply voltage REGVCC for the on-chip voltage regulators. The output voltage of the voltage regulators is supplied to the digital circuits in each power domain.
- Power supply voltages EVCC and BVCC for I/O port.
- Power supply voltages A0VREF and A1VREF for the A/D converters and the dedicated I/O ports.

39.1.1 Power Supply Pins

The table below lists all power supply pins and what they are used for.

Table 39.1 Power Supply Pins

Power Supply	Power Supply Pins	Power Supply for
Power supply for internal circuits	REGVCC	• On-chip voltage regulators for the Always-On area and Isolated area • Port group IP0 • MainOSC • SubOSC • POC / LVI
	AWOVCL*1	
	AWOVSS	
	ISOVCL*1	
	ISOVSS	
Power supply for I/O port	EVCC	$\overline{\text{RESET}}$, FLMD0 (233 pin device) • Port groups JP0, P0, P1, P2, P3, P8, P9, P20 (176 pin device) • Port groups JP0, P0, P1, P2, P8, P9, P20 (144 pin device) • Port groups JP0, P0, P1, P8, P9, P20
	EVSS	
	BVCC	(233 pin device) • Port groups P10, P11, P12, P13, P18, P19 (176 pin device) • Port groups P10, P11, P12, P18 (144 pin device) • Port groups P10, P11, P12, P18
	BVSS	
Power supply for A/D converters	A0VREF	• Analog circuits of ADCA0, port group AP0
	A0VSS	
	A1VREF	• Analog circuits of ADCA1, port group AP1
	A1VSS	

Note: Refer to the Electrical Characteristics in the data sheet for the voltage range of each power supply.

Note 1. Pin to connect a stabilization capacitor for on-chip voltage regulator.

39.1.2 Block Diagram of Power Domains

The figure below shows the overview of power supply circuit.

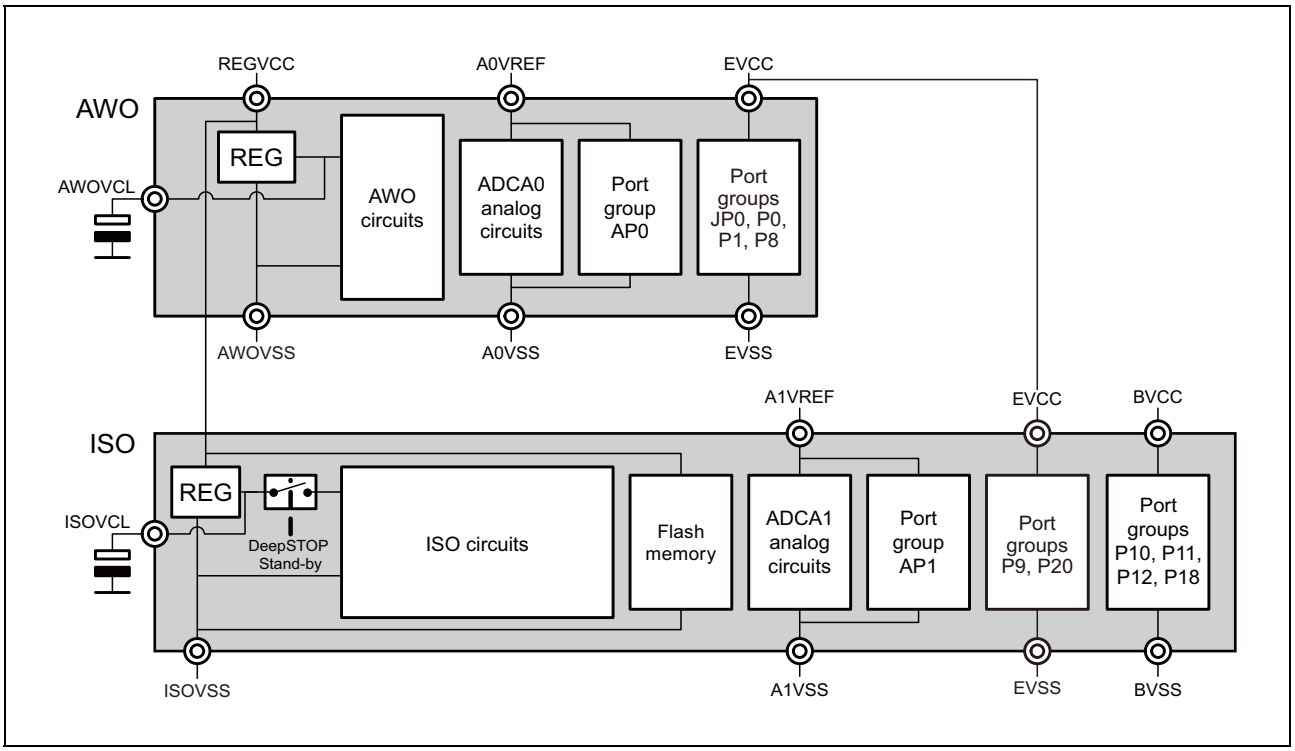


Figure 39.1 Overview of Power Supply Circuit (144 pin)

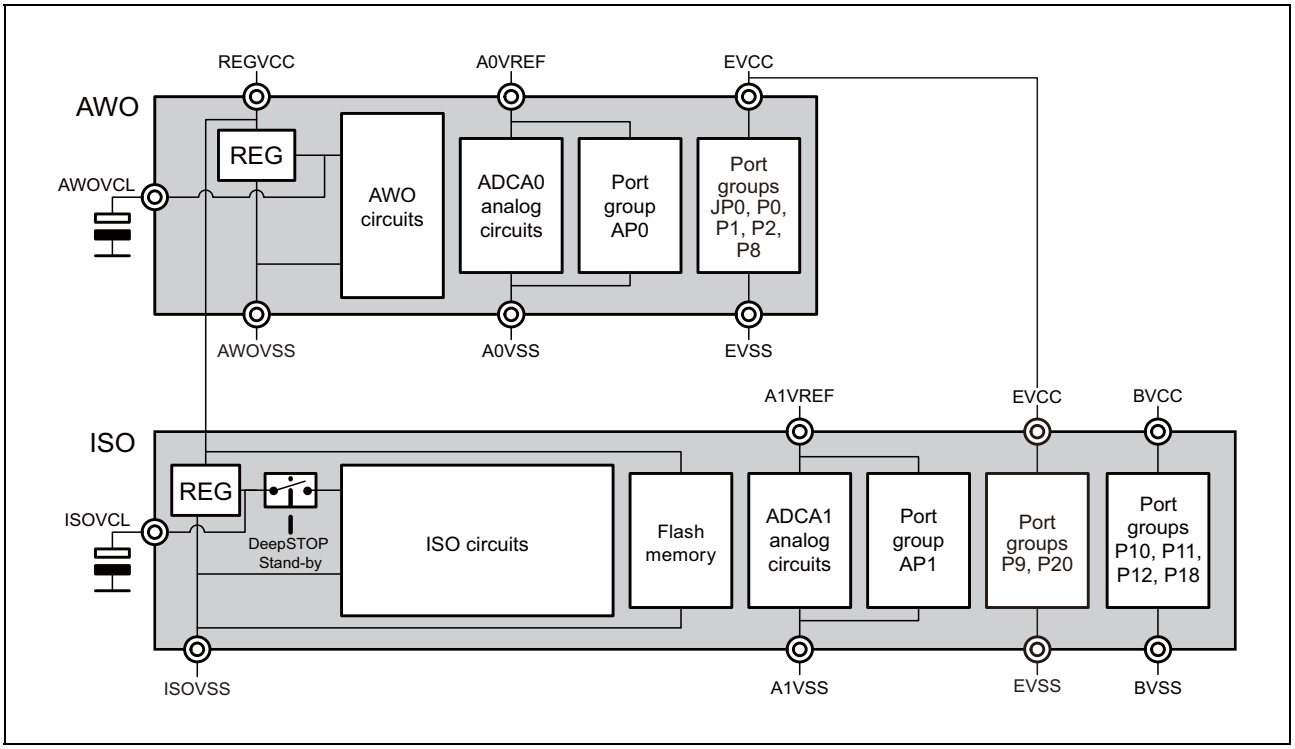


Figure 39.2 Overview of Power Supply Circuit (176 pin)

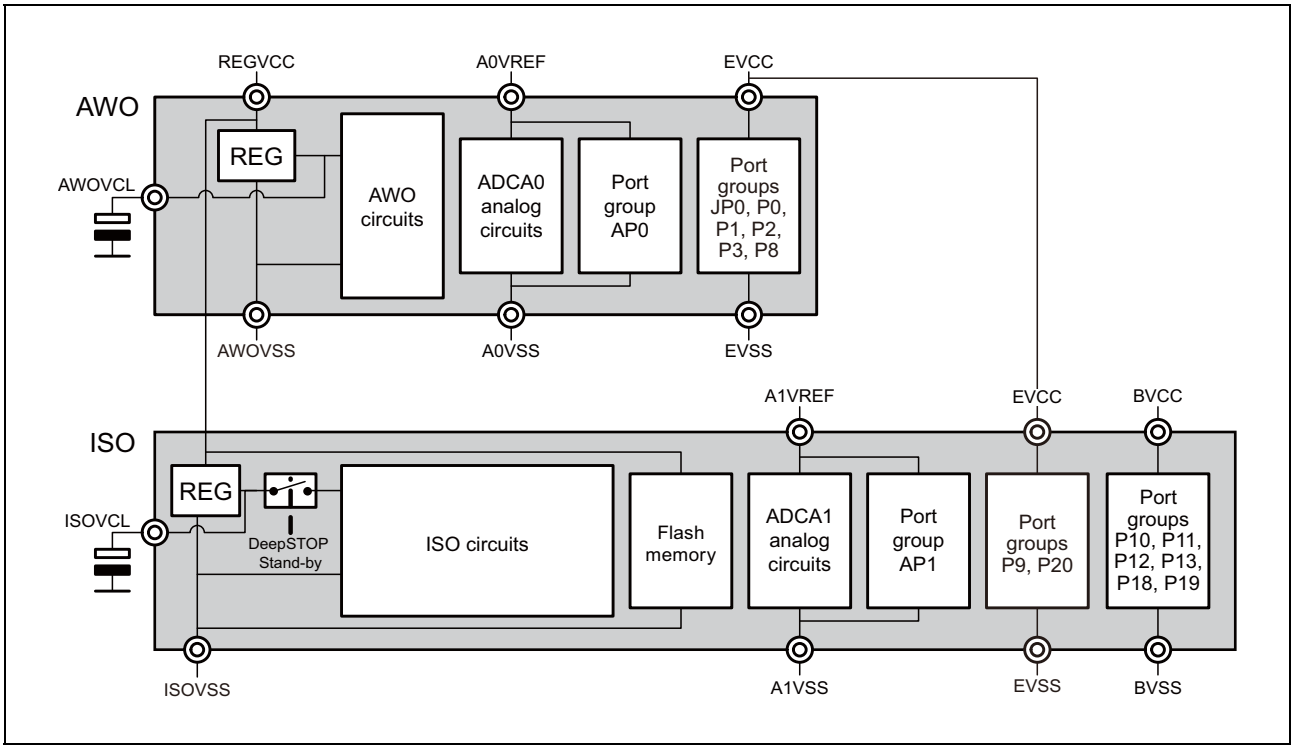


Figure 39.3 Overview of Power Supply Circuit (233 pin)

39.1.3 Power Domains Arrangement

The table below lists the microcontroller functional modules for each power domain.

Table 39.2 Functional Modules and Power Domain

Power Domain	Functions
AWO area	• STBC, Reset controller • Retention RAM • MainOSC, SubOSC, LS IntOSC, HS IntOSC, CLMA0, CLMA1 • WDTA0, RTCAn, TAUJ0, ADCA0, LPS • Port groups JP0, P0, P1, P2, P3, P8, AP0, IP0
ISO area	• CPU subsystem • Code flash, Data flash, Local RAM • PLL0, PLL1, CLMA2 • WDAT1, DCRAn, TAUDn, TAUBn, TAUJ1, OSTMn, PWM-Diag, CSIGn, CSIHn, RSCANn, RLIN2m, RLIN3n, RIICn, ADCA1, Motor Control, ENCAn, KRn, FLXAn • Port groups P9, P10, P11, P12, P13, P18, P19, P20, AP1

RH850/F1M User's Manual: Hardware

Publication Date: Rev.0.10 Nov 13, 2014
Rev.1.03 May 23, 2016

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL II Stage, Indiranagar, Bangalore, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 4.0

RH850/F1M

Renesas Electronics Corporation

R01UH0518EJ0103