User's Manual

32

RH850/F1L Group

User's Manual: Hardware

Renesas microcontroller
RH850 Family

Preliminary

Renesas Electronics
www.renesas.com

Rev.1.10 Jun, 2014

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.

(2012.4)

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/F1L and design application systems using the following RH850/F1L microcontrollers:

Purpose

This manual is intended to give users an understanding of the hardware functions of the RH850/F1L shown in the *Organization* below.

Organization

This manual is divided into two parts: Hardware (this manual) and Architecture (RH850 Family 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/F1L.

→ Read this manual according to the Contents.

To understand the details of an instruction function

→ See RH850 Family User's Manual: Software (R01US0058E) available separately.

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 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 are described below.

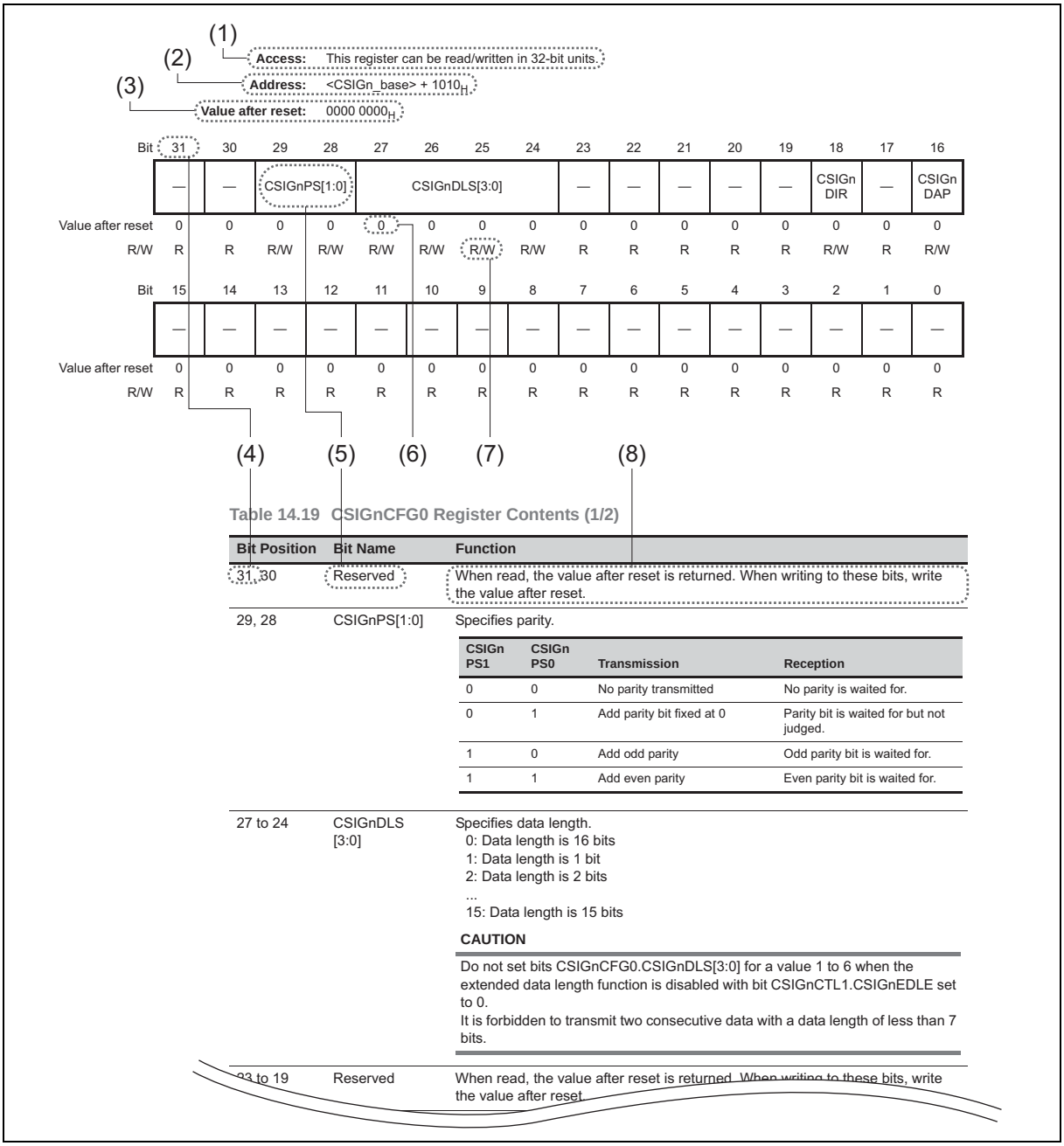


Table 14.19 CSIGnCFG0 Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
29, 28	CSIGnPS[1:0]	Specifies parity.

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.

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] for a 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 to these bits, write the value after reset.

(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.
When clearly identifying the digits of a bit field is required, do so by using a form such as CSIGnDLS[3:0] above.
Indicate reserved bits by using a dash (—).

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

This is the bit values 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/F1L Products Features

The features of the RH850/F1L are shown below.

The RH850/F1L is a 32-bit single-chip microcontroller with a G3K core. The key features of the F1L 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 is available. For example, there are Low Power Sampler (LPS), where analog and digital inputs are polled for input signals to input pins without CPU core interaction, and Deepstop mode, where power supply to the most circuits of the microcontroller can be turned off.

Applications

Applications include automotive electronics, such as BCM (body control module), gateway, HVAC, lighting modules, and many others.

1.1.1 RH850/F1L Functions

Table 1.1 Overview of Product (1/2)

Series name			RH850/F1L					
			48 pin	64 pin	80 pin	100 pin	144 pin	176 pin
Memory			See **Table 1.2, Product Lineup**.					
External memory interface (MEMC)			Not provided					Provided
CPU	CPU System		G3K					
	CPU frequency		80 MHz max. (ECO)*2	80 MHz max. (ECO/Gateway-512 KB)*2 96 MHz max. (ADVANCED/PREMIUM/Gateway-1 MB)				
	System Register Protection Unit (SRP)		Provided					
	Memory Protection Unit (MPU)		Provided					
DMA			16 channels					
Operating clock	Main Oscillator (Main Osc)		8 to 24 MHz					
	Low Speed Internal Oscillator (LS IntOsc)		240 kHz					
	High Speed Internal Oscillator (HS IntOsc)		8 MHz					
	PLL		Provided					
	Sub Oscillator (Sub Osc)		Not provided				32.768 kHz	
I/O port			33	49	65	81	120	150
A/D converter	ADC0	Physical input channels	Total 12 ch (12 bit resolution: 8 ch + 10 bit resolution: 4 ch)	Total 21 ch (12 bit resolution: 10 ch + 10 bit resolution: 11 ch)	Total 25 ch (12 bit resolution: 11 ch + 10 bit resolution: 14 ch)	Total 36 ch (12 bit resolution: 16 ch + 10 bit resolution: 20 ch)	Total 36 ch (12 bit resolution: 16 ch + 10 bit resolution: 20 ch)	Total 36 ch (12 bit resolution: 16 ch + 10 bit resolution: 20 ch)
		External multiplexer support for channel number extension	Provided					
		Channels with T&H	3			6		
	ADC1	Physical input channels	Not provided				Total 12 ch (12 bit resolution: 8 ch + 10 bit resolution: 4 ch)	Total 24 ch (12 bit resolution: 16 ch + 10 bit resolution: 8 ch)
		External multiplexer support for channel number extension	Not provided					
		Channels with T&H	Not provided					
Timer	Timer Array Unit D (TAUD)		1 unit (16 bit resolution timers × 16 channels /unit)					
	Timer Array Unit B (TAUB)		Not provided			1 unit (16 bit resolution timers × 16 channels /unit)		2 units (16 bit resolution timers × 16 channels / unit)
	Timer Array Unit J (TAUJ)		1 unit (32 bit resolution timers × 4 channels /unit)			2 units (32 bit resolution timers × 4 channels /unit)		
	Operating System Timer (OSTM)		1 unit					
	Real-Time Counter (RTCA)		Not provided				1 unit	
	Encoder Timer (ENCA)		1 unit					
	Window Watchdog Timer A (WDTA)		2 units					

Table 1.1 Overview of Product (2/2)

Series name		RH850/F1L					
		48 pin	64 pin	80 pin	100 pin	144 pin	176 pin
Serial interfaces	Clocked Serial Interface G (CSIG)	1 channel				2 channels	
	Clocked Serial Interface H (CSIH)	1 channel		3 channels	4 channels		
	CAN Interface (RS-CAN)	1 channel (Total 80 message buffers)	3 channels (Total 240 message buffers)	3 channels (Total 240 message buffers)	3 channels (Total 240 message buffers) for except F1L Gateway 6 channels (Total 480 message buffers) for F1L Gateway	4 channels (Total 320 message buffers) for Code Flash size 1 MB and less 6 channels (Total 480 message buffers) for Code Flash size 1.5 MB and more	6 channels (Total 480 message buffers)
	LIN/UART Interface (RLIN3)	1 channel	2 channels	3 channels	4 channels	6 channels	
	LIN Master Interface (RLIN2)	2 channels			3 channels	6 channels	10 channels
	I^2C Interface (RIIC)	1 channel					
External Interrupts	Maskable	8		12	13	16	
	Non-maskable (NMI)	1					
Other functions	Clock Monitors (CLMA)	For PLL, HS IntOsc, MainOsc					
	Data CRC (DCRA)	1 channel		4 channels			
	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, CSIH, RS-CAN					
	Low Power Sampling (LPS)	Provided					
	PWM diagnosis (PWM_DIAG)	13 channels	24 channels		48 channels	64 channels	72 channels
	Motor Control	1 unit					
	Key Return (KR)	6 channels	8 channels				
	CLOCK OUTPUT (FOUT)	Provided					
	RESET OUTPUT (RESETOUT)	Provided					
	ICUSB	Not provided	Not provided (ECO/Gateway-512 KB)*2 Provided (ADVANCED/PREMIU /Gateway-1 MB)				
	On-Chip debug (OCD)	Provided					
	Boundary Scan	Provided					
Voltage supply	Internal supply	V_{POC}*1 to 5.5 V					
	Input/output buffer supplies	V_{POC}*1 to 5.5 V					
	A/D Converter supplies	3.0 to 5.5 V					
Package		48 pin LQFP	64 pin LQFP	80 pin LQFP	100 pin LQFP	144 pin LQFP	176pin LQFP

Note 1. V_{POC} is the voltage level of power-on-clear circuit.
Refer to the data sheet for detail specification of electrical values.

Note 2. ECO (products in the eco range) is the standard product operating at 80 MHz.
ADVANCED (products in the advanced range) is the product having an ICUSB(*) and its CPU performance is enhanced to operate at 96 MHz.
PREMIUM (Products in the premium range) is the product having an ICUSB(*) and 64-Kbyte data flash memory and its CPU performance is enhanced to operate at 96 MHz.
Gateway-512KB (512-Kbyte Gateway products) is the 100-pin product having six CAN channels.
Gateway-1MB (1-Mbyte Gateway products) is the 100-pin product having six CAN channels and an ICUSB(*), and its CPU performance is enhanced to operate at 96 MHz.
"F1L for Gateway" is the general term for both Gateway-512KB and Gateway-1MB.

*ICUSB: Intelligent Cryptographic Unit

Table 1.2 Product Lineup (1/2)

Pin Count	CPU frequency	Memory					CAN Interface (RS-CAN)	Product Name			Line Name
		Code Flash	Local RAM (Primary)	Local RAM (Secondary)	Data Flash	Retention RAM (RRAM)		Operating Temperature (Ta)			
								–40°C to +85°C	–40°C to +105°C	–40°C to +125°C	
48 pins	80 MHz max.	256 KB	—	Not provided	32 KB	32 KB	1 channel (Total 80 message buffers)	R7F7010082AFP	R7F7010083AFP	R7F7010084AFP	ECO
		384 KB	16 KB					R7F7010092AFP	R7F7010093AFP	R7F7010094AFP	ECO
		512 KB	32 KB					R7F7010102AFP	R7F7010103AFP	R7F7010104AFP	ECO
64 pins	80 MHz max.	256 KB	—	Not provided	32 KB	32 KB	3 channels (Total 240 message buffers)	R7F7010112AFP	R7F7010113AFP	R7F7010114AFP	ECO
		384 KB	16 KB					R7F7010122AFP	R7F7010123AFP	R7F7010124AFP	ECO
		512 KB	32 KB					R7F7010132AFP	R7F7010133AFP	R7F7010134AFP	ECO
		768 KB	64 KB					R7F7010142AFP	R7F7010143AFP	R7F7010144AFP	ECO
		1024 KB	96 KB					R7F7010152AFP	R7F7010153AFP	R7F7010154AFP	ECO
80 pins	80 MHz max.	256 KB	—	Not provided	32 KB	32 KB	3 channels (Total 240 message buffers)	R7F7010162AFP	R7F7010163AFP	R7F7010164AFP	ECO
		384 KB	16 KB					R7F7010172AFP	R7F7010173AFP	R7F7010174AFP	ECO
		512 KB	32 KB					R7F7010182AFP	R7F7010183AFP	R7F7010184AFP	ECO
		768 KB	64 KB					R7F7010192AFP	R7F7010193AFP	R7F7010194AFP	ECO
		1024 KB	96 KB					R7F7010202AFP	R7F7010203AFP	R7F7010204AFP	ECO
100 pins	80 MHz max.	256 KB	—	Not provided	32 KB	32 KB	3 channels (Total 240 message buffers)	R7F7010212AFP	R7F7010213AFP	R7F7010214AFP	ECO
		384 KB	16 KB					R7F7010222AFP	R7F7010223AFP	R7F7010224AFP	ECO
		512 KB	32 KB					R7F7010232AFP	R7F7010233AFP	R7F7010234AFP	ECO
		768 KB	64 KB					R7F7010242AFP	R7F7010243AFP	R7F7010244AFP	ECO
		1024 KB	96 KB					R7F7010252AFP	R7F7010253AFP	R7F7010254AFP	ECO
		512 KB	32 KB				6 channels (Total 480 message buffers) for F1L Gateway	R7F7010022AFP	R7F7010023AFP	R7F7010024AFP	Gateway-512KB
	96 MHz max.	1024 KB	96 KB					R7F7010032AFP	R7F7010033AFP	R7F7010034AFP	Gateway-1MB
144 pins	80 MHz max.	384 KB	16 KB	Not provided	32 KB	32 KB	4 channels (Total 320 message buffers)	R7F7010262AFP	R7F7010263AFP	R7F7010264AFP	ECO
		512 KB	32 KB					R7F7010272AFP	R7F7010273AFP	R7F7010274AFP	ECO
		768 KB	64 KB					R7F7010282AFP	R7F7010283AFP	R7F7010284AFP	ECO
		1024 KB	96 KB					R7F7010292AFP	R7F7010293AFP	R7F7010294AFP	ECO
		1536 KB	128 KB				6 channels (Total 480 message buffers)	R7F7010302AFP	R7F7010303AFP	R7F7010304AFP	ECO
		2048 KB	128 KB	32 KB				R7F7010062AFP	R7F7010063AFP	R7F7010064AFP	ECO

Table 1.2 Product Lineup (2/2)

Pin Count	CPU frequency	Memory					CAN Interface (RS-CAN)	Product Name			Line Name
		Code Flash	Local RAM (Primary)	Local RAM (Secondary)	Data Flash	Retention RAM (RRAM)		Operating Temperature (Ta)			
								–40°C to +85°C	–40°C to +105°C	–40°C to +125°C	
176 pins	80 MHz max.	768 KB	64 KB	Not provided	32 KB	32 KB	6 channels (Total 480 message buffers)	R7F7010322AFP	R7F7010323AFP	R7F7010324AFP	ECO
		1024 KB	96 KB					R7F7010332AFP	R7F7010333AFP	R7F7010334AFP	ECO
		1536 KB	128 KB					R7F7010342AFP	R7F7010343AFP	R7F7010344AFP	ECO
		2048 KB	128 KB	32 KB				R7F7010072AFP	R7F7010073AFP	R7F7010074AFP	ECO
64 pins	96 MHz max.	768 KB	64 KB	Not Provided	32 KB	32 KB	3channels (Total 240 message buffers)	R7F7010402AFP	R7F7010403AFP	R7F7010404AFP	ADVANCED
		1024 KB	96 KB					R7F7010412AFP	R7F7010413AFP	R7F7010414AFP	ADVANCED
80 pins	96 MHz max.	768 KB	64 KB	Not Provided	32 KB	32 KB	3channels (Total 240 message buffers)	R7F7010422AFP	R7F7010423AFP	R7F7010424AFP	ADVANCED
		1024 KB	96 KB					R7F7010432AFP	R7F7010433AFP	R7F7010434AFP	ADVANCED
100 pins	96 MHz max.	768 KB	64 KB	Not Provided	32 KB	32 KB	3channels (Total 240 message buffers)	R7F7010442AFP	R7F7010443AFP	R7F7010444AFP	ADVANCED
		1024 KB	96 KB					R7F7010452AFP	R7F7010453AFP	R7F7010454AFP	ADVANCED
144 pins	96 MHz max.	768 KB	64 KB	Not Provided	32 KB	32 KB	4channels (Total 320 message buffers)	R7F7010462AFP	R7F7010463AFP	R7F7010464AFP	ADVANCED
		1024 KB	96 KB					R7F7010472AFP	R7F7010473AFP	R7F7010474AFP	ADVANCED
		1536 KB	128 KB				6channels (Total 480 message buffers)	R7F7010482AFP	R7F7010483AFP	R7F7010484AFP	ADVANCED
		2048 KB	128 KB	32 KB				R7F7010492AFP	R7F7010493AFP	R7F7010494AFP	ADVANCED
176 pins	96 MHz max.	768 KB	64 KB	Not Provided	32 KB	32 KB	6channels (Total 480 message buffers)	R7F7010502AFP	R7F7010503AFP	R7F7010504AFP	ADVANCED
		1024 KB	96 KB					R7F7010512AFP	R7F7010513AFP	R7F7010514AFP	ADVANCED
		1536 KB	128 KB					R7F7010522AFP	R7F7010523AFP	R7F7010524AFP	ADVANCED
		2048 KB	128 KB	32 KB				R7F7010532AFP	R7F7010533AFP	R7F7010534AFP	ADVANCED
144pins	96 MHz max.	1536 KB	128 KB	Not Provided	64 KB	32 KB	6channels (Total 480 message buffers)	R7F7010542AFP	R7F7010543AFP	R7F7010544AFP	PREMIUM
		2048 KB	128 KB	32 KB				R7F7010552AFP	R7F7010553AFP	R7F7010554AFP	PREMIUM
176pins	96 MHz max.	1536 KB	128 KB	Not Provided	64 KB	32 KB	6channels (Total 480 message buffers)	R7F7010562AFP	R7F7010563AFP	R7F7010564AFP	PREMIUM
		2048 KB	128 KB	32 KB				R7F7010572AFP	R7F7010573AFP	R7F7010574AFP	PREMIUM

1.1.2 Internal Block Diagram

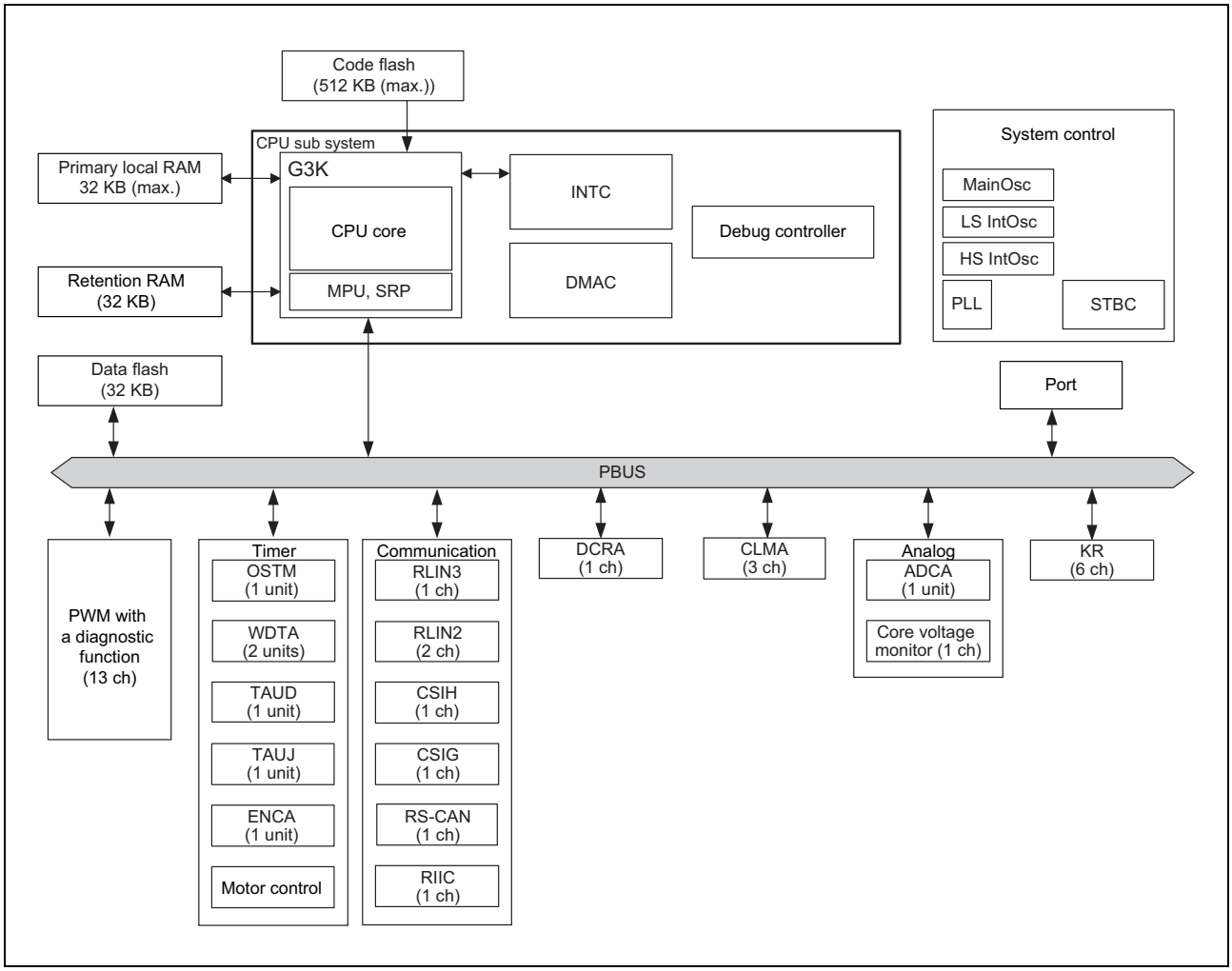
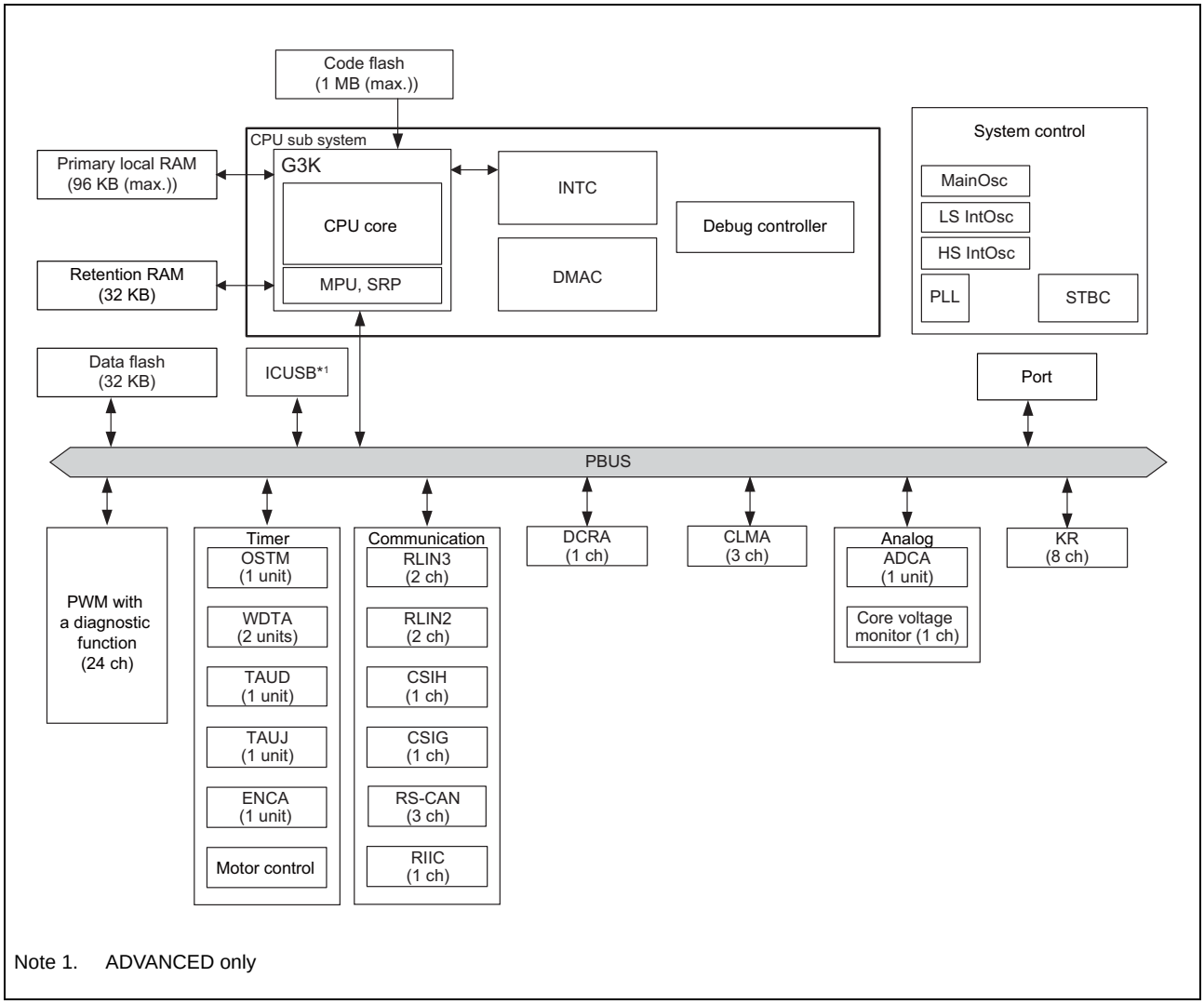
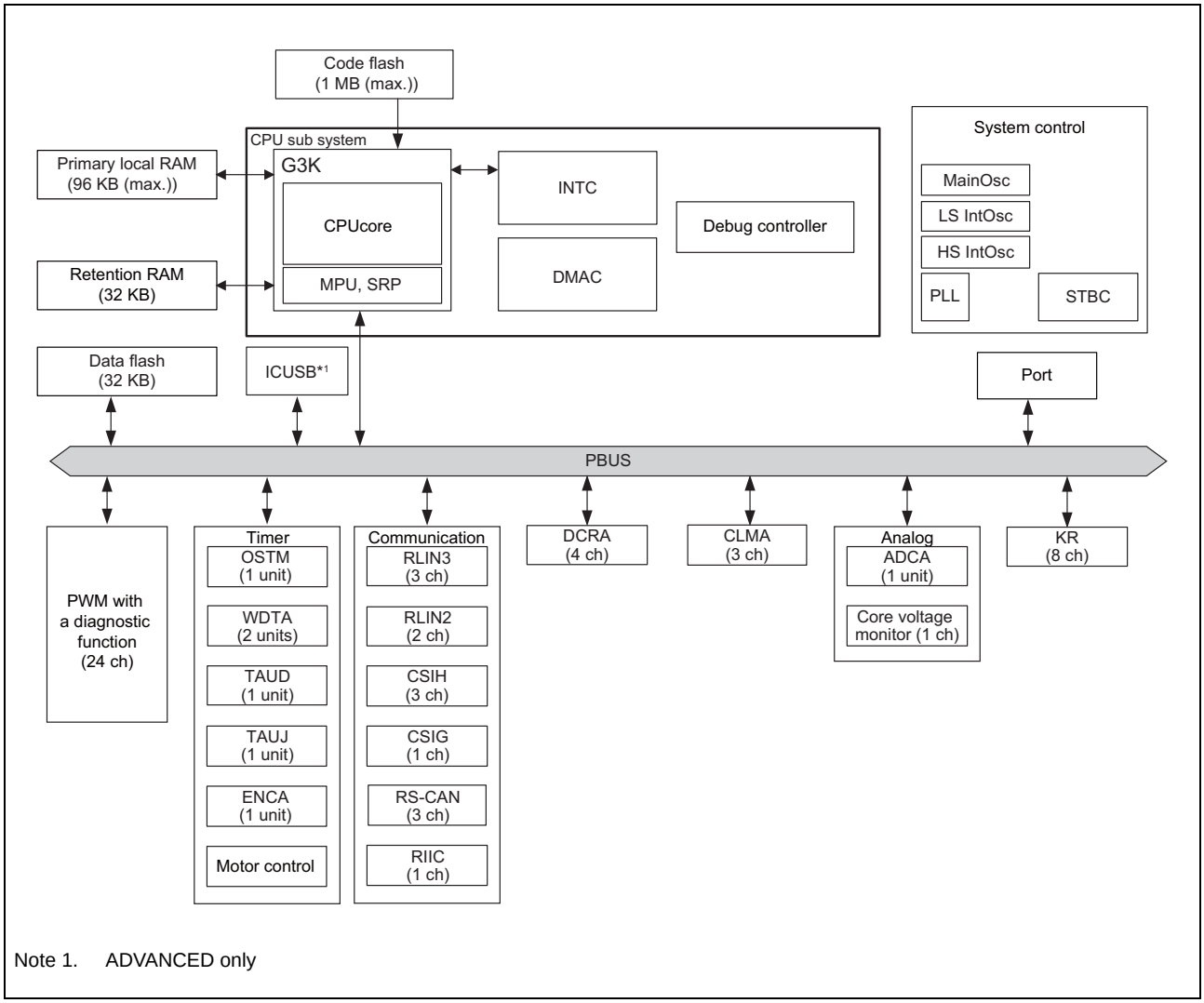


Figure 1.1 Internal Block Diagram (RH850/F1L 48-pin)



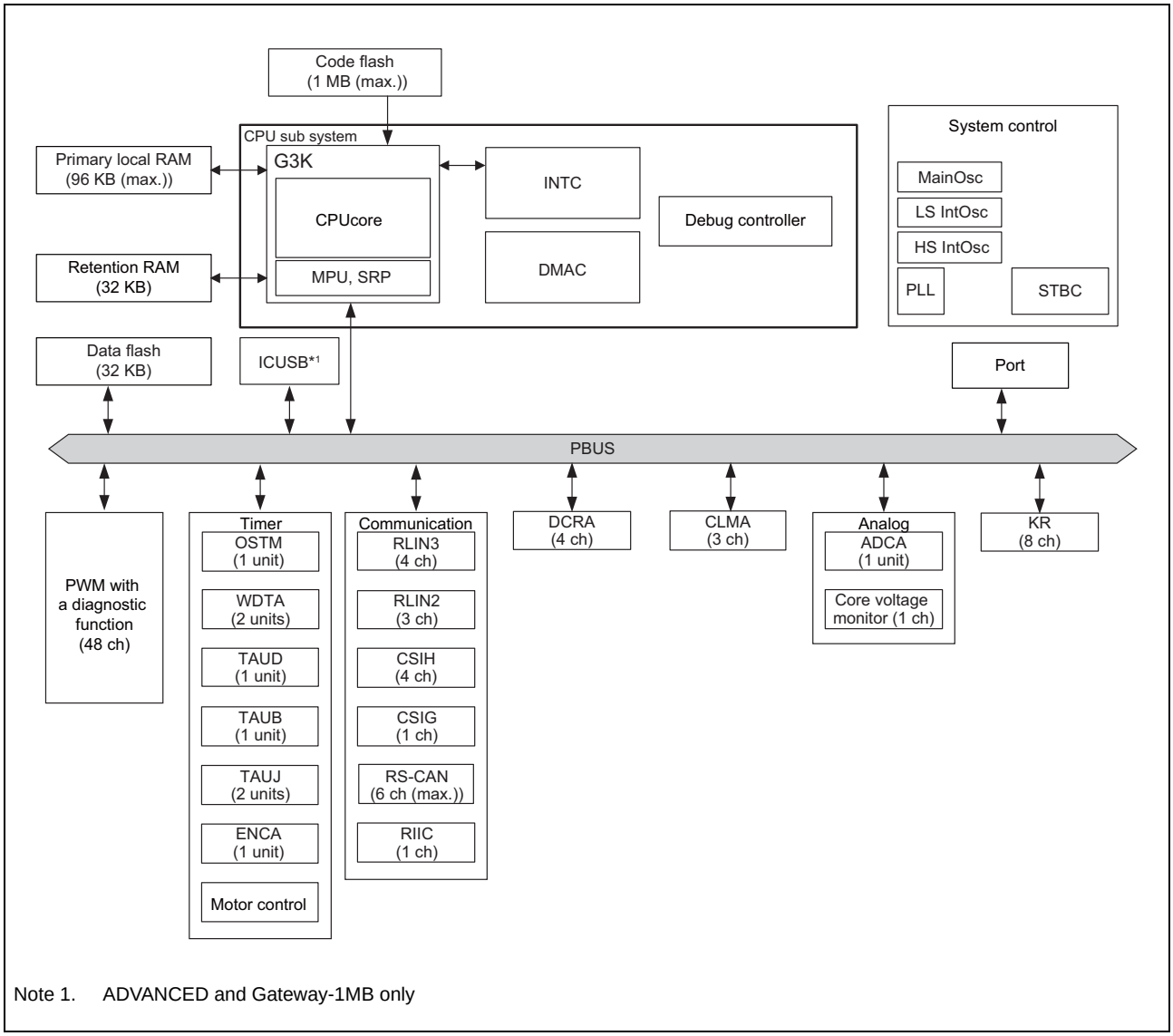
Note 1. ADVANCED only

Figure 1.2 Internal Block Diagram (RH850/F1L 64-pin)



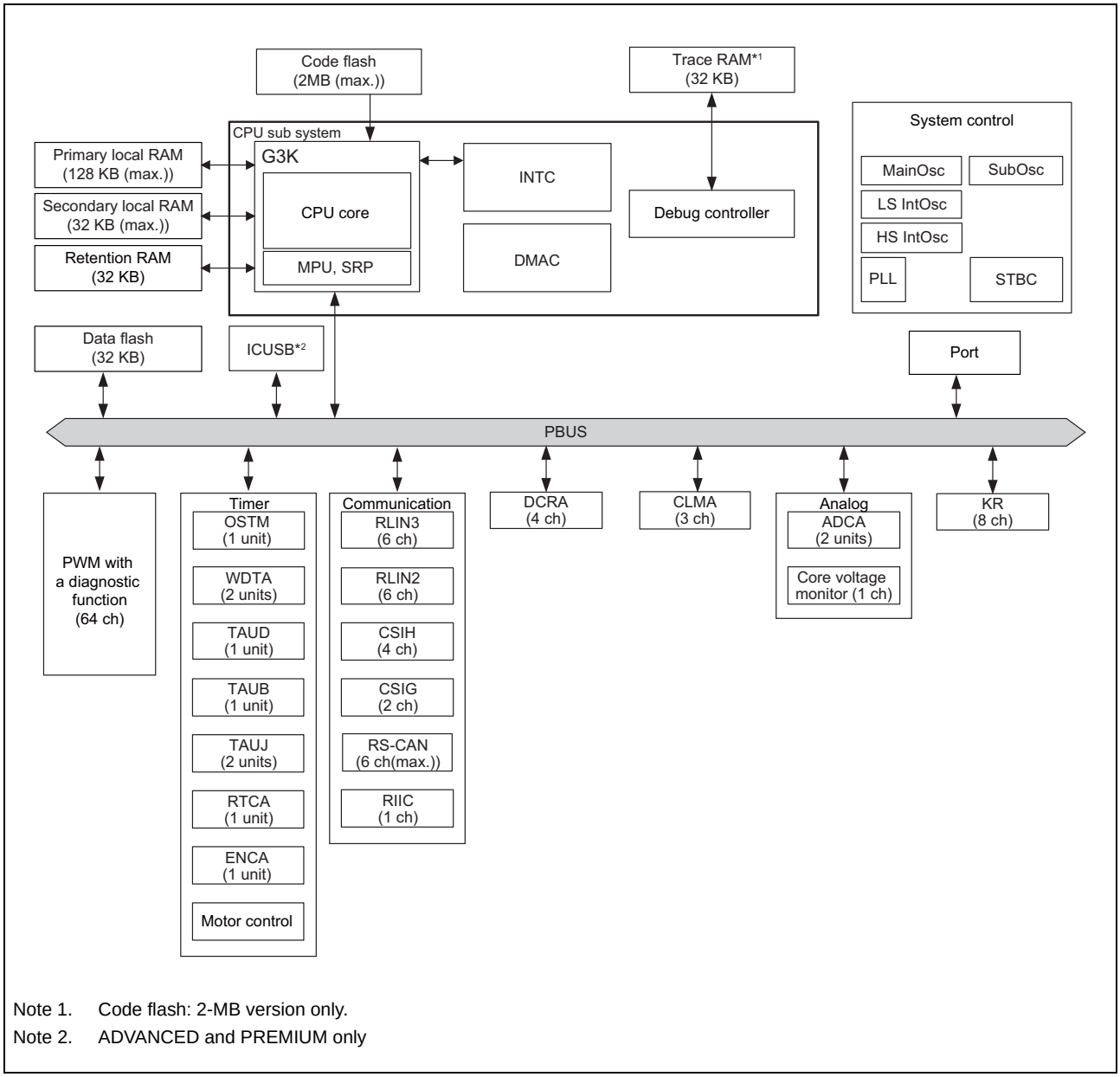
Note 1. ADVANCED only

Figure 1.3 Internal Block Diagram (RH850/F1L 80-pin)



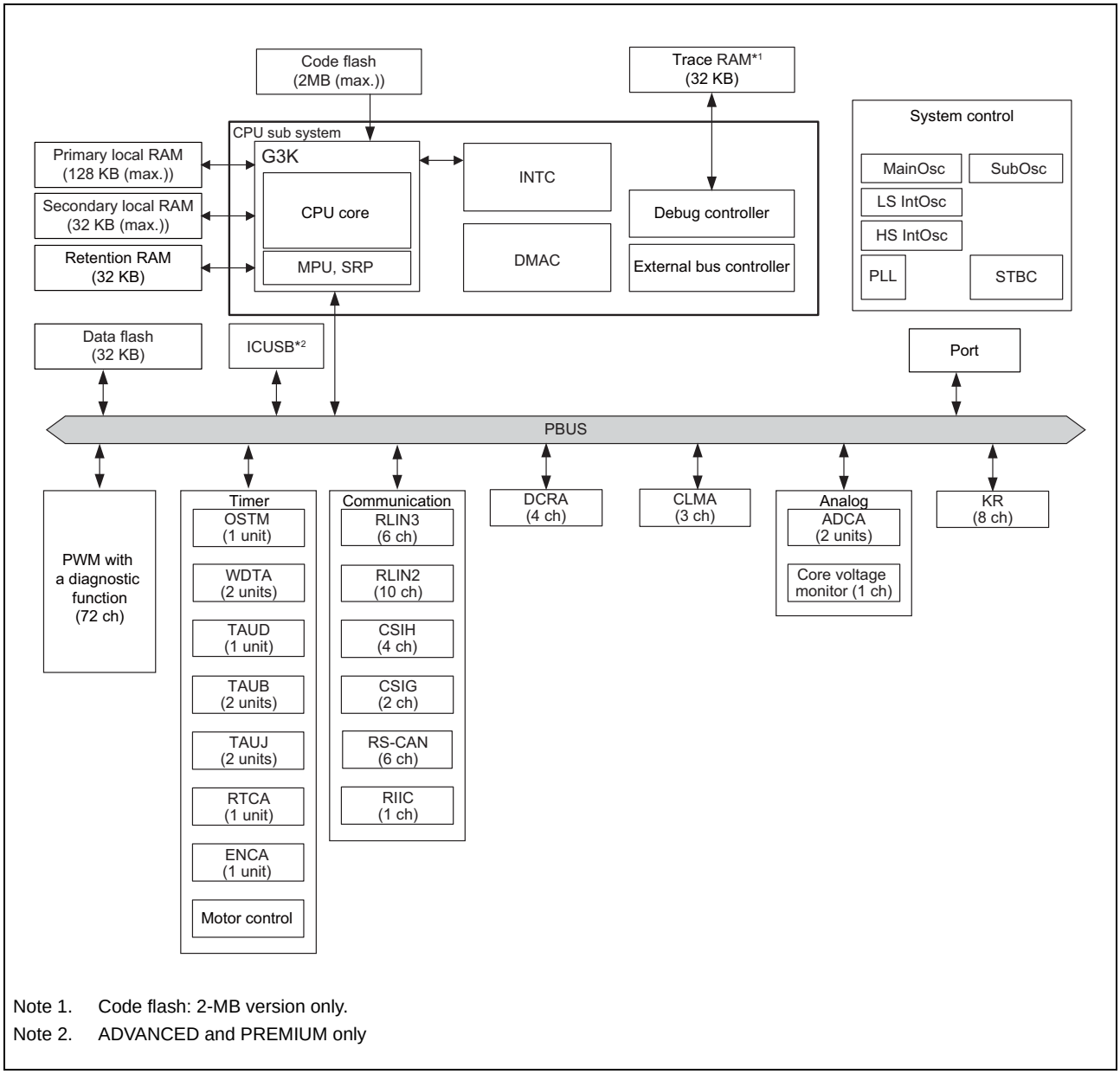
Note 1. ADVANCED and Gateway-1MB only

Figure 1.4 Internal Block Diagram (RH850/F1L 100-pin)



Note 1. Code flash: 2-MB version only.
Note 2. ADVANCED and PREMIUM only

Figure 1.5 Internal Block Diagram (RH850/F1L 144-pin)



Note 1. Code flash: 2-MB version only.
Note 2. ADVANCED and PREMIUM only

Figure 1.6 Internal Block Diagram (RH850/F1L 176-pin)

Section 2 Pin Function

This section describes the pin and port functions.

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

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

2.1 Pin Connection Diagram

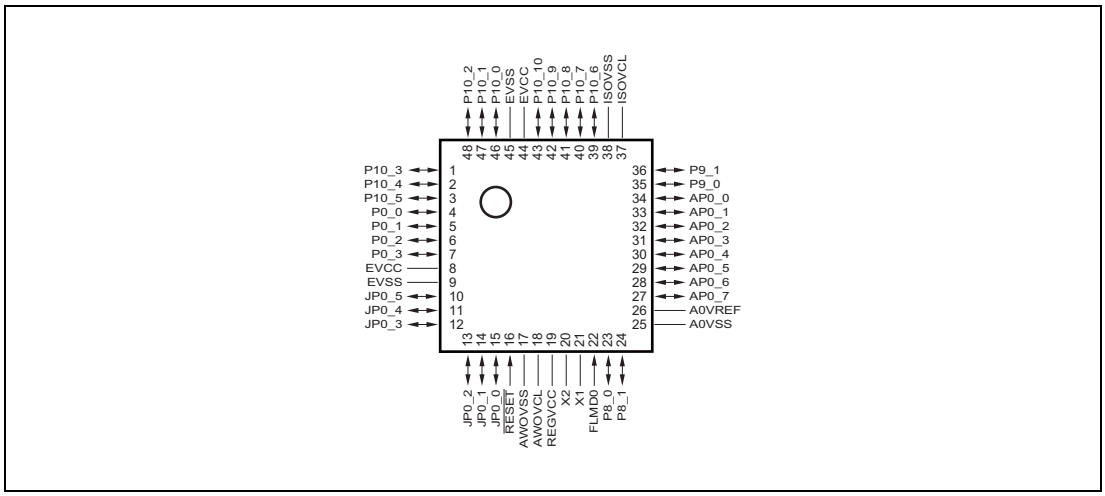


Figure 2.1 Pin Connection Diagram (48-Pin LQFP)

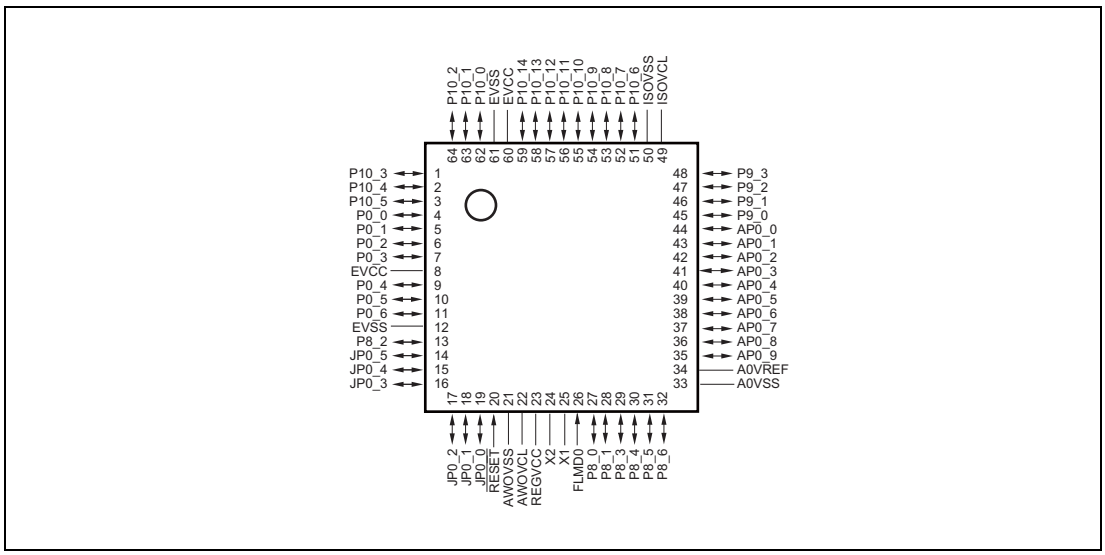


Figure 2.2 Pin Connection Diagram (64-Pin LQFP)

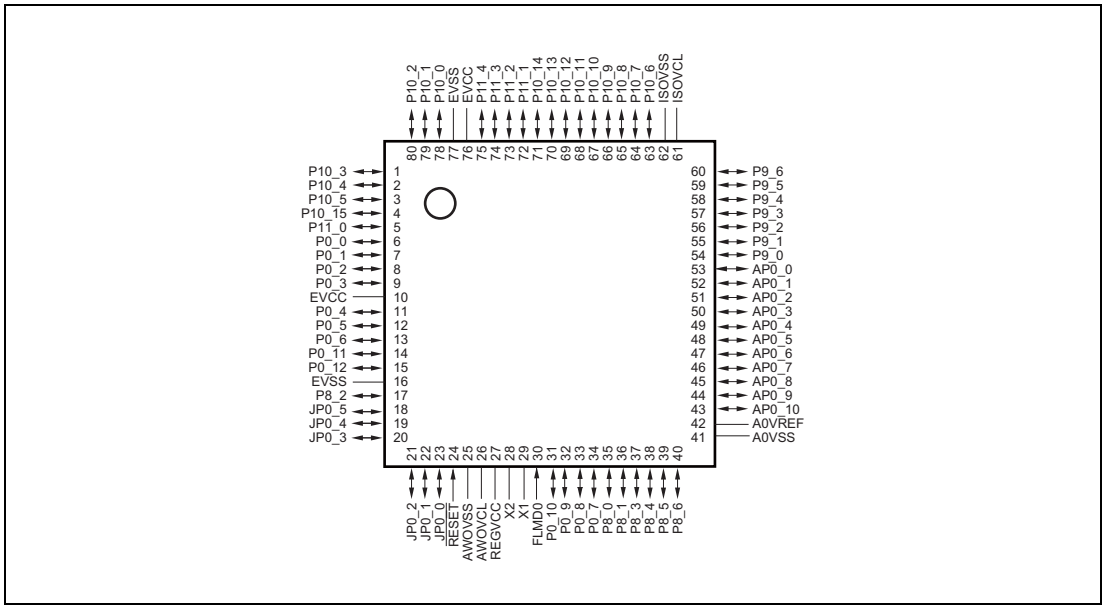


Figure 2.3 Pin Connection Diagram (80-Pin LQFP)

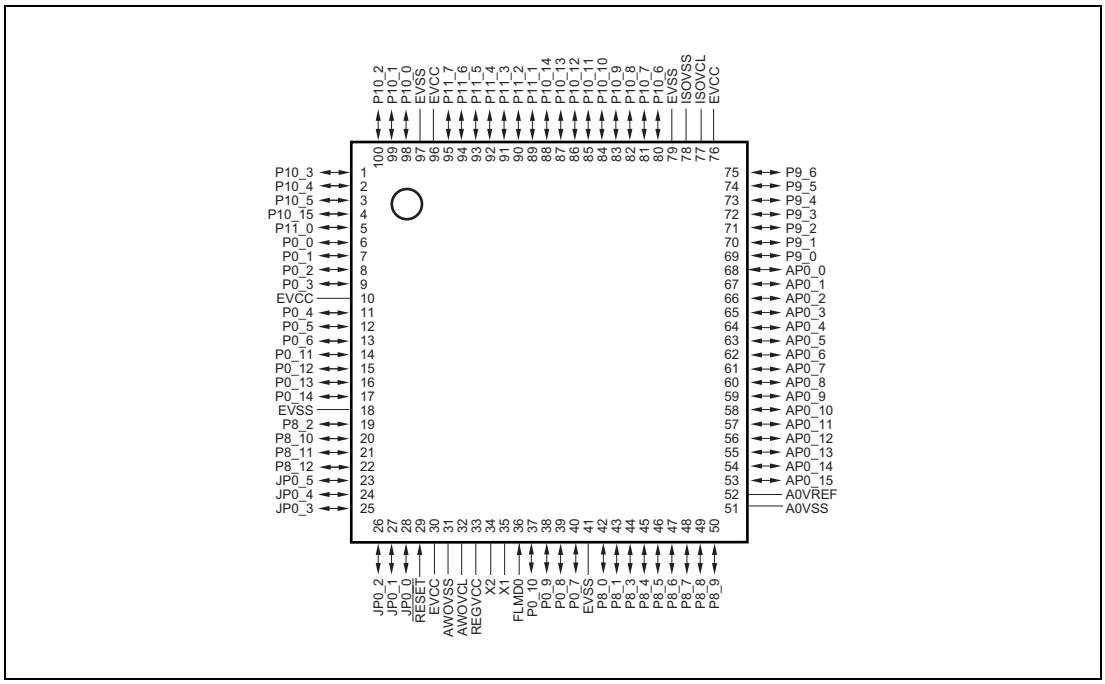


Figure 2.4 Pin Connection Diagram (100-Pin LQFP)

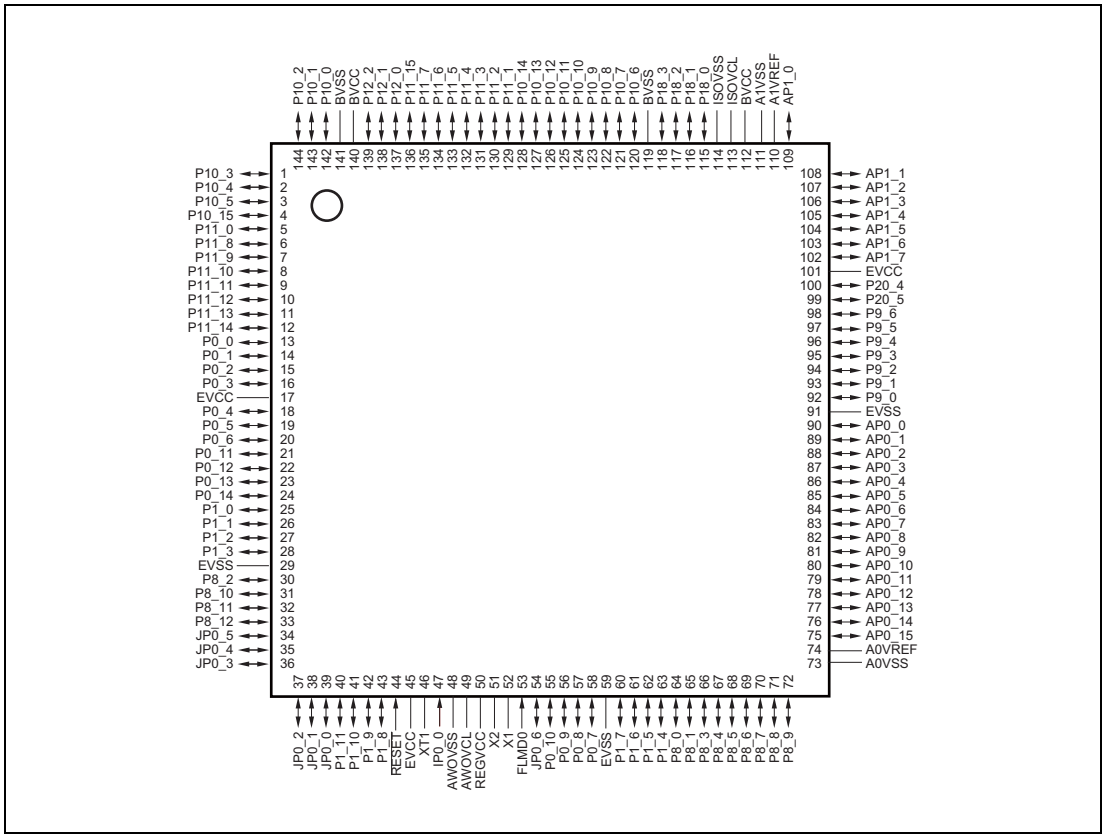


Figure 2.5 Pin Connection Diagram (144-Pin LQFP)

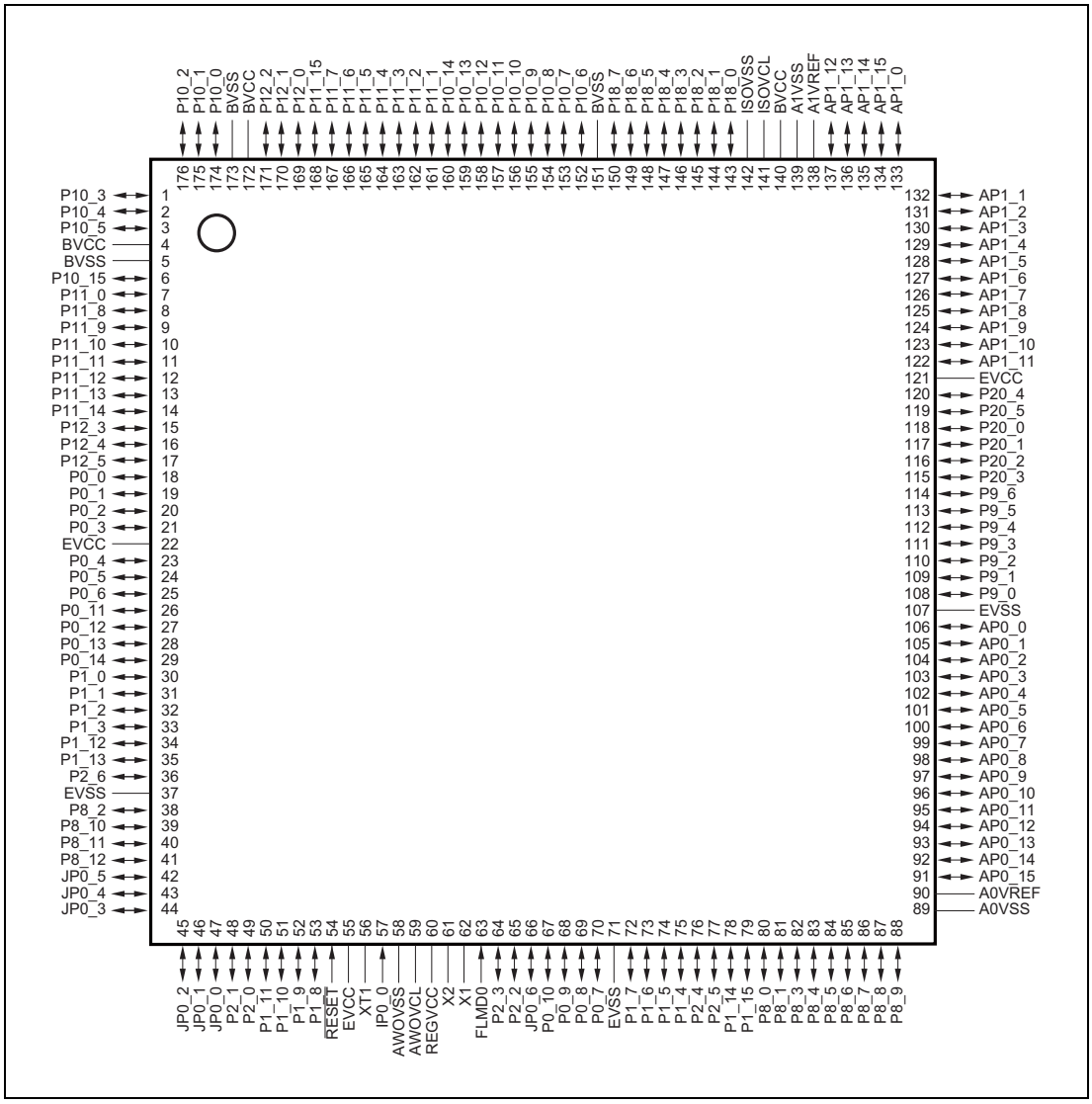


Figure 2.6 Pin Connection Diagram (176-Pin LQFP)

Table 2.1 Pin Assignment 48-Pin LQFP (1/2)

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN
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	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / $\overline{\text{RESETOUT}}$
5	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
6	P0_2 / TAUD0I6 / TAUD0O6 / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
7	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / DPIN1 / CSIH0SO / INTP10
8	EVCC
9	EVSS
10	JP0_5 / NMI / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
11	JP0_4 / $\overline{\text{DCUTRST}}$
12	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
13	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
14	JP0_1 / INTP1 /TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
15	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
16	$\overline{\text{RESET}}$
17	AWOVSS
18	AWOVCL
19	REGVCC
20	X2
21	X1
22	FLMD0
23	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / INTP4 / CSIH0CSS0 / ADCA0I0S
24	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / INTP5 / ADCA0I1S
25	A0VSS
26	A0VREF
27	AP0_7 / ADCA0I7
28	AP0_6 / ADCA0I6
29	AP0_5 / ADCA0I5
30	AP0_4 / ADCA0I4
31	AP0_3 / ADCA0I3
32	AP0_2 / ADCA0I2
33	AP0_1 / ADCA0I1
34	AP0_0 / ADCA0I0
35	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / KR0I4 / ADCA0I2S
36	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / ADCA0I3S
37	ISOVCL
38	ISOVSS
39	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2
40	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O
41	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / FLMD1

Table 2.1 Pin Assignment 48-Pin LQFP (2/2)

Pin No.	Pin Name
42	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX /INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO
43	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1
44	EVCC
45	EVSS
46	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX /INTP0 / CSCXFOUT / PWGA0O / TAPA0UP
47	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / MODE0
48	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / MODE1

Table 2.2 Pin Assignment 64-Pin LQFP (1/2)

Pin No.	Pin Name
1	P10_3 / TAUD0I7 / TAUD0O7 / RIIC0SCL / KR0I1 / PWGA3O / ADCA0TRG1 / TAPA0VN
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	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / $\overline{\text{RESETOUT}}$
5	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
6	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
7	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / INTP10
8	EVCC
9	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / SELDP0 /DPIN8
10	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1
11	P0_6 / INTP2 / DPIN10 / SELDP2
12	EVSS
13	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / PWGA22O / ADCA0I4S
14	JP0_5 / NMI / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
15	JP0_4 / $\overline{\text{DCUTRST}}$
16	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
17	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
18	JP0_1 / INTP1 /TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
19	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
20	$\overline{\text{RESET}}$
21	AWOVSS
22	AWOVCL
23	REGVCC
24	X2
25	X1
26	FLMD0
27	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / ADCA0I0S
28	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / ADCA0I1S
29	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / PWGA23O / ADCA0I5S
30	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / ADCA0I6S
31	P8_5 / TAUJ0I3 / TAUJ0O3 / CSIH0CSS3 / ADCA0I7S
32	P8_6 / NMI / ADCA0I8S
33	A0VSS
34	A0VREF
35	AP0_9 / ADCA0I9
36	AP0_8 / ADCA0I8
37	AP0_7 / ADCA0I7
38	AP0_6 / ADCA0I6
39	AP0_5 / ADCA0I5
40	AP0_4 / ADCA0I4
41	AP0_3 / ADCA0I3

Table 2.2 Pin Assignment 64-Pin LQFP (2/2)

Pin No.	Pin Name
42	AP0_2 / ADCA0I2
43	AP0_1 / ADCA0I1
44	AP0_0 / ADCA0I0
45	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / KR0I4 / ADCA0I2S
46	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / ADCA0I3S
47	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / ADCA0I9S
48	P9_3 / KR0I7 / PWGA21O / ADCA0I10S
49	ISOVCL
50	ISOVSS
51	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX /INTP1
52	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX
53	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / FLMD1
54	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX /INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO
55	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1
56	P10_11 / PWGA16O / RLIN31RX / INTP11
57	P10_12 / PWGA17O / RLIN31TX
58	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O
59	P10_14 / PWGA19O
60	EVCC
61	EVSS
62	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX /INTP0 / CSCXFOUT / PWGA0O / TAPA0UP
63	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / MODE0
64	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / MODE1

Table 2.3 Pin Assignment 80-Pin LQFP (1/2)

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
5	P11_0 / CSIH2RYI / CSIH2RYO
6	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / $\overline{\text{RESETOUT}}$
7	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
8	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
9	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / INTP10
10	EVCC
11	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 /DPIN8
12	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO
13	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC
14	P0_11 / RIIC0SDA / CSIH1CSS2
15	P0_12 / RIIC0SCL
16	EVSS
17	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
18	JP0_5 / NMI / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
19	JP0_4 / $\overline{\text{DCUTRST}}$
20	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
21	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
22	JP0_1 / INTP1 /TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
23	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
24	$\overline{\text{RESET}}$
25	AWOVSS
26	AWOVCL
27	REGVCC
28	X2
29	X1
30	FLMD0
31	P0_10 / INTP3 / CSIH1CSS1 / DPIN11
32	P0_9 / INTP12 / CSIH1CSS0 / DPIN7
33	P0_8/ RLIN21TX / DPIN6 / $\overline{\text{CSIH1SSI}}$
34	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO
35	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / ADCA0I0S
36	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / ADCA0I1S
37	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / ADCA0I5S
38	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / ADCA0I6S
39	P8_5 / TAUJ0I3 / TAUJ0O3 / CSIH0CSS3 / ADCA0I7S
40	P8_6 / NMI / CSIH0CSS4 / ADCA0I8S
41	A0VSS

Table 2.3 Pin Assignment 80-Pin LQFP (2/2)

Pin No.	Pin Name
42	A0VREF
43	AP0_10 / ADCA0I10
44	AP0_9 / ADCA0I9
45	AP0_8 / ADCA0I8
46	AP0_7 / ADCA0I7
47	AP0_6 / ADCA0I6
48	AP0_5 / ADCA0I5
49	AP0_4 / ADCA0I4
50	AP0_3 / ADCA0I3
51	AP0_2 / ADCA0I2
52	AP0_1 / ADCA0I1
53	AP0_0 / ADCA0I0
54	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S
55	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S
56	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
57	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / ADCA0I10S
58	P9_4 / CSIH0CSS5 / ADCA0I11S
59	P9_5 / CSIH0CSS6 / ADCA0I12S
60	P9_6 / CSIH0CSS7 / ADCA0I13S
61	ISOVCL
62	ISOVSS
63	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX /INTP1
64	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX
65	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / FLMD1
66	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX /INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO
67	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1
68	P10_11 / PWGA16O / RLIN31RX / INTP11 / CSIH1CSS0
69	P10_12 / PWGA17O / RLIN31TX / CSIH1CSS1
70	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12
71	P10_14 / PWGA19O / RLIN32TX
72	P11_1 / $\overline{\text{CSIH2SSI}}$ / RLIN20RX
73	P11_2 / CSIH2SO / RLIN20TX
74	P11_3 / CSIH2SC
75	P11_4 / CSIH2SI
76	EVCC
77	EVSS
78	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX /INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI
79	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / MODE0
80	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1

Table 2.4 Pin Assignment 100-Pin LQFP (1/3)

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 / PWGA25O / RLIN22TX / TAUB0I11 / TAUB0O11
6	P0_0 / TAUD0I2 / TAUD0O2 / RLIN20RX / CAN0TX / PWGA10O / $\overline{\text{CSIH0SSI}}$ / DPO / $\overline{\text{RESETOUT}}$
7	P0_1 / TAUD0I4 / TAUD0O4 / CAN0RX / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
8	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
9	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / INTP10
10	EVCC
11	P0_4 / RLIN31RX / INTP11 / CAN2TX / PWGA10O / CSIH1SI / SELDP0 /DPIN8
12	P0_5 / CAN2RX / INTP2 / RLIN31TX / DPIN9 / SELDP1 / CSIH1SO
13	P0_6 / INTP2 / DPIN10 / SELDP2 / CSIH1SC
14	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8
15	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI
16	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX*[1] / INTP5
17	P0_14 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX*[1]
18	EVSS
19	P8_2 / TAUJ0I0 / TAUJ0O0 / DPIN2 / CSIH0CSS0 / INTP6 / PWGA22O / ADCA0I4S
20	P8_10 / CSIH3CSS3 / DPIN14 / PWGA42O / ADCA0I17S
21	P8_11 / TAUJ1I2 / TAUJ1O2 / DPIN15 / PWGA43O / CSIH1CSS4 / ADCA0I18S
22	P8_12 / TAUJ1I3 / TAUJ1O3 / DPIN16 / PWGA44O / CSIH1CSS5 / ADCA0I19S
23	JP0_5 / NMI / TAUJ0I3 / TAUJ0O3 / $\overline{\text{DCURDY}}$ / LPDCLKOUT
24	JP0_4 / $\overline{\text{DCUTRST}}$
25	JP0_3 / INTP3 / CSCXFOUT / TAUJ0I2 / TAUJ0O2 / DCUTMS
26	JP0_2 / INTP2 / TAUJ0I1 / TAUJ0O1 / DCUTCK / LPDCLK
27	JP0_1 / INTP1 /TAUJ0I0 / TAUJ0O0 / DCUTDO / LPDO
28	JP0_0 / INTP0 / DCUTDI / LPDI / LPDIO
29	$\overline{\text{RESET}}$
30	EVCC
31	AWOVSS
32	AWOVCL
33	REGVCC
34	X2
35	X1
36	FLMD0
37	P0_10 / INTP3 / CSIH1CSS1 / DPIN11 / RLIN22TX / TAUB0I6 / TAUB0O6 / CAN4TX*[1]
38	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX*[1]/ INTP4
39	P0_8/ RLIN21TX / DPIN6 / $\overline{\text{CSIH1SSI}}$ / TAUB0I2 / TAUB0O2 / CAN3TX*[1]
40	P0_7 / RLIN21RX / DPIN5 / CSCXFOUT / CSIH1RYI / CSIH1RYO / TAUB0I0 / TAUB0O0 / CAN3RX*[1] / INTP3
41	EVSS

Table 2.4 Pin Assignment 100-Pin LQFP (2/3)

Pin No.	Pin Name
42	P8_0 / TAUJ0I0 / TAUJ0O0 / DPIN2 / PWGA14O / INTP4 / CSIH0CSS0 / ADCA0I0S
43	P8_1 / TAPA0ESO / TAUJ0O1 / DPIN0 / PWGA15O / INTP5 / CSIH1CSS3 / ADCA0I1S
44	P8_3 / TAUJ0I1 / TAUJ0O1 / DPIN3 / CSIH0CSS1 / INTP7 / PWGA23O / ADCA0I5S
45	P8_4 / TAUJ0I2 / TAUJ0O2 / DPIN4 / CSIH0CSS2 / INTP8 / PWGA36O / ADCA0I6S
46	P8_5 / TAUJ0I3 / TAUJ0O3 / CSIH0CSS3 / PWGA37O / ADCA0I7S
47	P8_6 / NMI / CSIH0CSS4 / PWGA38O / ADCA0I8S
48	P8_7 / CSIH3CSS0 / PWGA39O / ADCA0I14S
49	P8_8 / CSIH3CSS1 / PWGA40O / ADCA0I15S
50	P8_9 / CSIH3CSS2 / PWGA41O / ADCA0I16S
51	A0VSS
52	A0VREF
53	AP0_15 / ADCA0I15
54	AP0_14 / ADCA0I14
55	AP0_13 / ADCA0I13
56	AP0_12 / ADCA0I12
57	AP0_11 / ADCA0I11
58	AP0_10 / ADCA0I10
59	AP0_9 / ADCA0I9
60	AP0_8 / ADCA0I8
61	AP0_7 / ADCA0I7
62	AP0_6 / ADCA0I6
63	AP0_5 / ADCA0I5
64	AP0_4 / ADCA0I4
65	AP0_3 / ADCA0I3
66	AP0_2 / ADCA0I2
67	AP0_1 / ADCA0I1
68	AP0_0 / ADCA0I0
69	P9_0 / NMI / PWGA8O / TAUD0I0 / TAUD0O0 / ADCA0TRG0 / CSIH2CSS0 / KR0I4 / ADCA0I2S
70	P9_1 / INTP11 / PWGA9O / TAUD0I2 / TAUD0O2 / KR0I5 / CSIH2CSS1 / ADCA0I3S
71	P9_2 / KR0I6 / PWGA20O / TAPA0ESO / CSIH2CSS2 / ADCA0I9S
72	P9_3 / KR0I7 / PWGA21O / CSIH2CSS3 / ADCA0I10S
73	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
74	P9_5 / CSIH0CSS6 / PWGA34O / TAUJ1I1 / TAUJ1O1 / ADCA0I12S
75	P9_6 / CSIH0CSS7 / PWGA35O / ADCA0I13S
76	EVCC
77	ISOVCL
78	ISOVSS
79	EVSS
80	P10_6 / TAUD0I13 / TAUD0O13 / CSIG0SO / ENCA0TIN0 / ADCA0SEL2 / CAN1RX /INTP1
81	P10_7 / TAUD0I15 / TAUD0O15 / CSIG0SC / ENCA0TIN1 / PWGA4O / CAN1TX
82	P10_8 / TAUD0I10 / TAUD0O10 / CSIG0SI / ENCA0EC / PWGA5O / FLMD1

Table 2.4 Pin Assignment 100-Pin LQFP (3/3)

Pin No.	Pin Name
83	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX /INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO
84	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1
85	P10_11 / PWGA16O / RLIN31RX / INTP11 / CSIH1CSS0 / TAUB0I1 / TAUB0O1
86	P10_12 / PWGA17O / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3
87	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / TAUB0I5 / TAUB0O5
88	P10_14 / PWGA19O / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7
89	P11_1 / $\overline{\text{CSIH2SSI}}$ / RLIN20RX / PWGA26O / TAUB0I13 / TAUB0O13
90	P11_2 / CSIH2SO / RLIN20TX / PWGA27O / TAUB0I15 / TAUB0O15
91	P11_3 / CSIH2SC / CAN3RX*1 / INTP3 / PWGA28O
92	P11_4 / CSIH2SI / CAN3TX*1 / PWGA29O
93	P11_5 / CAN5RX*1 / INTP5 / RLIN33TX / PWGA30O / CSIH3SI
94	P11_6 / RLIN33RX / INTP13 / CAN5TX*1 / PWGA31O / CSIH3SO
95	P11_7 / INTP5 / PWGA32O / CSIH3SC
96	EVCC
97	EVSS
98	P10_0 / TAUD0I1 / TAUD0O1 / CAN0RX /INTP0 / CSCXFOUT / PWGA0O / TAPA0UP / CSIH1SI
99	P10_1 / TAUD0I3 / TAUD0O3 / CAN0TX / PWGA1O / TAPA0UN / CSIH1SC / MODE0
100	P10_2 / TAUD0I5 / TAUD0O5 / RIIC0SDA / KR0I0 / PWGA2O / ADCA0TRG0 / TAPA0VP / CSIH1SO / MODE1

Note 1. CANmTX and CANmRX (m = 3 to 5) are provided only in the F1L for Gateway.

Table 2.5 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
21	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8
22	P0_12 / RIIC0SCL / DPIN13 / PWGA45O / TAUB0I10 / TAUB0O10 / CSIG0SI
23	P0_13 / RLIN32RX / INTP12 / PWGA46O / TAUB0I12 / TAUB0O12 / CSIG0SO / CAN5RX*2 / INTP5
24	P0_14 / RLIN32TX / PWGA47O / TAUB0I14 / TAUB0O14 / CSIG0SC / CAN5TX*2
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 2.5 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*2
56	P0_9 / INTP12 / CSIH1CSS0 / DPIN7 / RLIN22RX / TAUB0I4 / TAUB0O4 / CAN4RX*2 / INTP4
57	P0_8/ RLIN21TX / DPIN6 / $\overline{\text{CSIH1SSI}}$ / 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 2.5 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 / ADCA0I10S
96	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
97	P9_5 / CSIH0CSS6 / PWGA34O / TAUJ1I1 / TAUJ1O1 / ADCA0I12S
98	P9_6 / CSIH0CSS7 / PWGA35O / ADCA0I13S
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 / ENCA0EC / PWGA5O / FLMD1
123	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO

Table 2.5 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 / CSIH1CSS0 / TAUB0I1 / TAUB0O1
126	P10_12 / PWGA17O / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3
127	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / TAUB0I5 / TAUB0O5
128	P10_14 / ADCA1TRG0 / PWGA19O / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7
129	P11_1 / $\overline{\text{CSIH2SSI}}$ / RLIN20RX / 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*2 / INTP5 / RLIN33TX / PWGA30O / CSIH3SI
134	P11_6 / RLIN33RX / INTP13 / CAN5TX*2 / 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 2-MB code flash memory.

Note 2. The CANmTX and CANmRX pins (m = 4, 5) are only available in devices with 1.5- and 2-MB code flash memories.

Table 2.6 Pin Assignment 176-Pin LQFP (1/5)

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 / RLIN20TX / INTP0 / PWGA11O / CSIH0SI / APO
20	P0_2 / TAUD0I6 / TAUD0O6 / CAN1RX / RLIN30TX / PWGA12O / CSIH0SC / INTP1 / DPO
21	P0_3 / TAUD0I8 / TAUD0O8 / RLIN30RX / CAN1TX / DPIN1 / PWGA13O / CSIH0SO / INTP10
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
26	P0_11 / RIIC0SDA / DPIN12 / CSIH1CSS2 / TAUB0I8 / TAUB0O8 / RLIN26RX
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
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

Table 2.6 Pin Assignment 176-Pin LQFP (2/5)

Pin No.	Pin Name
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
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 / $\overline{\text{CSIH1SSI}}$ / 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
77	P2_5 / RLIN29TX
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

Table 2.6 Pin Assignment 176-Pin LQFP (3/5)

Pin No.	Pin Name
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
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 / ADCA0I10S
112	P9_4 / CSIH0CSS5 / PWGA33O / TAUJ1I0 / TAUJ1O0 / ADCA0I11S
113	P9_5 / CSIH0CSS6 / PWGA34O / TAUJ1I1 / TAUJ1O1 / ADCA0I12S
114	P9_6 / CSIH0CSS7 / PWGA35O / ADCA0I13S
115	P20_3 / CAN4TX / PWGA67O
116	P20_2 / CAN4RX / INTP4 / PWGA66O
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

Table 2.6 Pin Assignment 176-Pin LQFP (4/5)

Pin No.	Pin Name
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
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 / ENCA0EC / PWGA5O / MEMC0AD2 / FLMD1
155	P10_9 / TAUD0I12 / TAUD0O12 / RLIN30RX / INTP10 / ENCA0E0 / PWGA6O / CSIH0RYI / CSIH0RYO / MEMC0AD3
156	P10_10 / TAUD0I14 / TAUD0O14 / RLIN30TX / ENCA0E1 / PWGA7O / CSIH0CSS1 / MEMC0AD4
157	P10_11 / PWGA16O / RLIN31RX / INTP11 / CSIH1CSS0 / TAUB0I1 / TAUB0O1 / MEMC0AD5
158	P10_12 / PWGA17O / RLIN31TX / CSIH1CSS1 / TAUB0I3 / TAUB0O3 / MEMC0AD6
159	P10_13 / $\overline{\text{CSIH0SSI}}$ / PWGA18O / RLIN32RX / INTP12 / TAUB0I5 / TAUB0O5 / MEMC0AD7
160	P10_14 / ADCA1TRG0 / PWGA19O / RLIN32TX / $\overline{\text{CSIH3SSI}}$ / TAUB0I7 / TAUB0O7 / MEMC0AD8
161	P11_1 / $\overline{\text{CSIH2SSI}}$ / RLIN20RX / 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

Table 2.6 Pin Assignment 176-Pin LQFP (5/5)

Pin No.	Pin Name
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 2-MB code flash memory.

2.2 Pin Description

Table 2.7 Pin Function (1/5)

Pin Name	No. of Pins						IO	Pin Function	Unit
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins			
AnVREF	√	√	√	√	√	√	—	ADCAn voltage supply and reference voltage	ADCAn
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
AnVSS	√	√	√	√	√	√	—	ADCAn ground	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
ADCAnIm	√	√	√	√	√	√	I	ADCAn input channel m with 12-bit resolution	
	n = 0, m = 0 to 7	n = 0, m = 0 to 9	n = 0, m = 0 to 10	n = 0, m = 0 to 15	n = 0, m = 0 to 15 n = 1, m = 0 to 7	n = 0, 1, m = 0 to 15			
ADCA0ImS	√	√	√	√	√	√	I	ADCA0 input channel m with 10-bit resolution	
	m = 0 to 3	m = 0 to 10	m = 0 to 13	m = 0 to 19	m = 0 to 19	m = 0 to 19			
ADCA1ImS	—	—	—	—	√	√	I	ADCA1 input channel m with 10-bit resolution	
					m = 0 to 3	m = 0 to 7			
ADCA0SELy	√	√	√	√	√	√	O	Selection pin y for ADCA0 input and external MPX	
	y = 0 to 2	y = 0 to 2	y = 0 to 2	y = 0 to 2	y = 0 to 2	y = 0 to 2			
ADCAnTRGy	√	√	√	√	√	√	I	ADCAn external trigger pin y	
	n = 0, y = 0 to 2	n = 0, y = 0 to 2	n = 0, y = 0 to 2	n = 0, 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 7	m = 0 to 9	m = 0 to 10	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			
APO	√	√	√	√	√	√	O	Output for analog input port	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	m = 0 to 2	m = 0 to 2	m = 0 to 2, m = 0 to 5*2	m = 0 to 3, m = 0 to 5*3	m = 0 to 5			
CANmTX	√	√	√	√	√	√	O	CANm transmit data output	
	m = 0	m = 0 to 2	m = 0 to 2	m = 0 to 2, m = 0 to 5*2	m = 0 to 3, m = 0 to 5*3	m = 0 to 5			
CSCXFOUT	√	√	√	√	√	√	O	Clock output	Clock
CSIGnRYI	√	√	√	√	√	√	I	CSIGn ready (1) / busy (0) input signal	CSIGn
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
CSIGnRYO	√	√	√	√	√	√	O	CSIGn ready (1) / busy (0) output signal	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
CSIGnSC	√	√	√	√	√	√	IO	CSIGn serial clock signal	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
CSIGnSI	√	√	√	√	√	√	I	CSIGn serial data input	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
CSIGnSO	√	√	√	√	√	√	O	CSIGn serial data output	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			
$\overline{\text{CSIGnSSI}}$	√	√	√	√	√	√	I	CSIGn SS function control input signal	
	n = 0	n = 0	n = 0	n = 0	n = 0, 1	n = 0, 1			

Table 2.7 Pin Function (2/5)

Pin Name	No. of Pins 48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	IO	Pin Function	Unit
CSIHnCSS0	√	√	√	√	√	√	O	CSIHn serial peripheral chip select signal 0	CSIHn
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS1	√	√	√	√	√	√	O	CSIHn serial peripheral chip select signal 1	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS2	—	√	√	√	√	√	O	CSIHn serial peripheral chip select signal 2	
		n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS3	—	√	√	√	√	√	O	CSIHn serial peripheral chip select signal 3	
		n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnCSS4	—	—	√	√	√	√	O	CSIHn serial peripheral chip select signal 4	
			n = 0	n = 0, 1	n = 0 to 2	n = 0 to 2			
CSIHnCSS5	—	—	√	√	√	√	O	CSIHn serial peripheral chip select signal 5	
			n = 0	n = 0, 1	n = 0 to 2	n = 0 to 2			
CSIHnCSS6	—	—	√	√	√	√	O	CSIHn serial peripheral chip select signal 6	
			n = 0	n = 0	n = 0	n = 0			
CSIHnCSS7	—	—	√	√	√	√	O	CSIHn serial peripheral chip select signal 7	
			n = 0	n = 0	n = 0	n = 0			
CSIHnRYI	√	√	√	√	√	√	I	CSIHn ready (1) / busy (0) input signal	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnRYO	√	√	√	√	√	√	O	CSIHn ready (1) / busy (0) output signal	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSC	√	√	√	√	√	√	IO	CSIHn serial clock signal	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSI	√	√	√	√	√	√	I	CSIHn serial data input	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
CSIHnSO	√	√	√	√	√	√	O	CSIHn serial data output	
	n = 0	n = 0	n = 0 to 2	n = 0 to 3	n = 0 to 3	n = 0 to 3			
$\overline{\text{CSIHnSSI}}$	√	√	√	√	√	√	I	CSIHn slave select input signal	
	n = 0	n = 0	n = 0 to 2	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 2	m = 0 to 4, 8 to 10	m = 0 to 11	m = 0 to 16	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	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	O	Event output	TEU_OUT

Table 2.7 Pin Function (3/5)

Pin Name	No. of Pins						IO	Pin Function	Unit
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins			
FLMD0	√	√	√	√	√	√	I	Operating mode select pin 0	Mode
FLMD1	√	√	√	√	√	√	I	Operating mode select pin 1	
INTPm	√ m = 0 to 5, 10, 11	√ m = 0 to 5, 10, 11	√ m = 0 to 8, 10 to 12	√ m = 0 to 8, 10 to 13	√ m = 0 to 15	√ m = 0 to 15	I	External interrupt input m	INTC
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	√ m = 0 to 5	√ m = 0 to 5	√ m = 0 to 5	√ m = 0 to 5	√ m = 0 to 6	√ m = 0 to 6	IO	JTAG port 0_m	Port
KR0Im	√ m = 0 to 5	√ m = 0 to 7	√ m = 0 to 7	√ m = 0 to 7	√ m = 0 to 7	√ m = 0 to 7	I	KR0 key input signal	KR0
LPDCLK	√	√	√	√	√	√	I	LPD clock input (4-pin mode)	LPD
LPDCLKOUT	√	√	√	√	√	√	O	LPD clock output (4-pin mode)	
LPDI	√	√	√	√	√	√	I	LPD data input (4-pin mode)	
LPDIO	√	√	√	√	√	√	IO	LPD data input / output (1-pin mode)	
LPDO	√	√	√	√	√	√	O	LPD data output (4-pin mode)	
MEMC0Am	—	—	—	—	—	√ m = 16 to 19	O	MEMC0 address m	MEMC0
MEMC0ADm	—	—	—	—	—	√ m = 0 to 15	IO	MEMC0 address / data m	
$\overline{\text{MEMC0ASTB}}$	—	—	—	—	—	√	O	MEMC0 address strobe	
$\overline{\text{MEMC0BENm}}$	—	—	—	—	—	√ m = 0, 1	O	MEMC0 byte enable m	
MEMC0CLK	—	—	—	—	—	√	O	MEMC0 clock output	
$\overline{\text{MEMC0CSm}}$	—	—	—	—	—	√ m = 0 to 3	O	MEMC0 chip select m	
$\overline{\text{MEMC0RD}}$	—	—	—	—	—	√	O	MEMC0 read strobe	
$\overline{\text{MEMC0WAIT}}$	—	—	—	—	—	√	I	MEMC0 wait input	
$\overline{\text{MEMC0WR}}$	—	—	—	—	—	√	O	MEMC0 write strobe	
MODEm	√ m = 0, 1	√ m = 0, 1	√ m = 0, 1	√ m = 0, 1	√ m = 0, 1	√ m = 0, 1	I	Sub operating mode select (Boundary scan / OSC monitor)	Mode
NMI	√	√	√	√	√	√	I	External non-maskable interrupt input	INTC
P0_m	√ m = 0 to 3	√ m = 0 to 6	√ m = 0 to 12	√ m = 0 to 14	√ m = 0 to 14	√ m = 0 to 14	IO	Port 0_m	Port
P1_m	—	—	—	—	√ m = 0 to 11	√ m = 0 to 15	IO	Port 1_m	
P2_m	—	—	—	—	—	√ m = 0 to 6	IO	Port 2_m	
P8_m	√ m = 0, 1	√ m = 0 to 6	√ m = 0 to 6	√ m = 0 to 12	√ m = 0 to 12	√ m = 0 to 12	IO	Port 8_m	

Table 2.7 Pin Function (4/5)

Pin Name	No. of Pins						IO	Pin Function	Unit
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins			
P9_m	√	√	√	√	√	√	IO	Port 9_m	Port
	m = 0, 1	m = 0 to 3	m = 0 to 6	m = 0 to 6	m = 0 to 6	m = 0 to 6			
P10_m	√	√	√	√	√	√	IO	Port 10_m	
	m = 0 to 10	m = 0 to 14	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 15			
P11_m	—	—	√	√	√	√	IO	Port 11_m	
			m = 0 to 4	m = 0 to 7	m = 0 to 15	m = 0 to 15			
P12_m	—	—	—	—	√	√	IO	Port 12_m	
					m = 0 to 2	m = 0 to 5			
P18_m	—	—	—	—	√	√	IO	Port 18_m	
					m = 0 to 3	m = 0 to 7			
P20_m	—	—	—	—	√	√	IO	Port 20_m	
					m = 4, 5	m = 0 to 5			
PWGAnO	√	√	√	√	√	√	O	PWGAn output signal	PWM_diag
	n = 0 to 12	n = 0 to 23	n = 0 to 23	n = 0 to 47	n = 0 to 63	n = 0 to 71			
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	RLIN2m
	m = 0, 1	m = 0, 1	m = 0, 1	m = 0 to 2	m = 0 to 5	m = 0 to 9			
RLIN2mTX	√	√	√	√	√	√	O	RLIN2m transmit data output	
	m = 0, 1	m = 0, 1	m = 0, 1	m = 0 to 2	m = 0 to 5	m = 0 to 9			
RLIN3nRX	√	√	√	√	√	√	I	RLIN3n receive data input	RLIN3n
	n = 0	n = 0, 1	n = 0 to 2	n = 0 to 3	n = 0 to 5	n = 0 to 5			
RLIN3nTX	√	√	√	√	√	√	O	RLIN3n transmit data output	
	n = 0	n = 0, 1	n = 0 to 2	n = 0 to 3	n = 0 to 5	n = 0 to 5			
RTCA0OUT	—	—	—	—	√	√	O	RTCA0 1Hz output	RTCA0
SELDPk	—	√	√	√	√	√	O	External multiplexer select signal output k for the digital port	LPS
		k = 0 to 2	k = 0 to 2	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)	
TAUD0Im	√	√	√	√	√	√	I	TAUD0 channel input m	TAUD0
	m = 0 to 15	m = 0 to 15	m = 0 to 15	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	m = 0 to 15	m = 0 to 15	m = 0 to 15			

Table 2.7 Pin Function (5/5)

Pin Name	No. of Pins						IO	Pin Function	Unit
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins			
TAUBnIm	—	—	—	√	√	√	I	TAUBn channel input m	TAUBn
				n = 0, m = 0 to 15	n = 0, 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, m = 0 to 15	n = 0, 1, m = 0 to 15			
TAUJnIm	√	√	√	√	√	√	I	TAUJn channel input m	TAUJn
	n = 0, m = 0 to 3	n = 0, m = 0 to 3	n = 0, m = 0 to 3	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, m = 0 to 3	n = 0, m = 0 to 3	n = 0, m = 0 to 3	n = 0, 1, m = 0 to 3	n = 0, 1, m = 0 to 3	n = 0, 1, m = 0 to 3			
X1, X2	√	√	√	√	√	√	—	Main OSC connections	MOSC
XT1, XT2	—	—	—	—	√	√	—	Sub OSC connections	SOSC

Note 1. The $\overline{\text{EVTO}}$ pin is only available in devices with 2-MB code flash memory.

Note 2. “m = 0 to 5” is only available for F1L for Gateway.

Note 3. “m = 0 to 5” is only available in devices with 1.5- and 2- MB code flash memories.

CAUTION

- **When pin functions for a peripheral module are allocated to multiple pins, select and use a set of peripheral function pins from the same pin group.**
  (e.g.) When RS-CAN channel 0 is used:
  CAN0TX P0_0 P10_1
  CAN0RX P0_1 P10_0
  Use one of the pin combinations:
  - **P0_0 and P0_1, or**
  - **P10_0 and P10_1.**

***The combinations of P0_0 and P10_0, and P0_1 and P10_1 are not allowed.**

2.3 Ports and Pin Functions During and After Reset

Table 2.8 Ports and Pin Functions During and After Reset

Pins	During Reset	After Reset
JP0_0 to JP0_3, JP0_5, P0_6	High impedance	Mode dependent*1
JP0_4	Input*3	Mode dependent*1
P0_0	Output*2	Output*2
P0_1 to P0_14	High impedance	High impedance
P1, P2, P8-12, P18, P20	High impedance	High impedance*4
FLMD0	Input	Input
$\overline{\text{RESET}}$	Input	Input
AP0, AP1	High impedance	High impedance

Note 1. For each operating mode, see **Section 3, CPU System** and **Section 5, Operating Modes**.
Note that these ports are also used as on-chip debug interface. For port functions when these ports are used as on-chip debug interface, see **Section 2.10, Port (General I/O) Function Overview**.

Note 2. $\overline{\text{RESETOUT}}$ is output. For details, see **Section 2.11, Port (Special I/O) Function Overview**.

Note 3. When a power-on reset is released, the JP0_4 pin should not be driven high.

Note 4. P10_1 (MODE1), P10_2 (MODE2), and P10_8 (FLMD1) depend on each operating mode.

2.4 Port State in Standby Mode

For the port state in standby mode, see **Section 11.1.4, 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 2.9 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 AWOVSS
IP0_0	Connected to REGVDD or AWOVSS through resistor (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 through resistor (when the PIBCn_m and PMCn_m bits are 1) Output: Open
P0_0	Input: Open (when the PIBCn_m and PMCn_m bits are 0) Connected to EVSS through resistor (when the PIBCn_m and PMCn_m bits are 1) Output: Open
JP0_4	Connected to EVSS through resistor*3
P10_1, P10_2, P10_8	Input: Open (when the PIBCn_m and PMCn_m bits are 0) For 48 pins, 64 pins, 80 pins, 100 pins: Connected to EVSS through resistor (when the PIBCn_m and PMCn_m bits are 1) For 144 pins and 176 pins: Connected to BVSS through resistor (when the PIBCn_m and PMCn_m bits are 1) Output: Open
P10 (excluding P10_1, P10_2, P10_8) P11 P12 P18	Input: Open (when the PIBCn_m and PMCn_m bits are 0) For 48 pins, 64 pins, 80 pins, 100 pins: Connected to EVCC or EVSS through resistor (when the PIBCn_m and PMCn_m bits are 1) For 144 pins and 176 pins: Connected to BVCC or BVSS through resistor (when the PIBCn_m and PMCn_m bits are 1) Output: Open
AP0	Input: Open (when the PIBCn_m bit is 0) Connected to A0VREF or A0VSS through 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 through resistor (when the PIBCn_m bit is 1) Output: Open
Nexus/LPD I/F (JP0)	DCUTDI/LPDI/LPDIO (JP0_0): Connected to EVCC through resistor DCUTDO/LPDO (JP0_1): Open DCUTCK/LPDCLK (JP0_2): Open DCUTMS (JP0_3): Open*2 $\overline{\text{DCUTRST}}$ (JP0_4): Connected to EVSS through resistor*3 $\overline{\text{DCURDY}}$/LPDCLKOUT (JP0_5):Open $\overline{\text{EVTO}}$ (JP0_6): Open*1

Note 1. The $\overline{\text{EVTO}}$ pin is only available in devices with 2-MB code flash memory.

Note 2. In case when a debugging interface is used, this pin should be connected to EVCC through resistor depending on the development tool made by a third party.

Note 3. For details, see the specifications of the development tool.

2.6 RH850/F1L Port Features

2.6.1 Port Group

The RH850/F1L provides the following port groups, indicated by the numbers in the table below.

Table 2.10 Port Groups in RH850/F1L

No. of Pins	Port Group	RH850/F1L
48 pins	Number	6
	Name	P0, P8 to P10, JP0, AP0
64 pins	Number	6
	Name	P0, P8 to P10, JP0, AP0
80 pins	Number	7
	Name	P0, P8 to P11, JP0, AP0
100 pins	Number	7
	Name	P0, P8 to P11, JP0, AP0
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

2.6.2 Port Group Index n

Throughout this section, the port groups are identified by using the index “n” (n = 0 to 2, 8 to 12, 18, 20, JP0, AP0, AP1, and IP0). 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 in general.

Table 2.11 Register Base Address

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

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/F1L 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)
- The alternative I/O direction can also be set by the port mode register (PM).
- All possible port functions are shown in the tables listed below.
  Table 2.38, **Table 2.40**, **Table 2.42**, **Table 2.44**, **Table 2.46**, **Table 2.48**, **Table 2.50**, **Table 2.52**, **Table 2.54**, **Table 2.56**, **Table 2.58**, **Table 2.60**, and **Table 2.62** in **Section 2.9.2, Pin Function Configuration**.

CAUTION

Some input or output functions may be assigned to more than one port. Activate only one port. Do not activate multiple ports at the same time.

[e.g.]

INTP0 is assigned to the following ports on this device. However, only one of them should be activated at most. Do not activate the other ports.

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

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.

Port type

The port controller is determined by a setting register specification. The type of each different controller is known as the port type.

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 JPMn.JPMn_m denotes the JPMn_m bit of the JPMn port mode control register.

NOTE

In this section, the description about all ports other than the JTAG port and their registers 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 by using software.

- 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 selected by the alternative function.

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

Table 2.12 Pin Function Configuration (Overview)

	Bit			
Mode	**PMCn_m**	**PMn_m**	**PIPCn_m**	**I/O**
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).

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

- Software I/O control alternative mode (PIPCn.PIPCn_m = 0)
  - Output (PMn_m = 0): Alternative output mode 1 to Alternative output mode 5
  - Input (PMn_m = 1): Alternative input mode 1 to Alternative input mode 5
- Direct I/O control alternative mode (PIPCn.PIPCn_m = 1)
  - The I/O mode for Alternative output mode 1 to Alternative output mode 5 and Alternative input mode 1 to Alternative input mode 5 is directly selected by the alternative function.

Table 2.13 Alternative Mode Selection Overview (PMCn.PMCn_m Bit = 1)

Mode	Register					I/O
	PIPC*1	PM*1	PFCAE	PFCE	PFC	
Alternative-function output mode 1 (ALT-OUT1)	0	0	0	0	0	O
Alternative-function input mode 1 (ALT-IN1)		1				I
Alternative-function output mode 2 (ALT-OUT2)		0	0	0	1	O
Alternative-function input mode 2 (ALT-IN2)		1				I
Alternative-function output mode 3 (ALT-OUT3)		0	0	1	0	O
Alternative-function input mode 3 (ALT-IN3)		1				I
Alternative-function output mode 4 (ALT-OUT4)		0	0	1	1	O
Alternative-function input mode 4 (ALT-IN4)		1				I
Alternative-function output mode 5 (ALT-OUT5)		0	1	0	0	O
Alternative-function input mode 5 (ALT-IN5)		1				I
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 five 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 corresponding bit of the port register Pn.Pn_m, 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 2.14 PPRn_m Read Values

PMCn_m	PMn_m	PIBCn_m	PIPCn_m	PODCn_m	Mode	PPRn_m Read Value
0	1	0	X	X	Port input, input buffer disabled	Pn.Pn_m register
		1		X	Port input, input buffer enabled	Pn_m pin
	0	X		0	Port push-pull output	Pn.Pn_m register*1
				1	Port open-drain output	
1	1	X	0	X	Software I/O control alternative-function input	Pn_m pin
	0			0	Software I/O control alternative-function push-pull output	Alternative-function internal output signal*1
				1	Software I/O control alternative-function open-drain output	
	X		1	0	Direct I/O control alternative-function push-pull output	I/O port in alternative mode: • Input: Pn_m pin • Output: Alternative-function internal output signal*1
				1	Direct I/O control alternative-function open-drain output	

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

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-function mode or direct I/O control alternative-function mode.

- PODCn.PODn_m bit
  This bit selects push-pull output (PODCn_m = 0) or open-drain output (PODCn_m = 1).
- PBDCn.PBDCn_m bit
  When this bit is set to 1, PPRn_m is commanded to read the level of the Pn_m pin. In other words, if the port is in output mode, the bidirectional mode, in which the level of the Pn_m pin can be read, is enabled.

CAUTION

When using Pn_m as an alternative 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 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) = 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 Port Type

The port type of each port must be classified into specific port types. This section describes the maximum possible port structure.

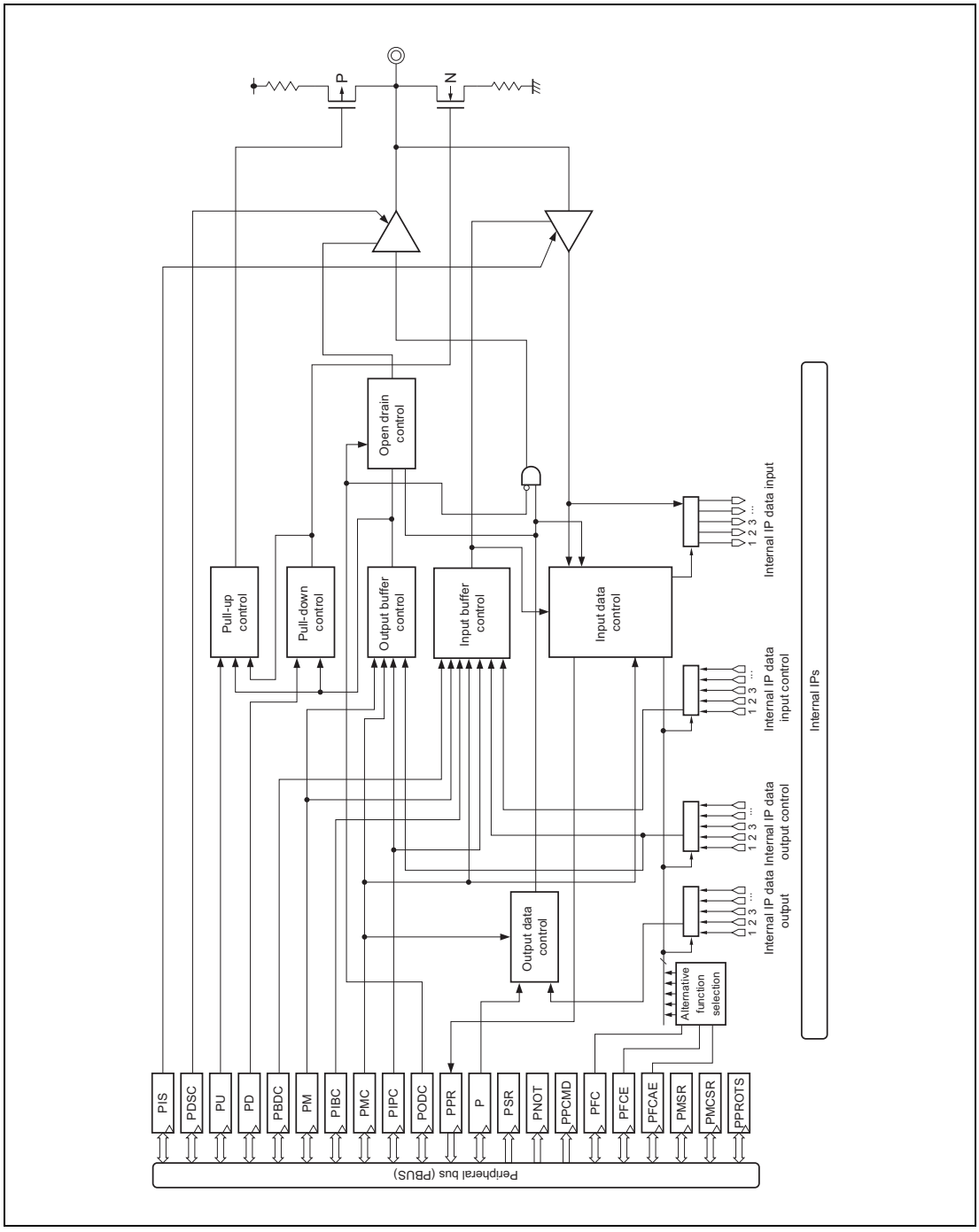


Figure 2.7 Port Control Logic Diagram

CAUTION

Use documented alternative functions only. The behavior and performance are not guaranteed when undocumented alternative functions are selected.

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 Registers**

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 2.15 Port Group Configuration Registers (1/2)

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

Table 2.15 Port Group Configuration Registers (2/2)

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

Index n

In **Table 2.15, Port Group Configuration Registers**, the index “n” in register abbreviations denotes the actual indexes 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 2.16**.

Table 2.16 Number of Pins on the Device, Name of Port Groups, and Values for “n” in Register Abbreviations

Number of Pins on the Device	Name of Port Groups	Values for “n”
48 pins	P0, P8, P9, P10	0, 8, 9, 10
	AP0	0
64 pins	P0, P8, P9, P10	0, 8, 9, 10
	AP0	0
80 pins	P0, P8, P9, P10, P11	0, 8, 9, 10, 11
	AP0	0
100 pins	P0, P8, P9, P10, P11	0, 8, 9, 10, 11
	AP0	0
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

JTAG port registers

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

All descriptions (except for those of the PFCEn register, PECAEn 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/written in 16-bit units.
JPMC0: This register can be read/written in 8-bit units.

Address: PMCn: <PORTn_base> + 0400_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)
JPMC0: <JPORT0_base> + 0040_H Note 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 n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, and **Table 2.59, Control Registers (P20)**.

Table 2.17 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). 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 pins that pass through the noise filter are used. These pins may require the setting of the FCLA0CTLm_<name>, DFNA<name>CTL bits 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 PMCn.PMCn_m is set by the corresponding bit in the lower 16 bits of PMCSRn.

Access: This register can be read/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.

Address: PMCSRn: <PORTn_base> + 0900_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)
JPMCSR0: <JPORT0_base> + 0090_H Note 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 effective bit positions (value for the index n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, and **Table 2.59, Control Registers (P20)**.

Table 2.18 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 pin Pn_m is controlled by the port mode register PMn.PMn_m or by an alternative function.

If pin Pn_m 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 sharing function that needs to set PIPC register, see **Section 2.11, Port (Special I/O) Function Overview**

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

Address: PIPCn: <PORTn_base> + 4200_H + n × 4 (n = 0, 10, 11)[Note 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 n) 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 2.41, Control Registers (P0)**, **Table 2.51, Control Registers (P10)**, and **Table 2.53, Control Registers (P11)**.

Table 2.19 PIPCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PIPCn_[15:0]	Specifies the I/O control 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: This register can be read/written in 16-bit units.
JPM0: This register can be read/written in 8-bit units.

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

Value after reset: $FFFF_H$*1

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*2
R/W	R/W	R/W	R/W	R/W	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 PM0 register is $FFFE_H$.
Note 2. The PM0 register is 0.

Note 1. The effective bit positions (value for the index n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, **Table 2.59, Control Registers (P20)**, **Table 2.61, Control Registers (AP0)**, and **Table 2.63, Control Registers (AP1)**.

Table 2.20 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.PMnm 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.
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: This register can be read/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 PMn.

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

Value after reset: 0000 $FFFF_H$*1

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*2
R/W	R/W	R/W	R/W	R/W	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 PMSR0 register is 0000 $FFFE_H$.
Note 2. The PMSR0 register is 0.

Note 1. The effective bit positions (value for the index n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, **Table 2.59, Control Registers (P20)**, **Table 2.61, Control Registers (AP0)**, and **Table 2.63, Control Registers (AP1)**.

Table 2.21 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 specifies the value of PMn_m if PMSRn_m of the corresponding upper bit (PMCSRn_[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 / IPIBCn — 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: This register can be read/written in 16-bit units.
JPIB0: This register can be read/written in 8-bit units.

Address: PIBCn: <PORTn_base> + 4000_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)
APIBCn: <PORTn_base> + $40C8_H$+ n × 4 (n = 0, 1)
JPIBC0: <JPORT0_base> + 0400_H
IPIBCn: <PORTn_base> + $40F0_H$ Note 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 n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, **Table 2.59, Control Registers (P20)**, **Table 2.61, Control Registers (AP0)**, **Table 2.63, Control Registers (AP1)**, and **Table 2.65, Control Registers (IP0)**.

Table 2.22 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 require direct I/O control of pin Pn_m. 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/written in 16-bit units.
JPFC0: This register can be read/written in 8-bit units.

Address: PFCn: <PORTn_base> + 0500_H + n × 4 (n = 0, 1, 8, 9, 10, 11, 12, 18, 20)
JPFC0: <JPORT0_base> + 0050_H **Note 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 n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, and **Table 2.59, Control Registers (P20)**.

Table 2.23 PFCn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCn_[15:0]	Specifies the alternative function of the pin. For the detail, see **Table 2.26, 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 require direct I/O control of pin Pn_m. 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/written in 16-bit units.

Address: PFCEn: <PORTn_base> + 0600_H + n × 4 (n = 0, 8, 9, 10, 11, 12)[Note 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 n) 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 2.41, Control Registers (P0)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, and **Table 2.55, Control Registers (P12)**.

Table 2.24 PFCEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCEn_[15:0]	Specifies the alternative function of the pin. For the detail, see **Table 2.26, 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 require direct I/O control of pin Pn_m. 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/written in 16-bit units.

Address: PFCAEn: <PORTn_base> + $0A00_H$ + n × 4 (n = 0, 10, 11)[Note 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 n) 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 2.41, Control Registers (P0)**, **Table 2.51, Control Registers (P10)**, and **Table 2.53, Control Registers (P11)**.

Table 2.25 PFCAEn Register Contents

Bit Position	Bit Name	Function
15 to 0	PFCAEn_[15:0]	Specifies the alternative function of the pin. For the detail, see **Table 2.26, Setting Alternative Functions**.

Table 2.26 Setting Alternative Functions

PFCAEn_m	PFCEn_m	PFCn_m	PMn_m	Function
0	0	0	1	1st alternative function / Input
			0	1st alternative function / Output
		1	1	2nd alternative function / Input
			0	2nd alternative function / Output
	1	0	1	3rd alternative function / Input
			0	3rd alternative function / Output
		1	1	4th alternative function / Input
			0	4th alternative function / Output
1	0	0	1	5th alternative function / Input
			0	5th alternative function / Output
		1	X	Setting prohibited
	1	X	X	Setting prohibited

CAUTION

- After selecting the alternative function by the PFCn_m, PFCEn_m, or PFCAEn_m register, set the PMCn_m register to “1”.
- With this product, the I/O of some functions is multiplexed in two pins, but only either one can be used as a pin function. Setting the same pin function in two pins 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 the pin function other than b.

NOTE

For more details on the assignment of each function, see **Sections 2.10.1** to **2.10.14**.

2.9.3 Pin Data Input/Output

2.9.3.1 PBDCn / APBDCn / JPBDC0 — Port Bidirection Control Register

This register enables the input buffer and sets the port to bidirectional mode. In bidirection mode, PPRn.PPRn_m can read the level of the Pn_m pin.

Access: PBDCn: This register can be read/written in 16-bit units.
JPBDC0: This register can be read/written in 8-bit units.

Address: PBDCn: <PORTn_base> + 4100_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)
APBDCn: <PORTn_base> + $41C8_H$ + n × 4 (n = 0, 1)
JPBDC0: <JPORT0_base> + 0410_H **Note 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 n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, **Table 2.59, Control Registers (P20)**, **Table 2.61, Control Registers (AP0)**, and **Table 2.63, Control Registers (AP1)**.

Table 2.27 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 input function is input to the alternative input function of the same pin (the alternative input function set by PFCn.PFCn_m, PFCEn.PFCEn_m, or PFCAEn.PFCAEn_m). *Care must be taken that output of the alternative output function and alternative input on another pin is internally input (OR input)*.

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

Access: PPRn: This register can be read only in 16-bit units.
JPPR0: This register can be read only in 8-bit units.

Address: PPRn: <PORTn_base> + 0200_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)
APPRn: <PORTn_base> + $02C8_H$ + n × 4 (n = 0, 1)
JPPR0: <JPORT0_base> + 0020_H
IPPR0: <PORTn_base> + $02F0_H$ Note 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 n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, **Table 2.59, Control Registers (P20)**, **Table 2.61, Control Registers (AP0)**, **Table 2.63, Control Registers (AP1)**, and **Table 2.65, Control Registers (IP0)**.

Table 2.28 PPRn Register Contents

Bit Position	Bit Name	Function
15 to 0	PPRn_[15:0]	Pin Pn_m, 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 data Pn.Pn_m to be output via the related port Pn_m in output port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 0).

Access: Pn: This register can be read/written in 16-bit units.
JP0: This register can be read/written in 8-bit units.

Address: Pn: <PORTn_base> + 0000_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)
APn: <PORTn_base> + $008C8_H$ n × 4 (n = 0, 1)
JP0: <JPORT0_base> + 0000_H Note 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 n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, **Table 2.59, Control Registers (P20)**, **Table 2.61, Control Registers (AP0)**, and **Table 2.63, Control Registers (AP1)**.

Table 2.29 Pn Register Contents

Bit Position	Bit Name	Function
15 to 0	Pn_[15:0]	Sets the output level of pin Pn_m (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: This register can be read only in 16-bit units. The value is always read as 0000_H.
JPNOT0: This register can be read only in 8-bit units. The value is always read as 00_H.

Address: PNOTn: <PORTn_base> + 0700_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)
APNOTn: <PORTn_base> + $07C8_H$ + n × 4 (n = 0, 1)
JPNOT0: <JPORT0_base> + 0070_H Note 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 n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, **Table 2.59, Control Registers (P20)**, **Table 2.61, Control Registers (AP0)**, and **Table 2.63, Control Registers (AP1)**.

Table 2.30 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: This register can be read/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 Pn.

Address: PSRn: <PORTn_base> + 0100_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11, 12, 18, 20)
APSRn: <PORTn_base> + $01C8_H$ + n × 4 (n = 0, 1)
JPSR0: <JPORT0_base> + 0010_H Note 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 n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, **Table 2.59, Control Registers (P20)**, **Table 2.61, Control Registers (AP0)**, and **Table 2.63, Control Registers (AP1)**.

Table 2.31 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 Registers

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/written in 16-bit units.
JPU0: This register can be read/written in 8-bit units.

Address: PUn: <PORTn_base> + 4300_H + n × 4 (n = 0, 1, 2, 8, 9, 10, 11,12, 18, 20)
JPU0: <JPORT0_base> + 0430_H **Note 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 n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, and **Table 2.59, Control Registers (P20)**.

Table 2.32 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/written in 16-bit units.
JPU0: This register can be read/written in 8-bit units.

Address: PDn: <PORTn_base> + 4400_H + n × 4 (n = 0, 8, 9, 10, 11)
JPD0: <JPORT0_base> + 0440_H Note 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 n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, and **Table 2.53, Control Registers (P11)**.

Table 2.33 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 is also related to the fast mode (high drive strength) and slow mode (low drive strength) of the output buffer. The correct write sequence using the PPCMD register is required in order to update this register. Regarding the alternative function that needs to set the PDSC register, see **Section 2.11.3.3, Output Buffer Control (PDSC)**.

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

Address: PDSCn: <PORTn_base> + 4600_H + n × 4 (n = 0, 10, 11, 12)[Note 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 n) 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 2.41, Control Registers (P0)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, and **Table 2.55, Control Registers (P12)**.

Table 2.34 PDSCn Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing to these bits, 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).

2.9.4.4 PODCn / JPODC0 — Port Open Drain Control Register

This register selects push-pull or open-drain as output buffer function. Writing to this register is protected by a special sequence of instructions. See Section **Section 2.9.5, Port Register Protection**.

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

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

Value after reset: $0000\ 0000_H$*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
	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*2
R/W	R/W	R/W	R/W	R/W	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 PODC0 register is $0000\ 0001_H$.
Note 2. The PODC0 register is 1.

Note 1. The effective bit positions (value for the index n) 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 2.39, Control Registers (JP0)**, **Table 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.47, Control Registers (P8)**, **Table 2.49, Control Registers (P9)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, **Table 2.57, Control Registers (P18)**, and **Table 2.59, Control Registers (P20)**.

Table 2.35 PODCn Register Contents

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

NOTE

The control bits of the JTAG port open drain control register (JPODC0) are JPODC0_[15:0].

2.9.4.5 PISn — Port Input Buffer Selection Register

This register specifies the input buffer characteristics.

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

Address: PISn: <PORTn_base> + 4700_H + n × 4 (n = 0, 1, 2, 10, 11, 12, 20)[Note 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 n) 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 2.41, Control Registers (P0)**, **Table 2.43, Control Registers (P1)**, **Table 2.45, Control Registers (P2)**, **Table 2.51, Control Registers (P10)**, **Table 2.53, Control Registers (P11)**, **Table 2.55, Control Registers (P12)**, and **Table 2.59, Control Registers (P20)**.

Table 2.36 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 details of type 1 and type 2 is given in **Section 2.11.3.2, Input Buffer Control (PIS)**. Also see the data sheet for input buffer characteristics.

2.9.5 Port Register Protection

RH850/F1L has Port Register Protection Command Registers (PPCMDn) and Port Protection Status Registers (PPROTSn) which implement the Port Protection Cluster Function. For details on the registers, see **Section 4, 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 = 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, either the interrupt does not occur or it is ignored because the mode temporarily becomes the alternative input mode.

2.9.6.1 Batch Setting

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

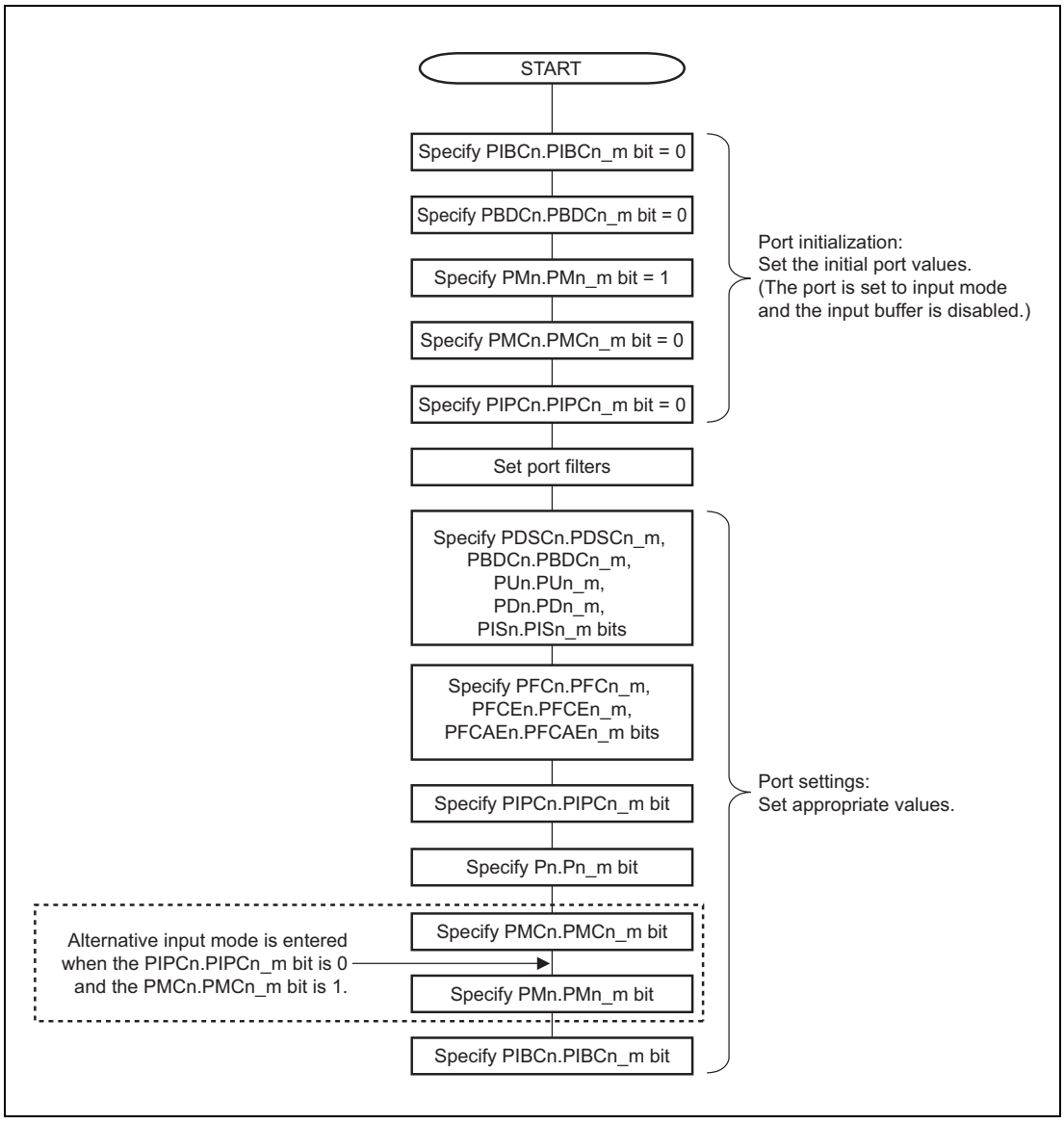


Figure 2.8 Example of Port Settings (When Specified in Batch)

2.9.6.2 Individual Settings

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

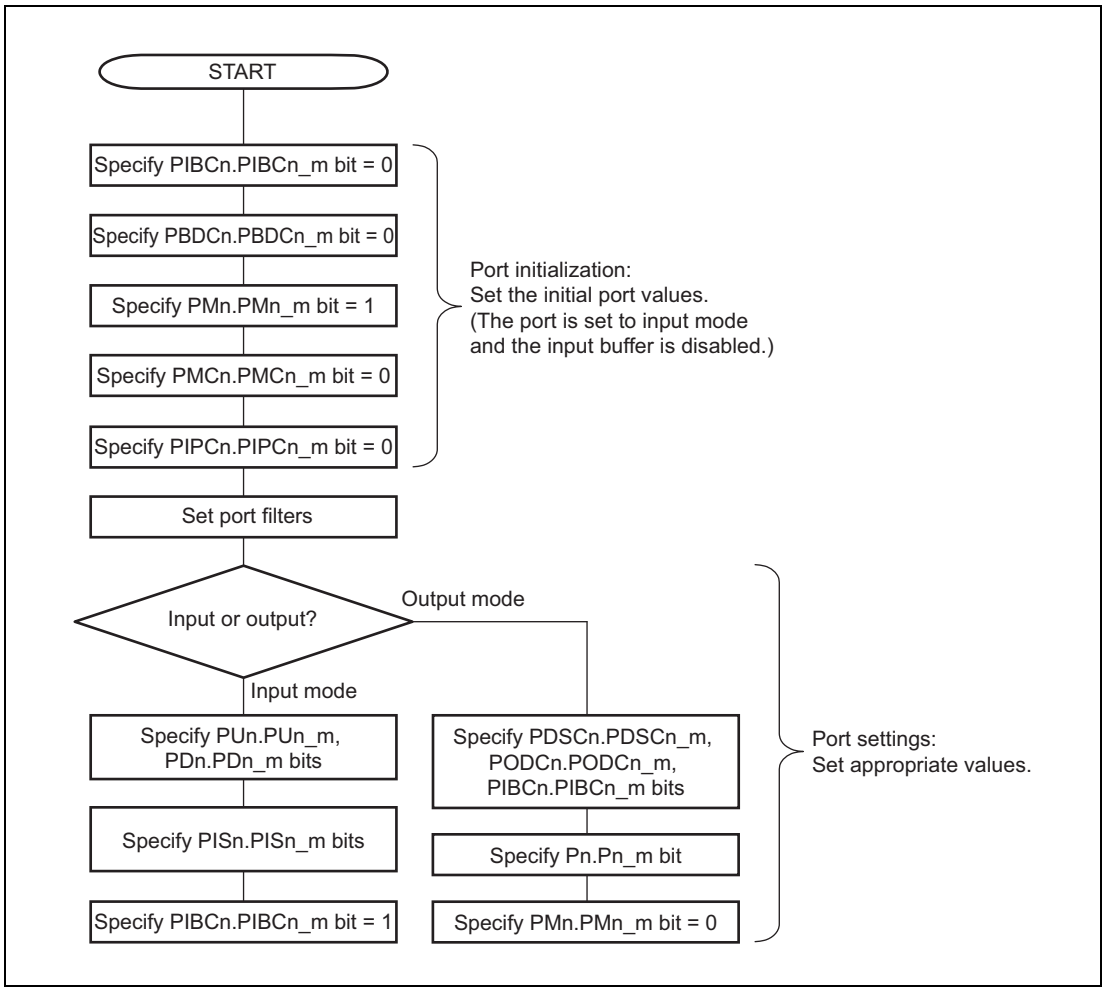


Figure 2.9 Example of Port Settings (in Port Mode)

(a) With IP control

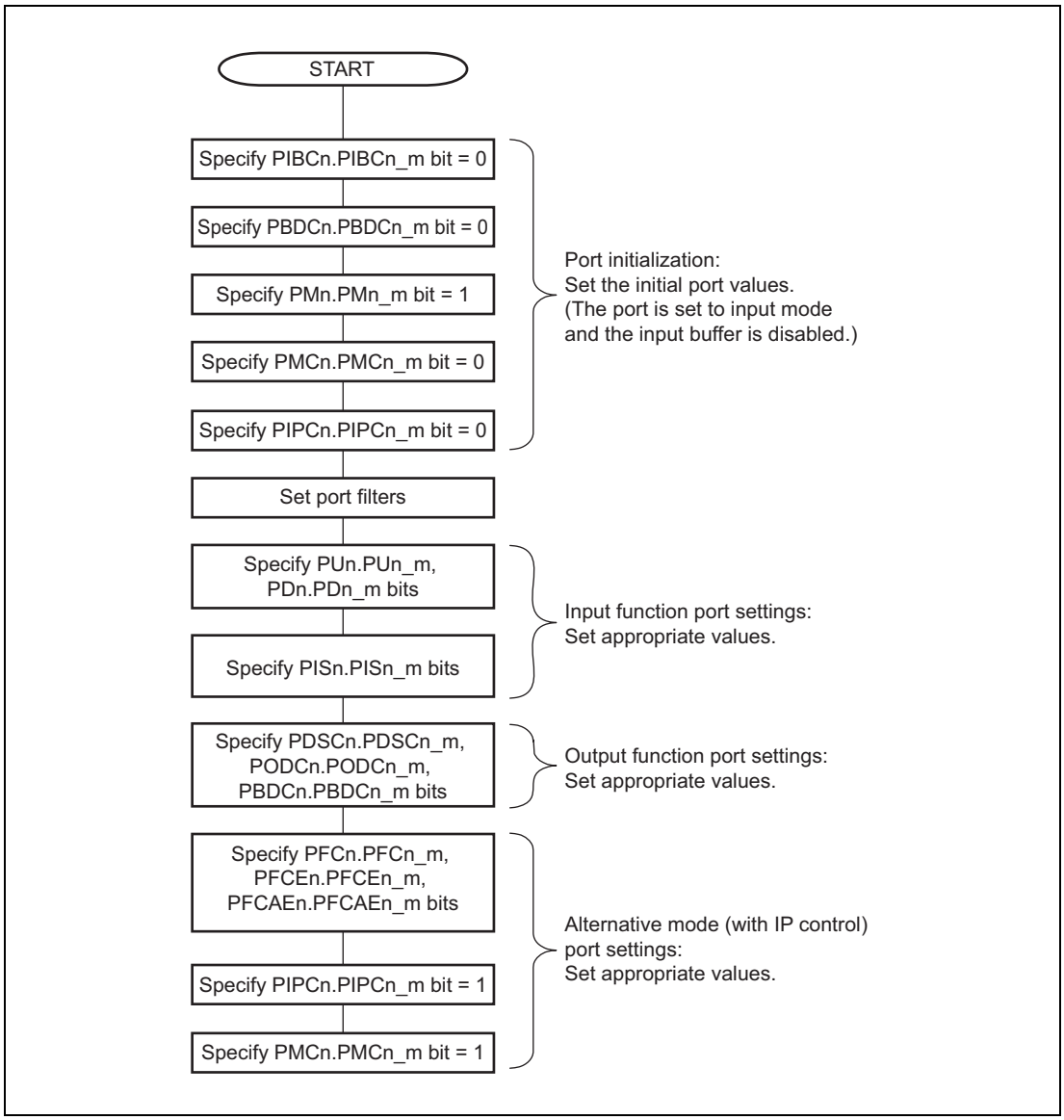


Figure 2.10 Example of Port Settings (in Alternative Mode) (1/2)

(b) Without IP control

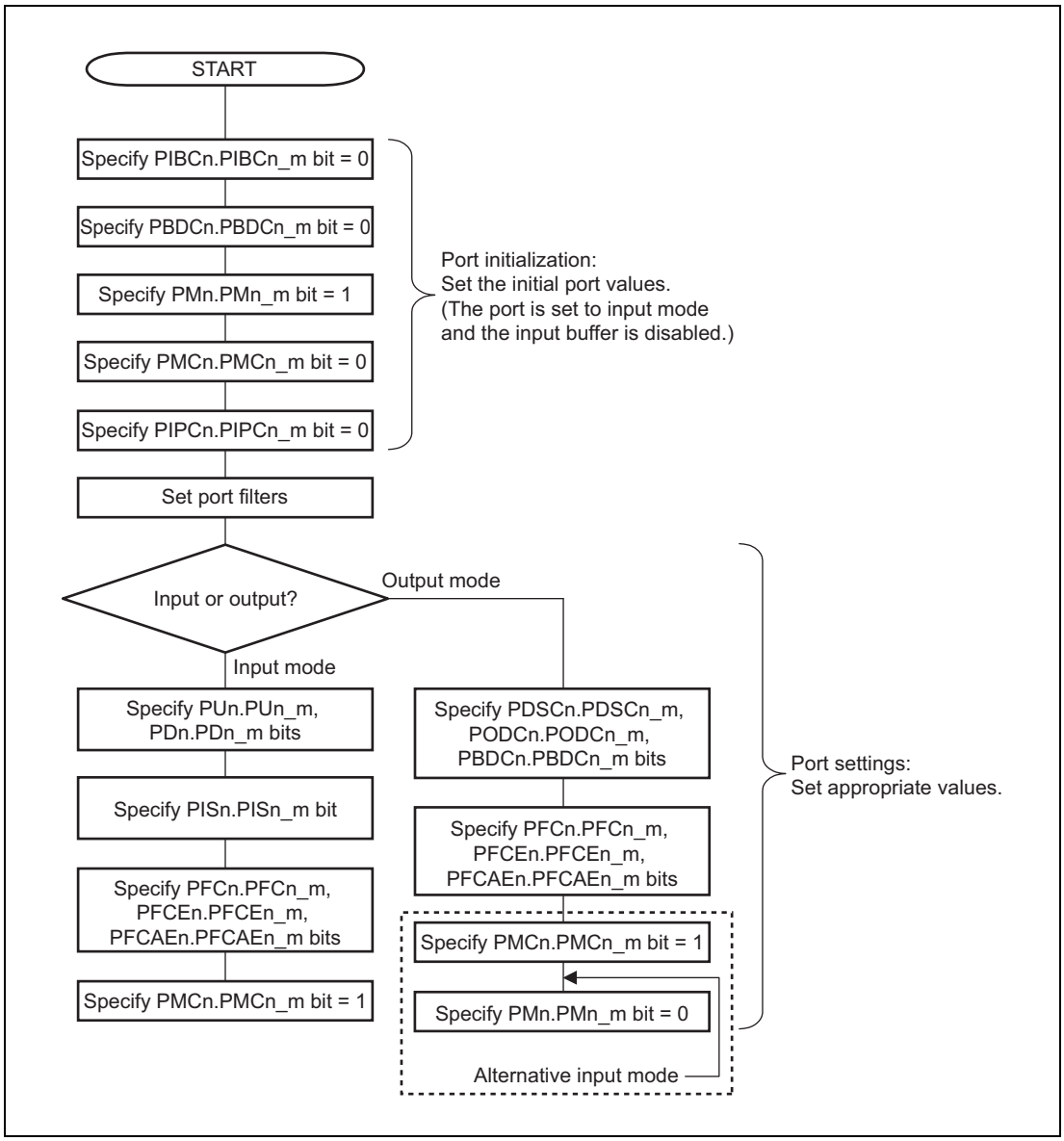


Figure 2.11 Example of Port Settings (in Alternative Mode) (2/2)

2.10 Port (General I/O) Function Overview

This section explains the port (general I/O) functionality and all the assigned functionality on 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 2.37 Port Function

Port	Pin Name	Size	Direction	Power Domain	Special Alternative Function	Device					
						48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
JTAG Port 0	JP0_0 - 5	6 bits	In/Out	AWO	JTAG, LPD	√	√	√	√	—	—
	JP0_0 - 6	7 bits				—	—	—	—	√	√
Port 0	P0_0 - 3	4 bits	In/Out	AWO	$\overline{\text{RESETOUT}}$	√	—	—	—	—	—
	P0_0 - 6	7 bits				—	√	—	—	—	—
	P0_0 - 12	13 bits				—	—	√	—	—	—
	P0_0 - 14	15 bits				—	—	—	√	√	√
Port 1	P1_0 - 11	12 bits	In/Out	AWO		—	—	—	—	√	—
	P1_0 - 15	16 bits				—	—	—	—	—	√
Port 2	P2_0 - 6	7 bits	In/Out	AWO		—	—	—	—	—	√
Port 8	P8_0 - 1	2 bits	In/Out	AWO	ADCA0 (10-bit resolution)	√	—	—	—	—	—
	P8_0 - 6	7 bits				—	√	√	—	—	—
	P8_0 - 12	13 bits				—	—	—	√	√	√
Port 9	P9_0 - 1	2 bits	In/Out	ISO	ADCA0 (10-bit resolution)	√	—	—	—	—	—
	P9_0 - 3	4 bits				—	√	—	—	—	—
	P9_0 - 6	7 bits				—	—	√	√	√	√
Port 10	P10_0 - 10	11 bits	In/Out	ISO		√	—	—	—	—	—
	P10_0 - 14	15 bits				—	√	—	—	—	—
	P10_0 - 15	16 bits				—	—	√	√	√	√
Port 11	P11_0 - 4	5 bits	In/Out	ISO		—	—	√	—	—	—
	P11_0 - 7	8 bits				—	—	—	√	—	—
	P11_0 - 15	16 bits				—	—	—	—	√	√
Port 12	P12_0 - 2	3 bits	In/Out	ISO		—	—	—	—	√	—
	P12_0 - 5	6 bits				—	—	—	—	—	√
Port 18	P18_0 - 3	4 bits	In/Out	ISO	ADCA1 (10-bit resolution)	—	—	—	—	√	—
	P18_0 - 7	8 bits				—	—	—	—	—	√
Port 20	P20_4 - 5	2 bits	In/Out	ISO		—	—	—	—	√	—
	P20_0 - 5	6 bits				—	—	—	—	—	√
Analog Port 0	AP0_0 - 7	8 bits	In/Out	AWO	ADCA0 (12/10-bit resolution)	√	—	—	—	—	—
	AP0_0 - 9	10 bits				—	√	—	—	—	—
	AP0_0 - 10	11 bits				—	—	√	—	—	—
	AP0_0 - 15	16 bits				—	—	—	√	√	√
Analog Port 1	AP1_0 - 7	8 bits	In/Out	ISO	ADCA0 (12/10-bit resolution)	—	—	—	—	√	—
	AP1_0 - 15	16 bits				—	—	—	—	—	√
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 to be a general-purpose I/O. This configuration must be done by having an appropriate flash option byte as OPJTAG = 0.

Table 2.38 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			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	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		TAUJ0I3	TAUJ0O3							$\overline{\text{DCURDY}}$/ LPDCLKOUT	√	√	√	√	—	—
	NMI	RTCA0OUT	TAUJ0I3	TAUJ0O3							$\overline{\text{DCURDY}}$/ LPDCLKOUT	—	—	—	—	√	√
JP0_6												—	—	—	—	*1	*1
											$\overline{\text{EVTO}}$	—	—	—	—	*2	*2

Note 1. Available in devices except for ones with 2-MB code flash memory.

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

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2.10.1.2 Control Registers

Table 2.39 Control Registers (JP0)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
JP0	JTAG port register 0	8	5-0	R/W	0000_H	00_H	√	√	√	√	—	—
			6-0				—	—	—	—	√	√
JPSR0	JTAG port set/reset register 0	32	21-16, 5-0	W, R/W	0010_H	$0000\ 0000_H$	√	√	√	√	—	—
			22-16, 6-0				—	—	—	—	√	√
JPPR0	JTAG port pin read register 0	8	5-0	R	0020_H	00_H	√	√	√	√	—	—
			6-0				—	—	—	—	√	√
JPM0	JTAG port mode register 0	8	5-0	R/W	0030_H	FF_H	√	√	√	√	—	—
			6-0				—	—	—	—	√	√
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	5-0	W	0070_H	00_H	√	√	√	√	—	—
			6-0				—	—	—	—	√	√
JPMSR0	JTAG port mode set/reset register 0	32	21-16, 5-0	W, R/W	0080_H	$0000\ FFFF_H$	√	√	√	√	—	—
			22-16, 6-0				—	—	—	—	√	√
JPMCSR0	JTAG port mode control set/reset register 0	32	21, 19-16, 5, 3-0	W, W, R/W, R/W	0090_H	$0000\ 0000_H$	√	√	√	√	√	√
JPIBC0	JTAG port input buffer control register 0	8	5-0	R/W	0400_H	00_H	√	√	√	√	—	—
			6-0				—	—	—	—	√	√
JPBDC0	JTAG port bidirection control register 0	8	5-0	R/W	0410_H	00_H	√	√	√	√	—	—
			6-0				—	—	—	—	√	√
JPU0	Pull-up option register 0	8	5-0	R/W	0430_H	00_H	√	√	√	√	—	—
			6-0				—	—	—	—	√	√
JPD0	Pull-down option register 0	8	5-0	R/W	0440_H	00_H	√	√	√	√	—	—
			6-0				—	—	—	—	√	√
JPODC0*1	JTAG port open drain control register 0	32	5-0	R/W	0450_H	$0000\ 0000_H$	√	√	√	√	—	—
			6-0				—	—	—	—	√	√
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	$04C0_H$	$0000\ 0000_H$	√	√	√	√	√	√

Note 1. In LPD (1-pin) mode, the JPODC0.0 bit is fixed to 1.

2.10.2 Port 0 (P0)

2.10.2.1 Alternative Function

Table 2.40 Port 0 (P0) (1/2)

Port Mode (PMC0_m = 0)	Control Mode (PMC0_m =1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
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		RLIN30TX		PWGA12O	CSIH0SC		INTP1	DPO		√	—	—	—	—	—
	TAUD0I6	TAUD0O6	CAN1RX/ INTP1	RLIN30TX		PWGA12O	CSIH0SC		INTP1	DPO		—	√	√	√	√	√
P0_3	TAUD0I8	TAUD0O8	RLIN30RX/ INTP10		DPIN1			CSIH0SO	INTP10			√	—	—	—	—	—
	TAUD0I8	TAUD0O8	RLIN30RX/ INTP10	CAN1TX	DPIN1	PWGA13O		CSIH0SO	INTP10			—	√	√	√	√	√
P0_4	RLIN31RX/ INTP11	CAN2TX	INTP11	PWGA10O		SELDP0	DPIN8					—	√	—	—	—	—
	RLIN31RX/ INTP11	CAN2TX	INTP11	PWGA10O	CSIH1SI	SELDP0	DPIN8					—	—	√	√	√	√
P0_5	CAN2RX/ INTP2	RLIN31TX	DPIN9	SELDP1								—	√	—	—	—	—
	CAN2RX/ INTP2	RLIN31TX	DPIN9	SELDP1		CSIH1SO						—	—	√	√	√	√
P0_6	INTP2		DPIN10	SELDP2								—	√	—	—	—	—
	INTP2		DPIN10	SELDP2	CSIH1SC							—	—	√	√	√	√
P0_7	RLIN21RX		DPIN5	CSCXFOUT	CSIH1RYI	CSIH1RYO						—	—	√	—	—	—
	RLIN21RX		DPIN5	CSCXFOUT	CSIH1RYI	CSIH1RYO	TAUB0I0	TAUB0O0	INTP3			—	—	—	*1	—	—
	RLIN21RX		DPIN5	CSCXFOUT	CSIH1RYI	CSIH1RYO	TAUB0I0	TAUB0O0	CAN3RX/ INTP3			—	—	—	*2	√	√
P0_8		RLIN21TX	DPIN6		$\overline{\text{CSIH1SSI}}$							—	—	√	—	—	—
		RLIN21TX	DPIN6		$\overline{\text{CSIH1SSI}}$		TAUB0I2	TAUB0O2				—	—	—	*1	—	—
		RLIN21TX	DPIN6		$\overline{\text{CSIH1SSI}}$		TAUB0I2	TAUB0O2		CAN3TX		—	—	—	*2	√	√

Table 2.40 Port 0 (P0) (2/2)

Port Mode (PMC0_m = 0)	Control Mode (PMC0_m =1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P0_9	INTP12	CSIH1CSS0	DPIN7									—	—	√	—	—	—
	INTP12	CSIH1CSS0	DPIN7		RLIN22RX		TAUB0I4	TAUB0O4	INTP4			—	—	—	*1	*3	—
	INTP12	CSIH1CSS0	DPIN7		RLIN22RX		TAUB0I4	TAUB0O4	CAN4RX/ INTP4			—	—	—	*2	*4	√
P0_10	INTP3	CSIH1CSS1	DPIN11									—	—	√	—	—	—
	INTP3	CSIH1CSS1	DPIN11			RLIN22TX	TAUB0I6	TAUB0O6				—	—	—	*1	*3	—
	INTP3	CSIH1CSS1	DPIN11			RLIN22TX	TAUB0I6	TAUB0O6		CAN4TX		—	—	—	*2	*4	√
P0_11	RIIC0SDA	RIIC0SDA		CSIH1CSS2								—	—	√	—	—	—
	RIIC0SDA	RIIC0SDA	DPIN12	CSIH1CSS2	TAUB0I8	TAUB0O8						—	—	—	√	√	—
	RIIC0SDA	RIIC0SDA	DPIN12	CSIH1CSS2	TAUB0I8	TAUB0O8	RLIN26RX					—	—	—	—	—	√
P0_12	RIIC0SCL	RIIC0SCL										—	—	√	—	—	—
	RIIC0SCL	RIIC0SCL	DPIN13	PWGA45O	TAUB0I10	TAUB0O10	CSIG0SI					—	—	—	√	√	—
	RIIC0SCL	RIIC0SCL	DPIN13	PWGA45O	TAUB0I10	TAUB0O10	CSIG0SI	RLIN26TX				—	—	—	—	—	√
P0_13	RLIN32RX/ INTP12		INTP12	PWGA46O	TAUB0I12	TAUB0O12		CSIG0SO	INTP5			—	—	—	*1	*3	—
	RLIN32RX/ INTP12		INTP12	PWGA46O	TAUB0I12	TAUB0O12		CSIG0SO	CAN5RX/ INTP5			—	—	—	*2	*4	√
P0_14		RLIN32TX		PWGA47O	TAUB0I14	TAUB0O14	CSIG0SC					—	—	—	*1	*3	—
		RLIN32TX		PWGA47O	TAUB0I14	TAUB0O14	CSIG0SC			CAN5TX		—	—	—	*2	*4	√

Note 1. Available except in F1L for Gateway

Note 2. Only available in F1L for Gateway

Note 3. Available in devices except for ones with 1.5- and 2-MB code flash memories

Note 4. Only available in devices with 1.5- and 2-MB code flash memories

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2.10.2.2 Control Registers

Table 2.41 Control Registers (P0) (1/3)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P0	Port register 0	16	3-0	R/W	0000_H	0000_H	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			12-0				—	—	√	—	—	—
			14-0				—	—	—	√	√	√
PSR0	Port set/reset register 0	32	19-16, 3-0	W, R/W	0100_H	0000 0000_H	√	—	—	—	—	—
			22-16, 6-0				—	√	—	—	—	—
			28-16, 12-0				—	—	√	—	—	—
			30-16, 14-0				—	—	—	√	√	√
PPR0	Port pin read register 0	16	3-0	R	0200_H	0000_H	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			12-0				—	—	√	—	—	—
			14-0				—	—	—	√	√	√
PM0	Port mode register 0	16	3-0	R/W	0300_H	$FFFE_H$	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			12-0				—	—	√	—	—	—
			14-0				—	—	—	√	√	√
PMC0	Port mode control register 0	16	3-0	R/W	0400_H	0000_H	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			12-0				—	—	√	—	—	—
			14-0				—	—	—	√	√	√
PFC0	Port function control register 0	16	3-0	R/W	0500_H	0000_H	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			12-0				—	—	√	—	—	—
			14-0				—	—	—	√	√	√
PFCE0	Port function control expansion register 0	16	3-0	R/W	0600_H	0000_H	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			8-0				—	—	√	—	—	—
			14-0				—	—	—	√	√	√
PNOT0	Port NOT register 0	16	3-0		0700_H	0000_H	√	—	—	—	—	—
			6-0	W			—	√	—	—	—	—
			12-0				—	—	√	—	—	—
			14-0				—	—	—	√	√	√

Table 2.41 Control Registers (P0) (2/3)

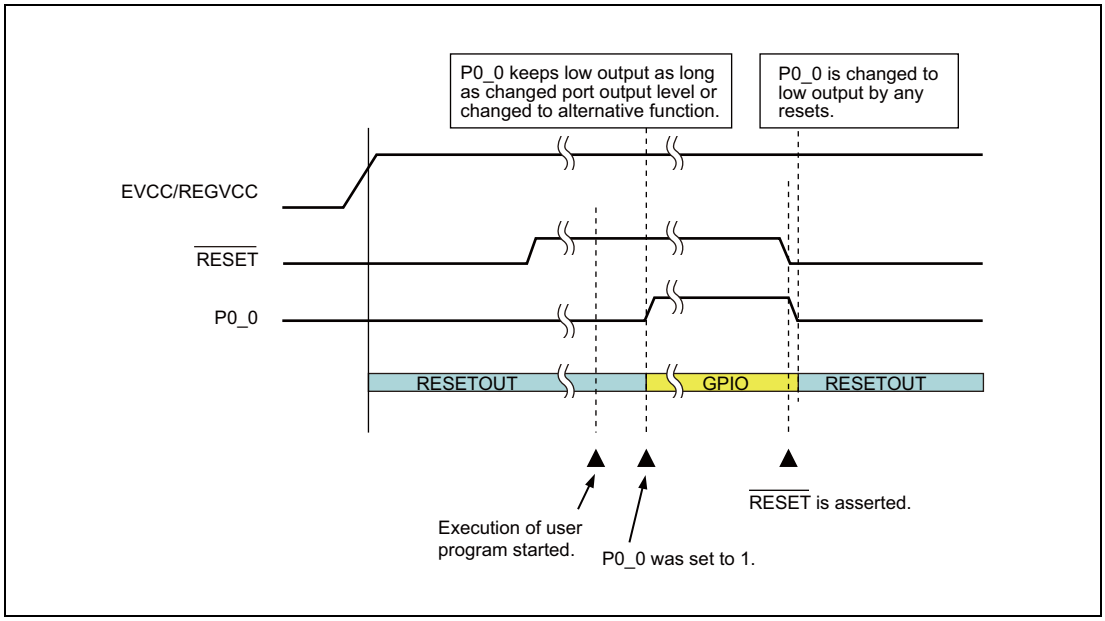
Register	Function	Register Size	Effective Bit Position	Effective Bit R/W	Offset Address	Value after Reset	Device 48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
PMSR0	Port mode set/reset register 0	32	19-16, 3-0	W, R/W	0800_H	0000 $FFFE_H$	√	—	—	—	—	—
			22-16, 6-0				—	√	—	—	—	—
			28-16, 12-0				—	—	√	—	—	—
			30-16, 14-0				—	—	—	√	√	√
PMCSR0	Port mode control set/reset register 0	32	19-16, 3-0	W, R/W	0900_H	0000 0000_H	√	—	—	—	—	—
			22-16, 6-0				—	√	—	—	—	—
			28-16, 12-0				—	—	√	—	—	—
			30-16, 14-0				—	—	—	√	√	√
PFCAE0	Port function control additional expansion register 0	16	3, 2	R/W	$0A00_H$	0000_H	√	√	√	—	—	—
			14, 13, 10-7, 3, 2				—	—	—	√	√	√
PIBC0	Port input buffer control register 0	16	3-0	R/W	4000_H	0000_H	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			12-0				—	—	√	—	—	—
			14-0				—	—	—	√	√	√
PBDC0	Port bidirection control register 0	16	3-0	R/W	4100_H	0000_H	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			12-0				—	—	√	—	—	—
			14-0				—	—	—	√	√	√
PIPC0	Port IP control register 0	16	3, 2	R/W	4200_H	0000_H	√	—	—	—	—	—
			6, 5, 3, 2				—	√	√	—	—	—
			14, 13, 6, 5, 3, 2				—	—	—	√	√	√
PU0	Pull-up option register 0	16	3-0	R/W	4300_H	0000_H	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			12-0				—	—	√	—	—	—
			14-0				—	—	—	√	√	√
PD0	Pull-down option register 0	16	3-0	R/W	4400_H	0000_H	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			12-0				—	—	√	√	√	√
PODC0	Port open drain control register 0	32	3-0	R/W	4500_H	0000 0001_H	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			12-0				—	—	√	—	—	—
			14-0				—	—	—	√	√	√
PDSC0	Port drive strength control register 0	32	3, 2	R/W	4600_H	0000 0000_H	√	—	—	—	—	—
			6, 5, 3, 2				—	√	—	—	—	—
			7-5, 3, 2				—	—	√	—	—	—
			14, 13, 7-5, 3, 2				—	—	—	√	√	√

Table 2.41 Control Registers (P0) (3/3)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
PIS0	Port input buffer selection register 0	16	3-0	R/W	4700_H	$FFFF_H$	√	—	—	—	—	—
			6-0				—	√	—	—	—	—
			12-11, 9, 7-0				—	—	√	—	—	—
			13-11, 9, 7-0				—	—	—	√	√	√
PPROTS0	Port protection status register 0	32	0	R	$4B00_H$	$0000\ 0000_H$	√	√	√	√	√	√
PPCMD0	Port protection command register 0	32	7-0	W	$4C00_H$	$0000\ 0000_H$	√	√	√	√	√	√

CAUTION

P0_0 drives a low level after any kind of reset, until it is later configured differently by register settings.
To avoid a data collision, the outside circuit connected to this pin must not drive a high level.


P0_0 keeps low output as long as changed port output level or changed to alternative function.
P0_0 is changed to low output by any resets.
EVCC/REGVCC
RESET
P0_0
RESETOUT
GPIO
RESETOUT
Execution of user program started.
P0_0 was set to 1.
RESET is asserted.

2.10.3 Port 1 (P1)

2.10.3.1 Alternative Function

Table 2.42 Port 1 (P1)

Port Mode (PMC1_m = 0)	Control Mode (PMC1_m = 1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	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 undocumented alternative functions are selected.

2.10.3.2 Control Registers

Table 2.43 Control Registers (P1)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P1	Port register 1	16	11-0	R/W	0004_H	0000_H	—	—	—	—	√	—
			15-0				—	—	—	—	—	√
PSR1	Port set/reset register 1	32	27-16, 11-0	W, R/W	0104_H	$0000\ 0000_H$	—	—	—	—	√	—
			31-16, 15-0				—	—	—	—	—	√
PPR1	Port pin read register 1	16	11-0	R	0204_H	0000_H	—	—	—	—	√	—
			15-0				—	—	—	—	—	√
PM1	Port mode register 1	16	11-0	R/W	0304_H	$FFFF_H$	—	—	—	—	√	—
			15-0				—	—	—	—	—	√
PMC1	Port mode control register 1	16	11-0	R/W	0404_H	0000_H	—	—	—	—	√	—
			15-0				—	—	—	—	—	√
PFC1	Port function control register 1	16	11-2, 0	R/W	0504_H	0000_H	—	—	—	—	√	—
			12-2, 0				—	—	—	—	—	√
PNOT1	Port NOT register 1	16	11-0	W	0704_H	0000_H	—	—	—	—	√	—
			15-0				—	—	—	—	—	√
PMSR1	Port mode set/reset register 1	32	27-16, 11-0	W, R/W	0804_H	$0000\ FFFF_H$	—	—	—	—	√	—
			31-16, 15-0				—	—	—	—	—	√
PMCSR1	Port mode control set/reset register 1	32	27-16, 11-0	W, R/W	0904_H	$0000\ 0000_H$	—	—	—	—	√	—
			31-16, 15-0				—	—	—	—	—	√
PIBC1	Port input buffer control register 1	16	11-0	R/W	4004_H	0000_H	—	—	—	—	√	—
			15-0				—	—	—	—	—	√
PBDC1	Port bidirection control register 1	16	11-0	R/W	4104_H	0000_H	—	—	—	—	√	—
			15-0				—	—	—	—	—	√
PU1	Pull-up option register 1	16	11-0	R/W	4304_H	0000_H	—	—	—	—	√	—
			15-0				—	—	—	—	—	√
PODC1	Port open drain control register 1	32	11-0	R/W	4504_H	$0000\ 0000_H$	—	—	—	—	√	—
			15-0				—	—	—	—	—	√
PIS1	Port input buffer selection register 1	16	10, 8, 6, 4, 2,0	R/W	4704_H	$FFFF_H$	—	—	—	—	√	—
			14, 12, 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	$4C04_H$	$0000\ 0000_H$	—	—	—	—	√	√

2.10.4 Port 2 (P2)

2.10.4.1 Alternative Function

Table 2.44 Port 2 (P2)

Port Mode (PMC2_m =0)	Control Mode (PMC2_m =1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	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											—	—	—	—	—	√
P2_5		RLIN29TX										—	—	—	—	—	√
P2_6												—	—	—	—	—	√

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2.10.4.2 Control Registers

Table 2.45 Control Registers (P2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P2	Port register 2	16	6-0	R/W	0008_H	0000_H	—	—	—	—	—	√
PSR2	Port set/reset register 2	32	22-16, 6-0	W, R/W	0108_H	$0000\ 0000_H$	—	—	—	—	—	√
PPR2	Port pin read register 2	16	6-0	R	0208_H	0000_H	—	—	—	—	—	√
PM2	Port mode register 2	16	6-0	R/W	0308_H	$FFFF_H$	—	—	—	—	—	√
PMC2	Port mode control register 2	16	5-0	R/W	0408_H	0000_H	—	—	—	—	—	√
PNOT2	Port NOT register 2	16	6-0	W	0708_H	0000_H	—	—	—	—	—	√
PMSR2	Port mode set/reset register 2	32	22-16, 6-0	W, R/W	0808_H	$0000\ FFFF_H$	—	—	—	—	—	√
PMCSR2	Port mode control set/reset register 2	32	21-16, 5-0	W, R/W	0908_H	$0000\ 0000_H$	—	—	—	—	—	√
PIBC2	Port input buffer control register 2	16	6-0	R/W	4008_H	0000_H	—	—	—	—	—	√
PBDC2	Port bidirection control register 2	16	6-0	R/W	4108_H	0000_H	—	—	—	—	—	√
PU2	Pull-up option register 2	16	6-0	R/W	4308_H	0000_H	—	—	—	—	—	√
PODC2	Port open drain control register 2	32	6-0	R/W	4508_H	$0000\ 0000_H$	—	—	—	—	—	√
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	$4C08_H$	$0000\ 0000_H$	—	—	—	—	—	√

2.10.5 Port 8 (P8)

2.10.5.1 Alternative Function

Table 2.46 Port 8 (P8)

Port Mode (PMC8_m = 0)	Control Mode (PMC8_m = 1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P8_0	TAUJ0I0	TAUJ0O0	DPIN2		INTP4	CSIH0CSS0					ADCA0I0S	√	—	—	—	—	—
	TAUJ0I0	TAUJ0O0	DPIN2	PWGA14O	INTP4	CSIH0CSS0					ADCA0I0S	—	√	√	√	√	√
P8_1	TAPA0ESO	TAUJ0O1	DPIN0		INTP5						ADCA0I1S	√	—	—	—	—	—
	TAPA0ESO	TAUJ0O1	DPIN0	PWGA15O	INTP5						ADCA0I1S	—	√	—	—	—	—
	TAPA0ESO	TAUJ0O1	DPIN0	PWGA15O	INTP5	CSIH1CSS3					ADCA0I1S	—	—	√	√	√	√
P8_2	TAUJ0I0	TAUJ0O0	DPIN2	CSIH0CSS0		PWGA22O					ADCA0I4S	—	√	—	—	—	—
	TAUJ0I0	TAUJ0O0	DPIN2	CSIH0CSS0	INTP6	PWGA22O					ADCA0I4S	—	—	√	√	√	√
P8_3	TAUJ0I1	TAUJ0O1	DPIN3	CSIH0CSS1		PWGA23O					ADCA0I5S	—	√	—	—	—	—
	TAUJ0I1	TAUJ0O1	DPIN3	CSIH0CSS1	INTP7	PWGA23O					ADCA0I5S	—	—	√	√	√	√
P8_4	TAUJ0I2	TAUJ0O2	DPIN4	CSIH0CSS2							ADCA0I6S	—	√	—	—	—	—
	TAUJ0I2	TAUJ0O2	DPIN4	CSIH0CSS2	INTP8						ADCA0I6S	—	—	√	—	—	—
	TAUJ0I2	TAUJ0O2	DPIN4	CSIH0CSS2	INTP8	PWGA36O					ADCA0I6S	—	—	—	√	√	√
P8_5	TAUJ0I3	TAUJ0O3		CSIH0CSS3							ADCA0I7S	—	√	√	—	—	—
	TAUJ0I3	TAUJ0O3		CSIH0CSS3		PWGA37O					ADCA0I7S	—	—	—	√	—	—
	TAUJ0I3	TAUJ0O3		CSIH0CSS3	INTP9	PWGA37O					ADCA0I7S	—	—	—	—	√	√
P8_6	NMI										ADCA0I8S	—	√	—	—	—	—
	NMI	CSIH0CSS4									ADCA0I8S	—	—	√	—	—	—
	NMI	CSIH0CSS4		PWGA38O							ADCA0I8S	—	—	—	√	—	—
	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 undocumented alternative functions are selected.
2. Use the special function with initial settings.

2.10.5.2 Control Registers

Table 2.47 Control Registers (P8) (1/2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P8	Port register 8	16	1, 0	R/W	0020_H	0000_H	√	—	—	—	—	—
			6-0				—	√	√	—	—	—
			12-0				—	—	—	√	√	√
PSR8	Port set/reset register 8	32	17, 16, 1, 0	W, R/W	0120_H	$0000\ 0000_H$	√	—	—	—	—	—
			22-16, 6-0				—	√	√	—	—	—
			28-16, 12-0				—	—	—	√	√	√
PPR8	Port pin read register 8	16	1, 0	R	0220_H	0000_H	√	—	—	—	—	—
			6-0				—	√	√	—	—	—
			12-0				—	—	—	√	√	√
PM8	Port mode register 8	16	1, 0	R/W	0320_H	$FFFF_H$	√	—	—	—	—	—
			6-0				—	√	√	—	—	—
			12-0				—	—	—	√	√	√
PMC8	Port mode control register 8	16	1, 0	R/W	0420_H	0000_H	√	—	—	—	—	—
			6-0				—	√	√	—	—	—
			12-0				—	—	—	√	√	√
PFC8	Port function control register 8	16	1, 0	R/W	0520_H	0000_H	√	—	—	—	—	—
			5-0				—	√	√	—	—	—
			12-0				—	—	—	√	√	√
PFCE8	Port function control expansion register 8	16	1, 0	R/W	0620_H	0000_H	√	—	—	—	—	—
			3-0				—	√	—	—	—	—
			4-0				—	—	√	—	—	—
			12, 11, 5-0				—	—	—	√	—	—
			12, 11, 6-0				—	—	—	—	√	√
PNOT8	Port NOT register 8	16	1, 0	W	0720_H	0000_H	√	—	—	—	—	—
			6-0				—	√	√	—	—	—
			12-0				—	—	—	√	√	√
PMSR8	Port mode set/reset register 8	32	17, 16, 1, 0	W, R/W	0820_H	$0000\ FFFF_H$	√	—	—	—	—	—
			22-16, 6-0				—	√	√	—	—	—
			28-16, 12-0				—	—	—	√	√	√
PMCSR8	Port mode control set/reset register 8	32	17, 16, 1, 0	W, R/W	0920_H	$0000\ 0000_H$	√	—	—	—	—	—
			22-16, 6-0				—	√	√	—	—	—
			28-16, 12-0				—	—	—	√	√	√

Table 2.47 Control Registers (P8) (2/2)

Register	Function	Register Size	Effective Bit Position	Effective Bit R/W	Offset Address	Value after Reset	Device 48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
PIBC8	Port input buffer control register 8	16	1, 0	R/W	4020_H	0000_H	√	—	—	—	—	—
			6-0				—	√	√	—	—	—
			12-0				—	—	—	√	√	√
PBDC8	Port bidirection control register 8	16	1, 0	R/W	4120_H	0000_H	√	—	—	—	—	—
			6-0				—	√	√	—	—	—
			12-0				—	—	—	√	√	√
PU8	Pull-up option register 8	16	1, 0	R/W	4320_H	0000_H	√	—	—	—	—	—
			6-0				—	√	√	—	—	—
			12-0				—	—	—	√	√	√
PD8	Pull-down option register 8	16	1, 0	R/W	4420_H	0000_H	√	—	—	—	—	—
			6-0				—	√	√	√	√	√
PODC8	Port open drain control register 8	32	1, 0	R/W	4520_H	$0000\ 0000_H$	√	—	—	—	—	—
			6-0				—	√	√	—	—	—
			12-0				—	—	—	√	√	√
PPROTS8	Port protection status register 8	32	0	R	$4B20_H$	$0000\ 0000_H$	√	√	√	√	√	√
PPCMD8	Port protection command register 8	32	7-0	W	$4C20_H$	0000_0000_H	√	√	√	√	√	√

2.10.6 Port 9 (P9)

2.10.6.1 Alternative Function

Table 2.48 Port 9 (P9)

Port Mode (PMC9_m = 0)	Control Mode (PMC9_m = 1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P9_0	NMI	PWGA8O	TAUD0I0	TAUD0O0	ADCA0TRG0		KR0I4				ADCA0I2S	√	√	—	—	—	—
	NMI	PWGA8O	TAUD0I0	TAUD0O0	ADCA0TRG0	CSIH2CSS0	KR0I4				ADCA0I2S	—	—	√	√	√	√
P9_1	INTP11	PWGA9O	TAUD0I2	TAUD0O2	KR0I5						ADCA0I3S	√	√	—	—	—	—
	INTP11	PWGA9O	TAUD0I2	TAUD0O2	KR0I5	CSIH2CSS1					ADCA0I3S	—	—	√	√	√	√
P9_2	KR0I6	PWGA20O	TAPA0ESO								ADCA0I9S	—	√	—	—	—	—
	KR0I6	PWGA20O	TAPA0ESO	CSIH2CSS2							ADCA0I9S	—	—	√	√	√	√
P9_3	KR0I7	PWGA21O									ADCA0I10S	—	√	—	—	—	—
	KR0I7	PWGA21O		CSIH2CSS3							ADCA0I10S	—	—	√	√	√	√
P9_4		CSIH0CSS5									ADCA0I11S	—	—	√	—	—	—
		CSIH0CSS5		PWGA33O	TAUJ1I0	TAUJ1O0					ADCA0I11S	—	—	—	√	√	√
P9_5		CSIH0CSS6									ADCA0I12S	—	—	√	—	—	—
		CSIH0CSS6		PWGA34O	TAUJ1I1	TAUJ1O1					ADCA0I12S	—	—	—	√	√	√
P9_6		CSIH0CSS7									ADCA0I13S	—	—	√	—	—	—
		CSIH0CSS7		PWGA35O							ADCA0I13S	—	—	—	√	√	√

CAUTIONS

1. The behavior and performance are not guaranteed when undocumented alternative functions are selected.
2. Use the special function with initial settings.

2.10.6.2 Control Registers

Table 2.49 Control Registers (P9) (1/2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P9	Port register 9	16	1, 0	R/W	0024_H	0000_H	√	—	—	—	—	—
			3-0				—	√	—	—	—	—
			6-0				—	—	√	√	√	√
PSR9	Port set/reset register 9	32	17, 16, 1, 0	W, R/W	0124_H	$0000\ 0000_H$	√	—	—	—	—	—
			19-16, 3-0				—	√	—	—	—	—
			22-16, 6-0				—	—	√	√	√	√
PPR9	Port pin read register 9	16	1, 0	R	0224_H	0000_H	√	—	—	—	—	—
			3-0				—	√	—	—	—	—
			6-0				—	—	√	√	√	√
PM9	Port mode register 9	16	1, 0	R/W	0324_H	$FFFF_H$	√	—	—	—	—	—
			3-0				—	√	—	—	—	—
			6-0				—	—	√	√	√	√
PMC9	Port mode control register 9	16	1, 0	R/W	0424_H	0000_H	√	—	—	—	—	—
			3-0				—	√	—	—	—	—
			6-0				—	—	√	√	√	√
PFC9	Port function control register 9	16	1, 0	R/W	0524_H	0000_H	√	—	—	—	—	—
			2-0				—	√	—	—	—	—
			3-0				—	—	√	—	—	—
			6-0				—	—	—	√	√	√
PFCE9	Port function control expansion register 9	16	1.0	R/W	0624_H	0000_H		√	√	—	—	—
			5.4.1.0				—	—	—	√	√	√
PNOT9	Port NOT register 9	16	1, 0	W	0724_H	0000_H	√	—	—	—	—	—
			3-0				—	√	—	—	—	—
			6-0				—	—	√	√	√	√
PMSR9	Port mode set/reset register 9	32	17, 16, 1, 0	W, R/W	0824_H	$0000\ FFFF_H$	√	—	—	—	—	—
			19-16, 3-0				—	√	—	—	—	—
			22-16, 6-0				—	—	√	√	√	√
PMCSR9	Port mode control set/reset register 9	32	17, 16, 1, 0	W, R/W	0924_H	$0000\ 0000_H$	√	—	—	—	—	—
			19-16, 3-0				—	√	—	—	—	—
			22-16, 6-0				—	—	√	√	√	√
PIBC9	Port input buffer control register 9	16	1, 0	R/W	4024_H	0000_H	√	—	—	—	—	—
			3-0				—	√	—	—	—	—
			6-0				—	—	√	√	√	√
PBDC9	Port bidirection control register 9	16	1, 0	R/W	4124_H	0000_H	√	—	—	—	—	—
			3-0				—	√	—	—	—	—
			6-0				—	—	√	√	√	√

Table 2.49 Control Registers (P9) (2/2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
PU9	Pull-up option register 9	16	1, 0	R/W	4324_H	0000_H	√	—	—	—	—	—
			3-0				—	√	—	—	—	—
			6-0				—	—	√	√	√	√
PD9	Pull-down option register 9	16	1, 0	R/W	4424_H	0000_H	√	—	—	—	—	—
			3-0				—	√	—	—	—	—
			6-0				—	—	√	√	√	√
PODC9	Port open drain control register 9	32	1, 0	R/W	4524_H	$0000\ 0000_H$	√	—	—	—	—	—
			3-0				—	√	—	—	—	—
			6-0				—	—	√	√	√	√
PPROTS9	Port protection status register 9	32	0	R	$4B24_H$	$0000\ 0000_H$	√	√	√	√	√	√
PPCMD9	Port protection command register 9	32	7-0	W	$4C24_H$	$0000\ 0000_H$	√	√	√	√	√	√

2.10.7 Port 10 (P10)

2.10.7.1 Alternative Function

Table 2.50 Port 10 (P10) (1/2)

Port Mode (PMC10_m = 0)	Control Mode (PMC10_m = 1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P10_0	TAUD0I1	TAUD0O1	CAN0RX/ INTP0	CSCXFOUT		PWGA0O		TAPA0UP				√	√	—	—	—	—
	TAUD0I1	TAUD0O1	CAN0RX/ INTP0	CSCXFOUT		PWGA0O		TAPA0UP	CSIH1SI			—	—	√	√	√	—
	TAUD0I1	TAUD0O1	CAN0RX/ INTP0	CSCXFOUT		PWGA0O		TAPA0UP	CSIH1SI	MEMC0 A19		—	—	—	—	—	√
P10_1	TAUD0I3	TAUD0O3		CAN0TX		PWGA1O		TAPA0UN			MODE0	√	√	—	—	—	—
	TAUD0I3	TAUD0O3		CAN0TX		PWGA1O		TAPA0UN	CSIH1SC		MODE0	—	—	√	√	√	√
P10_2	TAUD0I5	TAUD0O5		RIIC0SDA	KR0I0	PWGA2O	ADCA0TRG0	TAPA0VP			MODE1	√	√	—	—	—	—
	TAUD0I5	TAUD0O5		RIIC0SDA	KR0I0	PWGA2O	ADCA0TRG0	TAPA0VP		CSIH1SO	MODE1	—	—	√	√	√	√
P10_3	TAUD0I7	TAUD0O7		RIIC0SCL	KR0I1	PWGA3O	ADCA0TRG1	TAPA0VN				√	√	—	—	—	—
	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						√	—	—	—	—	—
	TAUD0I13	TAUD0O13		CSIG0SO	ENCA0TIN0	ADCA0SEL2	CAN1RX/ INTP1					—	√	√	√	√	—
	TAUD0I13	TAUD0O13		CSIG0SO	ENCA0TIN0	ADCA0SEL2	CAN1RX/ INTP1		MEMC0AD0			—	—	—	—	—	√
P10_7	TAUD0I15	TAUD0O15	CSIG0SC		ENCA0TIN1	PWGA4O						√	—	—	—	—	—
	TAUD0I15	TAUD0O15	CSIG0SC		ENCA0TIN1	PWGA4O		CAN1TX				—	√	√	√	√	—
	TAUD0I15	TAUD0O15	CSIG0SC		ENCA0TIN1	PWGA4O		CAN1TX	MEMC0AD1			—	—	—	—	—	√
P10_8	TAUD0I10	TAUD0O10	CSIG0SI		ENCA0EC	PWGA5O					FLMD1	√	√	√	√	√	—
	TAUD0I10	TAUD0O10	CSIG0SI		ENCA0EC	PWGA5O	MEMC0AD2				FLMD1	—	—	—	—	—	√
P10_9	TAUD0I12	TAUD0O12	RLIN30RX/ INTP10		ENCA0E0	PWGA6O	CSIH0RYI	CSIH0RYO				√	√	√	√	√	—
	TAUD0I12	TAUD0O12	RLIN30RX/ INTP10		ENCA0E0	PWGA6O	CSIH0RYI	CSIH0RYO	MEMC0AD3			—	—	—	—	—	√

Table 2.50 Port 10 (P10) (2/2)

Port Mode (PMC10_m = 0)	Control Mode (PMC10_m = 1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P10_10	TAUD0I14	TAUD0O14		RLIN30TX	ENCA0E1	PWGA7O		CSIH0CSS1				√	√	√	√	√	—
	TAUD0I14	TAUD0O14		RLIN30TX	ENCA0E1	PWGA7O		CSIH0CSS1	MEMC0AD4			—	—	—	—	—	√
P10_11		PWGA16O	RLIN31RX/ INTP11									—	√	—	—	—	—
		PWGA16O	RLIN31RX/ INTP11			CSIH1CSS0						—	—	√	—	—	—
		PWGA16O	RLIN31RX/ INTP11			CSIH1CSS0	TAUB0I1	TAUB0O1				—	—	—	√	√	—
		PWGA16O	RLIN31RX/ INTP11			CSIH1CSS0	TAUB0I1	TAUB0O1	MEMC0AD5			—	—	—	—	—	√
P10_12		PWGA17O		RLIN31TX								—	√	—	—	—	—
		PWGA17O		RLIN31TX		CSIH1CSS1						—	—	√	—	—	—
		PWGA17O		RLIN31TX		CSIH1CSS1	TAUB0I3	TAUB0O3				—	—	—	√	√	—
		PWGA17O		RLIN31TX		CSIH1CSS1	TAUB0I3	TAUB0O3	MEMC0AD6			—	—	—	—	—	√
P10_13	$\overline{\text{CSIH0SSI}}$	PWGA18O										—	√	—	—	—	—
	$\overline{\text{CSIH0SSI}}$	PWGA18O	RLIN32RX/ INTP12									—	—	√	—	—	—
	$\overline{\text{CSIH0SSI}}$	PWGA18O	RLIN32RX/ INTP12				TAUB0I5	TAUB0O5				—	—	—	√	√	—
	$\overline{\text{CSIH0SSI}}$	PWGA18O	RLIN32RX/ INTP12				TAUB0I5	TAUB0O5	MEMC0AD7			—	—	—	—	—	√
P10_14		PWGA19O										—	√	—	—	—	—
		PWGA19O		RLIN32TX								—	—	√	—	—	—
		PWGA19O		RLIN32TX	$\overline{\text{CSIH3SSI}}$		TAUB0I7	TAUB0O7				—	—	—	√	—	—
	ADCA1TRG0	PWGA19O		RLIN32TX	$\overline{\text{CSIH3SSI}}$		TAUB0I7	TAUB0O7				—	—	—	—	√	—
	ADCA1TRG0	PWGA19O		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 undocumented alternative functions are selected.

2.10.7.2 Control Registers

Table 2.51 Control Registers (P10) (1/3)

Register	Function	Register Size	Effective Bit Position	Effective Bit R/W	Offset Address	Value after Reset	Device 48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P10	Port register 10	16	10-0	R/W	0028_H	0000_H	√	—	—	—	—	—
			14-0				—	√	—	—	—	—
			15-0				—	—	√	√	√	√
PSR10	Port set/reset register 10	32	26-16, 10-0	W, R/W	0128_H	$0000\ 0000_H$	√	—	—	—	—	—
			30-16, 14-0				—	√	—	—	—	—
			31-16, 15-0				—	—	√	√	√	√
PPR10	Port pin read register 10	16	10-0	R	0228_H	0000_H	√	—	—	—	—	—
			14-0				—	√	—	—	—	—
			15-0				—	—	√	√	√	√
PM10	Port mode register 10	16	10-0	R/W	0328_H	$FFFF_H$	√	—	—	—	—	—
			14-0				—	√	—	—	—	—
			15-0				—	—	√	√	√	√
PMC10	Port mode control register 10	16	10-0	R/W	0428_H	0000_H	√	—	—	—	—	—
			14-0				—	√	√	—	—	—
			15-0				—	—	—	√	√	√
PFC10	Port function control register 10	16	10-0	R/W	0528_H	0000_H	√	—	—	—	—	—
			12-0				—	√	—	—	—	—
			14-0				—	—	√	—	—	—
			15-0				—	—	—	√	√	√
PFCE10	Port function control expansion register 10	16	10-0	R/W	0628_H	0000_H	√	—	—	—	—	—
			14, 13, 10-0				—	√	—	—	—	—
			14-0				—	—	√	—	—	—
			15-0				—	—	—	√	√	√
PNOT10	Port NOT register 10	16	10-0	W	0728_H	0000_H	√	—	—	—	—	—
			14-0				—	√	—	—	—	—
			15-0				—	—	√	√	√	√
PMSR10	Port mode set/reset register 10	32	25-16, 10-0	W, R/W	0828_H	$0000\ FFFF_H$	√	—	—	—	—	—
			30-16, 14-0				—	√	—	—	—	—
			31-16, 15-0				—	—	√	√	√	√

Table 2.51 Control Registers (P10) (2/3)

Register	Function	Register Size	Effective Bit Position	Effective Bit R/W	Offset Address	Value after Reset	Device 48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
PMCSR10	Port mode control set/reset register 10	32	26-16, 10-0	W, R/W	0928_H	$0000\ 0000_H$	√	—	—	—	—	—
			30-16, 14-0				—	√	√	—	—	—
			31-16, 15-0				—	—	—	√	√	√
PFCAE10	Port function control additional expansion register 10	16	5, 4	R/W	$0A28_H$	0000_H	√	√	—	—	—	—
			5-0				—	—	√	√	√	—
			15-9, 7-0				—	—	—	—	—	√
PIBC10	Port input buffer control register 10	16	10-0	R/W	4028_H	0000_H	√	—	—	—	—	—
			14-0				—	√	—	—	—	—
			15-0				—	—	√	√	√	√
PBDC10	Port bidirection control register 10	16	10-0	R/W	4128_H	0000_H	√	—	—	—	—	—
			14-0				—	√	—	—	—	—
			15-0				—	—	√	√	√	√
PIPC10	Port IP control register 10	16	7-0	R/W	4228_H	0000_H	√	√	√	√	√	—
			14-0				—	—	—	—	—	√
PU10	Pull-up option register 10	16	10-0	R/W	4328_H	0000_H	√	—	—	—	—	—
			14-0				—	√	—	—	—	—
			15-0				—	—	√	√	√	√
PD10	Pull-down option register 10	16	10-0	R/W	4428_H	0000_H	√	—	—	—	—	—
			14-0				—	√	—	—	—	—
			15-0				—	—	√	√	√	√
PODC10	Port open drain control register 10	32	10-0	R/W	4528_H	$0000\ 0000_H$	√	—	—	—	—	—
			14-0				—	√	—	—	—	—
			15-0				—	—	√	√	√	√
PDSC10	Port drive strength control register 10	32	10-0	R/W	4628_H	$0000\ 0000_H$	√	—	—	—	—	—
			14-0				—	√	—	—	—	—
			15-0				—	—	√	√	√	√

Table 2.51 Control Registers (P10) (3/3)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
PIS10	Port input buffer selection register 10	16	9, 6, 4-2, 0	R/W	4728_H	$FFFF_H$	√	—	—	—	—	—
			13, 11, 9, 6, 4-2, 0				—	√	—	—	—	—
			15, 13, 11, 9, 6, 4-2, 0				—	—	√	√	√	√
PPROTS10	Port protection status register 10	32	0	R	$4B28_H$	$0000\ 0000_H$	√	√	√	√	√	√
PPCMD10	Port protection command register 10	32	7-0	W	$4C28_H$	$0000\ 0000_H$	√	√	√	√	√	√

2.10.8 Port 11 (P11)

2.10.8.1 Alternative Function

Table 2.52 Port 11 (P11) (1/2)

Port mode (PMC11_m = 0)	Control Mode (PMC11_m = 1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P11_0	CSIH2RYI	CSIH2RYO										—	—	√	—	—	—
	CSIH2RYI	CSIH2RYO		PWGA25O		RLIN22TX	TAUB0I11	TAUB0O11				—	—	—	√	—	—
	CSIH2RYI	CSIH2RYO	ADCA1TRG2	PWGA25O		RLIN22TX	TAUB0I11	TAUB0O11				—	—	—	—	√	—
	CSIH2RYI	CSIH2RYO	ADCA1TRG2	PWGA25O		RLIN22TX	TAUB0I11	TAUB0O11		$\overline{\text{MEMC0WR}}$		—	—	—	—	—	√
P11_1	$\overline{\text{CSIH2SSI}}$		RLIN20RX									—	—	√	—	—	—
	$\overline{\text{CSIH2SSI}}$		RLIN20RX			PWGA26O	TAUB0I13	TAUB0O13				—	—	—	√	√	—
	$\overline{\text{CSIH2SSI}}$		RLIN20RX			PWGA26O	TAUB0I13	TAUB0O13	MEMC0AD9			—	—	—	—	—	√
P11_2		CSIH2SO		RLIN20TX								—	—	√	—	—	—
		CSIH2SO		RLIN20TX		PWGA27O	TAUB0I15	TAUB0O15				—	—	—	√	√	—
		CSIH2SO		RLIN20TX		PWGA27O	TAUB0I15	TAUB0O15	MEMC0AD10			—	—	—	—	—	√
P11_3	CSIH2SC											—	—	√	—	—	—
	CSIH2SC		INTP3	PWGA28O								—	—	—	*1	—	—
	CSIH2SC		CAN3RX/ INTP3	PWGA28O								—	—	—	*2	√	—
	CSIH2SC		CAN3RX/ INTP3	PWGA28O	TAUB1I1	TAUB1O1	MEMC0AD11					—	—	—	—	—	√
P11_4	CSIH2SI											—	—	√	—	—	—
	CSIH2SI			PWGA29O								—	—	—	*1	—	—
	CSIH2SI	CAN3TX		PWGA29O								—	—	—	*2	√	—
	CSIH2SI	CAN3TX		PWGA29O	TAUB1I3	TAUB1O3	MEMC0AD12					—	—	—	—	—	√
P11_5	INTP5	RLIN33TX		PWGA30O	CSIH3SI							—	—	—	*1	*3	—
	CAN5RX/ INTP5	RLIN33TX		PWGA30O	CSIH3SI							—	—	—	*2	*4	—
	CAN5RX/ INTP5	RLIN33TX		PWGA30O	CSIH3SI		TAUB1I5	TAUB1O5	MEMC0AD13			—	—	—	—	—	√

Table 2.52 Port 11 (P11) (2/2)

Port mode (PMC11_m = 0)	Control Mode (PMC11_m = 1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P11_6	RLIN33RX/ INTP13			PWGA31O		CSIH3SO						—	—	—	*1	—	—
	RLIN33RX/ INTP13	CAN5TX		PWGA31O		CSIH3SO						—	—	—	*2	—	—
	RLIN33RX/ INTP13		ADCA1TRG1	PWGA31O		CSIH3SO						—	—	—	—	*3	—
	RLIN33RX/ INTP13	CAN5TX	ADCA1TRG1	PWGA31O		CSIH3SO						—	—	—	—	*4	—
	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		$\overline{\text{MEMC0CS3}}$				—	—	—	—	—	√
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		$\overline{\text{MEMC0ASTB}}$				—	—	—	—	—	√

Note 1. Available except in F1L for Gateway

Note 2. Only available in F1L for Gateway

Note 3. Available in devices except for ones with 1.5- and 2-MB code flash memories
Note 4. Only available in devices with 1.5- and 2-MB code flash memories

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2.10.8.2 Control Registers

Table 2.53 Control Registers (P11) (1/2)

Register	Function	Register Size	Effective Bit Position	Effective Bit R/W	Offset Address	Value after Reset	Device 48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P11	Port register 11	16	4-0	R/W	$002C_H$	0000_H	—	—	√	—	—	—
			7-0				—	—	—	√	—	—
			15-0				—	—	—	—	√	√
PSR11	Port set/reset register 11	32	20-16, 4-0	W, R/W	$012C_H$	$0000\ 0000_H$	—	—	√	—	—	—
			23-16, 7-0				—	—	—	√	—	—
			31-16, 15-0				—	—	—	—	√	√
PPR11	Port pin read register 11	16	4-0	R	$022C_H$	0000_H	—	—	√	—	—	—
			7-0				—	—	—	√	—	—
			15-0				—	—	—	—	√	√
PM11	Port mode register 11	16	4-0	R/W	$032C_H$	$FFFF_H$	—	—	√	—	—	—
			7-0				—	—	—	√	—	—
			15-0				—	—	—	—	√	√
PMC11	Port mode control register 11	16	4-0	R/W	$042C_H$	0000_H	—	—	√	—	—	—
			7-0				—	—	—	√	—	—
			15-0				—	—	—	—	√	√
PFC11	Port function control register 11	16	2, 1	R/W	$052C_H$	0000_H	—	—	√	—	—	—
			7-0				—	—	—	√	—	—
			15-0				—	—	—	—	√	√
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	4-0	W	$072C_H$	0000_H	—	—	√	—	—	—
			7-0				—	—	—	√	—	—
			15-0				—	—	—	—	√	√
PMSR11	Port mode set/reset register 11	32	20-16, 4-0	W, R/W	$082C_H$	$0000\ FFFF_H$	—	—	√	—	—	—
			23-16, 7-0				—	—	—	√	—	—
			31-16, 15-0				—	—	—	—	√	√
PMCSR11	Port mode control set/reset register 11	32	20-16, 4-0	W, R/W	$092C_H$	$0000\ 0000_H$	—	—	√	—	—	—
			23-16, 7-0				—	—	—	√	—	—
			31-16, 15-0				—	—	—	—	√	√

Table 2.53 Control Registers (P11) (2/2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
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	4-0	R/W	$402C_H$	0000_H	—	—	√	—	—	—
			7-0				—	—	—	√	—	—
			15-0				—	—	—	—	√	√
PBDC11	Port bidirection control register 11	16	4-0	R/W	$412C_H$	0000_H	—	—	√	—	—	—
			7-0				—	—	—	√	—	—
			15-0				—	—	—	—	√	√
PIPC11	Port IP control register 11	16	3, 2	R/W	$422C_H$	0000_H	—	—	√	—	—	—
			7, 6, 3, 2				—	—	—	√	—	—
			10, 9, 7, 6, 3, 2				—	—	—	—	√	—
			10, 9, 7-1				—	—	—	—	—	√
PU11	Pull-up option register 11	16	4-0	R/W	$432C_H$	0000_H	—	—	√	—	—	—
			7-0				—	—	—	√	—	—
			15-0				—	—	—	—	√	√
PD11	Pull-down option register 11	16	4-0	R/W	$442C_H$	0000_H	—	—	√	√	√	√
PODC11	Port open drain control register 11	32	4-0	R/W	$452C_H$	$0000\ 0000_H$	—	—	√	—	—	—
			7-0				—	—	—	√	—	—
			15-0				—	—	—	—	√	√
PDSC11	Port drive strength control register 11	32	4-0	R/W	$462C_H$	$0000\ 0000_H$	—	—	√	—	—	—
			7-0				—	—	—	√	—	—
			15-13, 11-0				—	—	—	—	√	√
PIS11	Port input buffer selection register 11	16	3, 1	R/W	$472C_H$	$FFFF_H$	—	—	√	—	—	—
			6, 5, 3, 1				—	—	—	√	—	—
			15, 13, 12, 9, 6, 5, 3,1				—	—	—	—	√	√
PPROTS11	Port protection status register 11	32	0	R	$4B2C_H$	$0000\ 0000_H$	—	—	√	√	√	√
PPCMD11	Port protection command register 11	32	7-0	W	$4C2C_H$	$0000\ 0000_H$	—	—	√	√	√	√

2.10.9 Port 12 (P12)

2.10.9.1 Alternative Function

Table 2.54 Port 12 (P12)

Port Mode (PMC12_m =0)	Control Mode (PMC12_m =1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	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 undocumented alternative functions are selected.

2.10.9.2 Control Registers

Table 2.55 Control Registers (P12) (1/2)

Register	Function	Register Size	Effective Bit Position	Effective Bit R/W	Offset Address	Value after Reset	Device 48 pins	64 pins	80 pins	100 pins	144 pins	176 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	W, 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	2-0	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	W, R/W	0830_H	$0000\ FFFF_H$	—	—	—	—	√	—
			21-16, 5-0				—	—	—	—	—	√
PMCSR12	Port mode control set/reset register 12	32	18-16, 2-0	W, 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	$0000\ 0000_H$	—	—	—	—	√	√

Table 2.55 Control Registers (P12) (2/2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
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	$4C30_H$	$0000\ 0000_H$	—	—	—	—	√	√

2.10.10 Port 18 (P18)

2.10.10.1 Alternative Function

Table 2.56 Port 18 (P18)

Port Mode (PMC18_m = 0)	Control Mode (PMC18_m = 1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	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	—	—	—	—	—	√

CAUTIONS

1. The behavior and performance are not guaranteed when undocumented alternative functions are selected.
2. Use the special function with initial settings.

2.10.10.2 Control Registers

Table 2.57 Control Registers (P18)

Register	Function	Register Size	Effective Bit Position	Effective Bit R/W	Offset Address	Value after Reset	Device 48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P18	Port register 18	16	3-0	R/W	0048_H	0000_H	—	—	—	—	√	—
			7-0				—	—	—	—	—	√
PSR18	Port set/reset register 18	32	19-16, 3-0	W, R/W	0148_H	$0000\ 0000_H$	—	—	—	—	√	—
			23-16, 7-0				—	—	—	—	—	√
PPR18	Port pin read register 18	16	3-0	R	0248_H	0000_H	—	—	—	—	√	—
			7-0				—	—	—	—	—	√
PM18	Port mode register 18	16	3-0	R/W	0348_H	$FFFF_H$	—	—	—	—	√	—
			7-0				—	—	—	—	—	√
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				—	—	—	—	—	√
PMSR18	Port mode set/reset register 18	32	19-16, 3-0	W, R/W	0848_H	$0000\ FFFF_H$	—	—	—	—	√	—
			23-16, 7-0				—	—	—	—	—	√
PMCSR18	Port mode control set/reset register 18	32	18-16, 2-0	W, R/W	0948_H	$0000\ 0000_H$	—	—	—	—	√	—
			21-16, 5-0				—	—	—	—	—	√
PIBC18	Port input buffer control register 18	16	3-0	R/W	4048_H	0000_H	—	—	—	—	√	—
			7-0				—	—	—	—	—	√
PBDC18	Port bidirection control register 18	16	3-0	R/W	4148_H	0000_H	—	—	—	—	√	—
			7-0				—	—	—	—	—	√
PU18	Pull-up option register 18	16	3-0	R/W	4348_H	0000_H	—	—	—	—	√	—
			7-0				—	—	—	—	—	√
PODC18	Port open drain control register 18	32	3-0	R/W	4548_H	$0000\ 0000_H$	—	—	—	—	√	—
			7-0				—	—	—	—	—	√
PPROTS18	Port protection status register 18	32	0	R	$4B48_H$	$0000\ 0000_H$	—	—	—	—	√	√
PPCMD18	Port protection command register 18	32	7-0	W	$4C48_H$	$0000\ 0000_H$	—	—	—	—	√	√

2.10.11 Port 20 (P20)

2.10.11.1 Alternative Function

Table 2.58 Port 20 (P20)

Port Mode (PMC20_m =0)	Control Mode (PMC20_m = 1)										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
P20_0	RLIN26RX			PWGA64O								—	—	—	—	—	√
P20_1		RLIN26TX		PWGA65O								—	—	—	—	—	√
P20_2	CAN4RX/ INTP4			PWGA66O								—	—	—	—	—	√
P20_3		CAN4TX		PWGA67O								—	—	—	—	—	√
P20_4	RLIN23RX			PWGA59O								—	—	—	—	√	√
P20_5		RLIN23TX		PWGA60O								—	—	—	—	√	√

CAUTION

The behavior and performance are not guaranteed when undocumented alternative functions are selected.

2.10.11.2 Control Registers

Table 2.59 Control Registers (P20) (1/2)

Register	Function	Register Size	Effective Bit Position	Effective Bit R/W	Offset Address	Value after Reset	Device 48 pins	64 pins	80 pins	100 pins	144 pins	176 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	W, 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				—	—	—	—	—	√
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	W, R/W	0850_H	$0000\ FFFF_H$	—	—	—	—	√	—
			21-16, 5-0				—	—	—	—	—	√
PMCSR20	Port mode control set/reset register 20	32	21-20, 5, 4	W, 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				—	—	—	—	—	√

Table 2.59 Control Registers (P20) (2/2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
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	$4C50_H$	$0000\ 0000_H$	—	—	—	—	√	√

2.10.12 Analog Port 0 (AP0)

2.10.12.1 Alternative Function

Table 2.60 Analog Port 0 (AP0)

Port Mode	Control Mode										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	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 with initial settings.

2.10.12.2 Control Registers

Table 2.61 Control Registers (AP0) (1/2)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
AP0	Analog port register 0	16	7-0	R/W	$00C8_H$	0000_H	√	—	—	—	—	—
			9-0				—	√	—	—	—	—
			10-0				—	—	√	—	—	—
			15-0				—	—	—	√	√	√
APSR0	Analog port set/reset register 0	32	23-16, 7-0	W, R/W	$01C8_H$	$0000\ 0000_H$	√	—	—	—	—	—
			25-16, 9-0				—	√	—	—	—	—
			26-16, 10-0				—	—	√	—	—	—
			31-16, 15-0				—	—	—	√	√	√
APPR0	Analog port pin read register 0	16	7-0	R	$02C8_H$	0000_H	√	—	—	—	—	—
			9-0				—	√	—	—	—	—
			10-0				—	—	√	—	—	—
			15-0				—	—	—	√	√	√
APM0	Analog port mode register 0	16	7-0	R/W	$03C8_H$	$FFFF_H$	√	—	—	—	—	—
			9-0				—	√	—	—	—	—
			10-0				—	—	√	—	—	—
			15-0				—	—	—	√	√	√
APNOT0	Analog port NOT register 0	16	7-0	W	$07C8_H$	0000_H	√	—	—	—	—	—
			9-0				—	√	—	—	—	—
			10-0				—	—	√	—	—	—
			15-0				—	—	—	√	√	√
APMSR0	Analog port mode set/reset register 0	32	23-16, 7-0	W, R/W	$08C8_H$	$0000\ FFFF_H$	√	—	—	—	—	—
			25-16, 9-0				—	√	—	—	—	—
			26-16, 10-0				—	—	√	—	—	—
			31-16, 15-0				—	—	—	√	√	√
APIBC0	Analog port input buffer control register 0	16	7-0	R/W	$40C8_H$	0000_H	√	—	—	—	—	—
			9-0				—	√	—	—	—	—
			10-0				—	—	√	—	—	—
			15-0				—	—	—	√	√	√

Table 2.61 Control Registers (AP0) (2/2)

Register	Function	Register Size	Effective Bit Position	Effective Bit R/W	Offset Address	Value after Reset	Device 48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
APBDC0	Analog port bidirection control register 0	16	7-0	R/W	$41C8_H$	0000_H	√	—	—	—	—	—
			9-0				—	√	—	—	—	—
			10-0				—	—	√	—	—	—
			15-0				—	—	—	√	√	√

2.10.13 Analog Port 1 (AP1)

2.10.13.1 Alternative Function

Table 2.62 Analog Port 1 (AP1)

Port Mode	Control Mode										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	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 with initial settings.

2.10.13.2 Control Registers

Table 2.63 Control Registers (AP1)

Register	Function	Register Size	Effective Bit		Offset Address	Value after Reset	Device					
			Position	R/W			48 pins	64 pins	80 pins	100 pins	144 pins	176 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	W, 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	W, 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				—	—	—	—	—	√

2.10.14 Input Port 0 (IP0)

2.10.14.1 Alternative Function

Table 2.64 Input Port 0 (IP0)

Port Mode	Control Mode										Special Function	Device					
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		5th Alternative			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
	Input	Output	Input	Output	Input	Output	Input	Output	Input	Output							
IP0_0											XT2	—	—	—	—	√	√

2.10.14.2 Control Registers

Table 2.65 Control Registers (IP0)

Register	Function	Register Size	Effective Bit: Position	Effective Bit: R/W	Offset Address	Value after Reset	Device: 48 pins	64 pins	80 pins	100 pins	144 pins	176 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	—	—	—	—	√	√

CAUTION

When the IP0_0/XT2 pin is used as an input port, set the IPIBC0.0 bit to 1 and stop the SOSC operation. For details on the settings for SOSC operations, see **Section 10.4.2.6, SOSCE — SubOSC Enable Register**. When the IP0_0/XT2 pin is used for the SubOSC (SOSC) not as an input port, set the PIBC0.0 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: $\overline{\text{RESETOUT}}$

After reset release, P0_0 outputs a $\overline{\text{RESETOUT}}$ signal, which is low level during and after reset release. P0_0 functional register is configured as follows after reset release:

- PM0.PM0_0 = 0: output port
- PODC0.PODC0_0 = 1: open-drain output

Since P0.P0_0 = 0 after reset, low level is output.

Any change of the P0_0 configuration above terminates the $\overline{\text{RESETOUT}}$ output.

2.11.1.2 JP0_0 to JP0_5: Debug Interface

If the OPJTAG[1:0] setting is Nexus I/F, after reset release the pins of the JP0 port group can be used as a debug interface.

- JP0_0: DCUTDI input
- JP0_1: DCUTDO output
- JP0_2: DCUTCK input
- JP0_3: DCUTMS
- JP0_4: $\overline{\text{DCUTRST}}$
- JP0_5: $\overline{\text{DCUTRDY}}$

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

NOTE

For the OPJTAG[1:0] settings, see **Section 35.11.2, OPBT0 — Option Byte 0 Register**.

2.11.1.3 JP0_0, JP0_1, JP0_2: Flash Programmer

These ports are used for connecting a flash programmer. See *Hardware User's Manual of flash programmer* for the detail.

2.11.1.4 Mode Pins

FLMD0 pin in combination with P10_8: FLMD1 pin can set serial flash programming mode.

2.11.2 A/D Input Alternative I/O

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 2.66 A/D Input Alternative Pins (1/2)

Port	A/D Input	Device					
		48 pins	64 pins	80 pins	100 pins	144 pins	176 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	—	—	√	√	√	√
P9_5	ADCA0I12S	—	—	√	√	√	√
P9_6	ADCA0I13S	—	—	√	√	√	√
P18_0	ADCA1I0S	—	—	—	—	√	√
P18_1	ADCA1I1S	—	—	—	—	√	√
P18_2	ADCA1I2S	—	—	—	—	√	√
P18_3	ADCA1I3S	—	—	—	—	√	√
P18_4	ADCA1I4S	—	—	—	—	—	√
P18_5	ADCA1I5S	—	—	—	—	—	√
P18_6	ADCA1I6S	—	—	—	—	—	√
P18_7	ADCA1I7S	—	—	—	—	—	√
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	—	√	√	√	√	√

Table 2.66 A/D Input Alternative Pins (2/2)

Port	A/D Input	Device					
		48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
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 modules take over the input and output control of the used ports automatically.

These ports must be set in alternative mode by setting PMCn.PMCn_m, PFCn.PFCn_m and PFCEn.PFCEn_m accordingly, and I/O control must be handed over to the module by setting PIPCn.PIPCn_m = 1.

The setting of PMn.PMn_m has no more effect for these ports.

The following table lists all alternative modes, where PIPCn.PIPCn_m must be set to 1.

Table 2.67 Alternative Modes that Require Setting PIPCn.PIPCn_m = 1 (1/2)

Function	Pin Name	Port Name	Device						Reference Section
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	
MEMC	MEMC0AD0	P10_6	—	—	—	—	—	√	**Section 13**
	MEMC0AD1	P10_7	—	—	—	—	—	√	
	MEMC0AD2	P10_8	—	—	—	—	—	√	
	MEMC0AD3	P10_9	—	—	—	—	—	√	
	MEMC0AD4	P10_10	—	—	—	—	—	√	
	MEMC0AD5	P10_11	—	—	—	—	—	√	
	MEMC0AD6	P10_12	—	—	—	—	—	√	
	MEMC0AD7	P10_13	—	—	—	—	—	√	
	MEMC0AD8	P10_14	—	—	—	—	—	√	
	MEMC0AD9	P11_1	—	—	—	—	—	√	
	MEMC0AD10	P11_2	—	—	—	—	—	√	
	MEMC0AD11	P11_3	—	—	—	—	—	√	
	MEMC0AD12	P11_4	—	—	—	—	—	√	
	MEMC0AD13	P11_5	—	—	—	—	—	√	
	MEMC0AD14	P11_6	—	—	—	—	—	√	
	MEMC0AD15	P11_7	—	—	—	—	—	√	
TAPA	TAPA0UP	P10_0	√	√	√	√	√	√	**Section 27**
	TAPA0UN	P10_1	√	√	√	√	√	√	
	TAPA0VP	P10_2	√	√	√	√	√	√	
	TAPA0VN	P10_3	√	√	√	√	√	√	
	TAPA0WP	P10_4	√	√	√	√	√	√	
	TAPA0WN	P10_5	√	√	√	√	√	√	
CSIG	CSIG0SO	P0_13	—	—	—	√	√	√	**Section 14**
		P10_6	√	√	√	√	√	√	
	CSIG0SC	P0_14	—	—	—	√	√	√	
		P10_7	√	√	√	√	√	√	
	CSIG1SO	P11_9	—	—	—	—	√	√	
	CSIG1SC	P11_10	—	—	—	—	√	√	

Table 2.67 Alternative Modes that Require Setting PIPCn.PIPCn_m = 1 (2/2)

Function	Pin Name	Port Name	Device						Reference Section
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	
CSIH	CSIH0SO	P0_3	√	√	√	√	√	√	**Section 15**
	CSIH0SC	P0_2	√	√	√	√	√	√	
	CSIH1SO	P0_5	—	—	√	√	√	√	
		P10_2	—	—	√	√	√	√	
	CSIH1SC	P0_6	—	—	√	√	√	√	
		P10_1	—	—	√	√	√	√	
	CSIH2SO	P11_2	—	—	√	√	√	√	
	CSIH2SC	P11_3	—	—	√	√	√	√	
	CSIH3SO	P11_6	—	—	—	√	√	√	
	CSIH3SC	P11_7	—	—	—	√	√	√	

2.11.3.2 Input Buffer Control (PIS)

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.

Table 2.68 Input Buffer Characteristic Selection (1/2)

Port Name	Input Buffer Selection		Device					
	Type 1 (PISn_m = 0)	Type 2 (PISn_m = 1)	48 pins	64 pins	80 pins	100 pins	144 pins	176 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_13	SHMT1	SHMT4	—	√	√	√	√	√
P10_15	SHMT1	SHMT4	—	—	√	√	√	√
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	—	—	—	—	√	√

Table 2.68 Input Buffer Characteristic Selection (2/2)

Port Name	Input Buffer Selection		Device					
	Type 1 (PISn_m = 0)	Type 2 (PISn_m = 1)	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P11_15	SHMT1	SHMT4	—	—	—	—	√	√
P12_1	SHMT1	SHMT4	—	—	—	—	√	√
P12_3	SHMT1	SHMT4	—	—	—	—	—	√
P20_0	SHMT1	SHMT4	—	—	—	—	—	√
P20_2	SHMT1	SHMT4	—	—	—	—	—	√
P20_4	SHMT1	SHMT4	—	—	—	—	√	√

NOTES

1. For the SHMT1 and SHMT4 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 the ports other than the listed below.

Table 2.69 Output Buffer Characteristic Selection (1/2)

Port Name	Output Driver Strength Selection		Device					
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P0_2	10 MHz	40 MHz*1	√	√	√	√	√	√
P0_3	10 MHz	40 MHz*1	√	√	√	√	√	√
P0_5	10 MHz	40 MHz	—	√	√	√	√	√
P0_6	10 MHz	40 MHz	—	√	√	√	√	√
P0_7	10 MHz	40 MHz	—	—	√	√	√	√
P0_13	10 MHz	40 MHz	—	—	—	√	√	√
P0_14	10 MHz	40 MHz	—	—	—	√	√	√
P10_0	10 MHz	40 MHz	√	√	√	√	√	√
P10_1	10 MHz	40 MHz	√	√	√	√	√	√
P10_2	10 MHz	40 MHz	√	√	√	√	√	√
P10_3	10 MHz	40 MHz	√	√	√	√	√	√
P10_4	10 MHz	40 MHz	√	√	√	√	√	√
P10_5	10 MHz	40 MHz	√	√	√	√	√	√
P10_6	10 MHz	40 MHz	√	√	√	√	√	√
P10_7	10 MHz	40 MHz	√	√	√	√	√	√
P10_8	10 MHz	40 MHz	√	√	√	√	√	√
P10_9	10 MHz	40 MHz	√	√	√	√	√	√
P10_10	10 MHz	40 MHz	√	√	√	√	√	√
P10_11	10 MHz	40 MHz	—	√	√	√	√	√
P10_12	10 MHz	40 MHz	—	√	√	√	√	√
P10_13	10 MHz	40 MHz	—	√	√	√	√	√
P10_14	10 MHz	40 MHz	—	√	√	√	√	√
P10_15	10 MHz	40 MHz	—	—	√	√	√	√
P11_0	10 MHz	40 MHz	—	—	√	√	√	√
P11_1	10 MHz	40 MHz	—	—	√	√	√	√
P11_2	10 MHz	40 MHz	—	—	√	√	√	√
P11_3	10 MHz	40 MHz	—	—	√	√	√	√
P11_4	10 MHz	40 MHz	—	—	√	√	√	√
P11_5	10 MHz	40 MHz	—	—	—	√	√	√
P11_6	10 MHz	40 MHz	—	—	—	√	√	√
P11_7	10 MHz	40 MHz	—	—	—	√	√	√
P11_8	10 MHz	40 MHz	—	—	—	—	√	√
P11_9	10 MHz	40 MHz	—	—	—	—	√	√
P11_10	10 MHz	40 MHz	—	—	—	—	√	√
P11_11	10 MHz	40 MHz	—	—	—	—	√	√
P11_13	10 MHz	40 MHz	—	—	—	—	√	√
P11_14	10 MHz	40 MHz	—	—	—	—	√	√
P11_15	10 MHz	40 MHz	—	—	—	—	√	√

Table 2.69 Output Buffer Characteristic Selection (2/2)

Port Name	Output Driver Strength Selection		Device					
	Slow Mode (PDSCn_m = 0)	Fast Mode (PDSCn_m = 1)	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P12_0	10 MHz	40 MHz	—	—	—	—	√	√
P12_1	10 MHz	40 MHz	—	—	—	—	√	√
P12_2	10 MHz	40 MHz	—	—	—	—	√	√

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

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/F1L 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 signals 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 functions 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 2.70 Input Pins that Incorporate Analog Filter Type A

	FCLA0CTL Register Configuration		Device					
Input Signal	Register	Address	48 pins	64 pins	80 pins	100 pins	144 pins	176 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 and edge/level detection function. Edge/level detection is controlled by the registers for individual peripheral functions.

Table 2.71 Input Pins that Incorporate Analog Filter Type B

Input Signal	Edge/Level Detection	Device					
		48 pins	64 pins	80 pins	100 pins	144 pins	176 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 24.3, Registers**.

Note 2. For details on edge detection for TAPA, see **Section 27.3, Registers**.

2.12.1.3 Input Pins that Incorporate Analog Filter Type C

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

Table 2.72 Input Pins that Incorporate Analog Filter Type C

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

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 2.73 Input Pins that Incorporate Digital Filter Type D

Input Pin	Device						Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Filter Control Register	
	48 pin	64 pin	80 pin	100 pin	144 pin	176 pin	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 and edge detection function. 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 2.74 Input Pins that Incorporate Digital Filter Type E (1/3)

Input Pin	Device						Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Edge Detection
	48 pin	64 pin	80 pin	100 pin	144 pin	176 pin	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		

Table 2.74 Input Pins that Incorporate Digital Filter Type E (2/3)

Input Pin	Device						Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Edge Detection
	48 pin	64 pin	80 pin	100 pin	144 pin	176 pin	Control Register	Address	Control Register	Control Bit	Address	Register Name
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 2.74 Input Pins that Incorporate Digital Filter Type E (3/3)

Input Pin	Device						Digital Noise Elimination Control Register		Digital Noise Elimination Enable Register			Edge Detection
	48 pin	64 pin	80 pin	100 pin	144 pin	176 pin	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		
ENCA0AIN	√	√	√	√	√	√				DNFAENCA0IENL2		
ENCA0BIN	√	√	√	√	√	√				DNFAENCA0IENL3		
ENCA0ZIN	√	√	√	√	√	√				DNFAENCA0IENL4		

Note 1. For the setting for TAUD edge detection, see **Section 23.3.3.4, TAUDnCMURm — TAUDn Channel Mode User Register**.

Note 2. For the setting for TAUB edge detection, see **Section 22.3.3.4, TAUBnCMURm — TAUBn Channel Mode User Register**

Note 3. For the setting for ENCA edge detection, see **Section 26.3.3, 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 2.75 Clock Supply for Port Filters

Peripheral Function	Port Domain*[1]	Filter Type	Filter Clock	Setting Register	
				Source Clock Selection	Clock Selection
AD0	Always-On area	Digital filter type D	DNFATCKI	CKSC_AADCAS_CTL	CKSC_AADCAD_CTL
AD1	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 supply domain

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 functions 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 entailing 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 signal, asynchronous Hi-Z control input signal, and key return input signal.
- 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 entailing edge detection or level 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 signal and encoder input signal.

2.13.2 Analog Filters

2.13.2.1 Analog Filter Characteristic

The characteristics of an input signal pin, that is equipped with an analog filter, are specified in the data sheet document.

2.13.2.2 Analog Filters Control Registers

For each input signal, that is equipped with an analog filter, a dedicated control register FCLA0CTLm_<name> and a control register in each peripheral macro are provided.

The assignment of the input signals to the control registers and their addresses are given in **Table 2.70, Input Pins that Incorporate Analog Filter Type A**, in **Section 2.12.1, Port Filter Assignment**.

2.13.2.3 Analog Filters in Standby Mode

Analogue filters for the function of waking-up from the DEEPSTOP mode are located in the always-on area (AWO). The analogue filter of the AWO is always operating.

The analogue filter (level detection) in standby mode and its wake-up capability depend on the filter types. See the description of the analogue filter types below.

(1) Analog Filter Type A

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

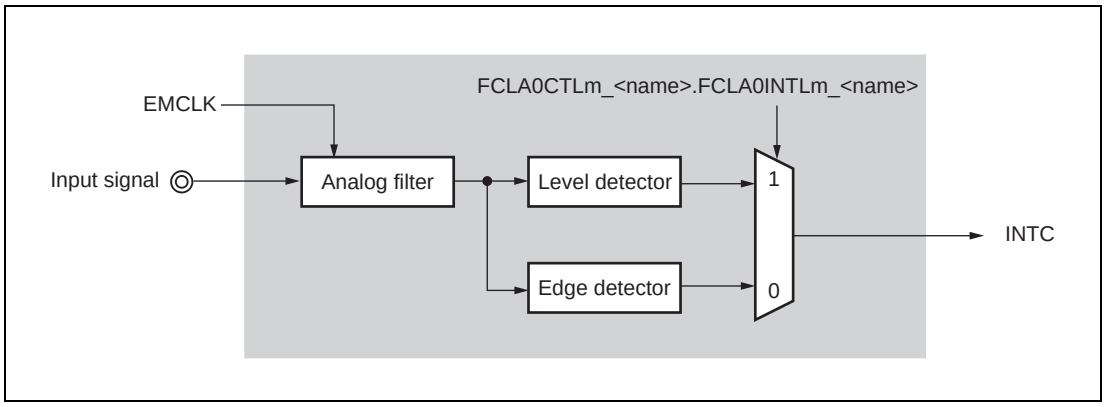


Figure 2.12 Block Diagram of Analog Filter Type A

After passing an external signal through the filter to eliminate noise and spikes, the filter generates an output signal 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>.FCLAnINTLm_<name>.

- FCLA0INTLm_<name> = 0: Edge detection

  Whether a rising or falling edge is detected can be specified by setting the FCLA0CTLm_n.FCLA0INTRm_<name> and FCLA0CTLm_<name>.FCLA0INTFm_<name> bits.

- FCLA0INTLm_<name> = 1: Level detection

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

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

Table 2.76 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

In case the clock PCLK is stopped in standby mode, the analog filter can only operate with the edge detection. Thus set FCLA0INTLm_<name> = 0 and select the required edge detection if the input signal shall 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.

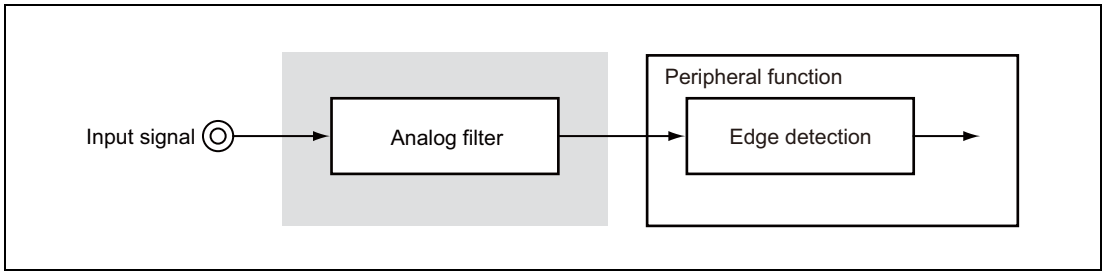


Figure 2.13 Block Diagram of Analog Filter Type B

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.

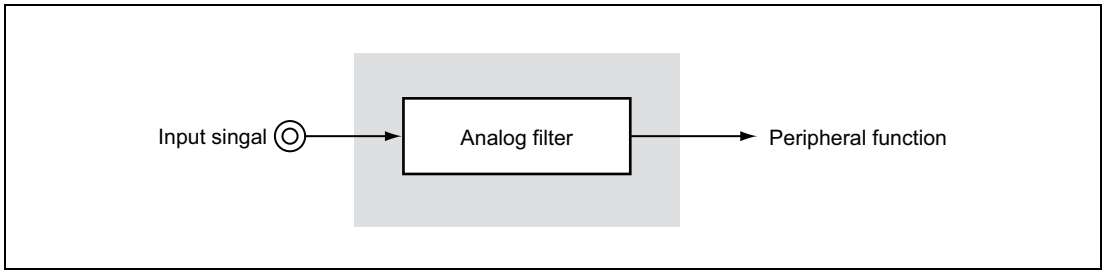


Figure 2.14 Block Diagram of Analog Filter Type C

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

Analog filter type C in Standby mode

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

2.13.3 Digital Filters

2.13.3.1 Digital Filter Characteristic

The digital filters allow to adjust the filter characteristics 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] allows to select the sampling frequency to $f_s = f_{DNFATCKI} / 2^{DNFA<name>PRS[2:0]}$ where $f_{DNFATCKI}$ is the frequency of the DNFATCKI clock.
- DNFA<name>CTL.DNFA<name>NFSTS[1:0] determines the number s of same level samples (2 to 5):

The number of samples, s, is defined below.

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

External signal pulses shorter than

$s \times 1/f_s$

are always suppressed.

External signal pulses longer than

$(s + 1) \times 1/f_s$

are always judged to be valid and passed on to the filter output.

Consequently, external signal pulses in the range

$s \times 1/f_s$ to $(s + 1) \times 1/f_s$

may be suppressed or judged to be valid.

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

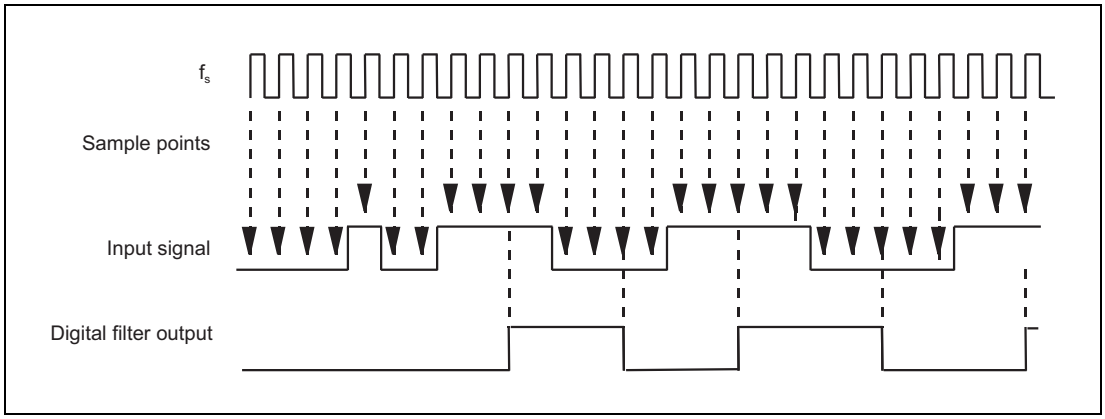


Figure 2.15 Digital Filter Function

2.13.3.2 Digital Filter Groups

The input signals 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 filters of the entire 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 set to an input for 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
   $s \times 1/f_S + 2 \times 1/f_{DNFATCKI}$
2. **When a digital filter is used with an interrupt acting as an event output 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.**
   $s \times 1/f_S + 3 \times 1/f_{DNFATCKI}$

2.13.3.3 Digital Filters in Standby Mode

Digital filters for the function of waking-up from the DEEPSTOP mode are located in the always-on area (AWO).

Digital noise elimination requires the clock supply DNFATCKI to operate. Therefore, if DNFATCKI is stopped in standby mode, digitally filtered signals cannot serve as standby mode wake-up event.

If DNFATCKI operates in standby mode, a wake-up event can be generated by an external signal.

2.13.3.4 Digital Filters Control Registers

For each group consisting of up to 16 digital filters, the common 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 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 in the range from 0 to 1.

The assignment of the input signals to the control registers and their addresses are given in **Table 2.77, Output Options for Digital Filter Type D** in **Section 2.12.1, Port Filter Assignment**.

CAUTION

Do not change any control register settings, while the concerned 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.

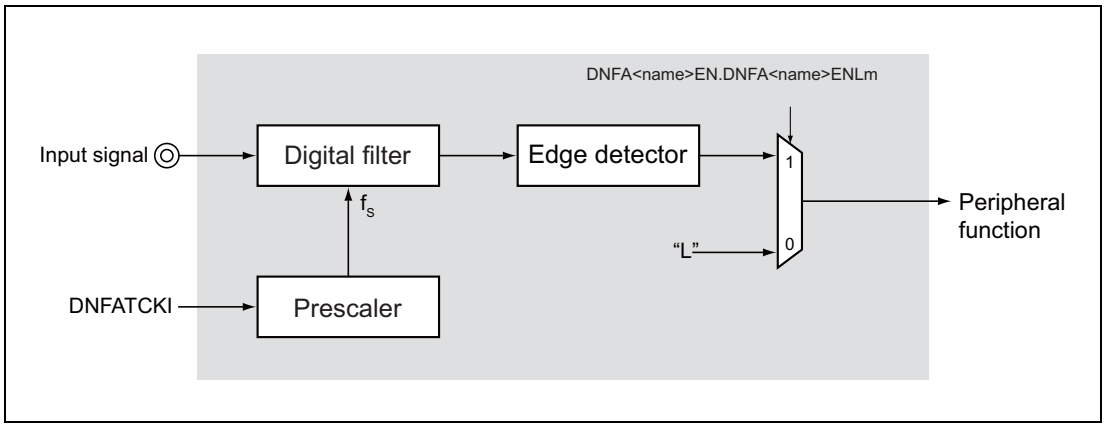


Figure 2.16 Block Diagram of Digital Filter Type D

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

Table 2.77 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.

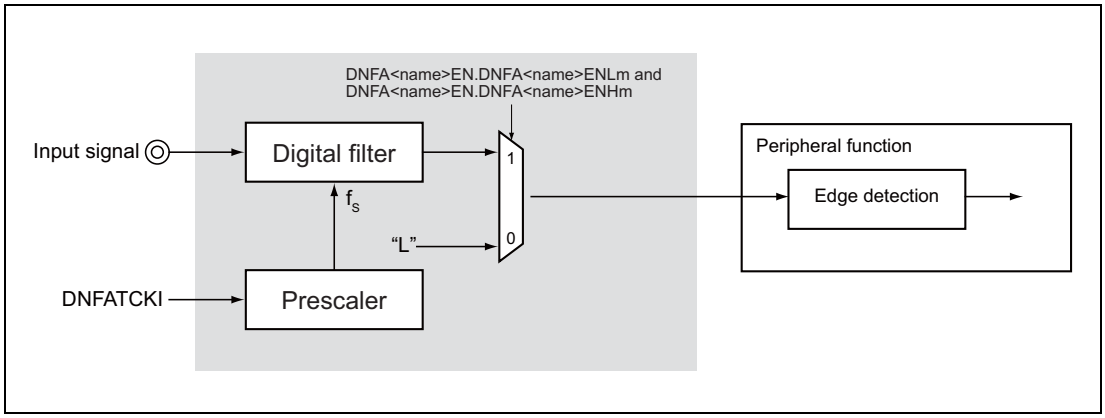


Figure 2.17 Block Diagram of Digital Filter Type E

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

Table 2.78 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 2.79 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

This register controls the analog and digital filter operation.

Access: This register can be read/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 2.80 FCLA0CTLm_<name> Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When writing to these bits, 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 write 0 to bit 2.

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/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 2.73, Input Pins that Incorporate Digital Filter Type D** and **Table 2.74, 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 2.81 DNFA<name>CTL Register Contents

Bit Position	Bit Name	Function
7	Reserved	When writing to this bit, 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.
4, 3	Reserved	When writing to these bits, write the value after reset.
2 to 0	DNFA<name> PRS[2:0]	Digital filter sampling clock selection

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

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/written in 16-bit units.
The higher- and lower-order bytes (DNFA<name>NFENH[7:0] and DNFA<name>NFENL[7:0]) are accessible in 8- or 1-bit units respectively by setting DNFA<name>ENH.DNFA<name>NFEH[7:0] and DNFA<name>ENL.DNFA<name>NFEN[7:0].

Address: For the correspondence between the DNFA<name>EN register and input signals, and the addresses of individual registers, see **Table 2.74, Input Pins that Incorporate Digital Filter Type E** and **Table 2.73, Input Pins that Incorporate Digital Filter Type D** in **Section 2.12.1, Port Filter Assignment**.

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 2.82 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 Control 0: Digital noise elimination disabled 1: Digital noise elimination enabled

Section 3 CPU System

This section describes the CPU-related items included in the RH850 Family Software User's Manual that require specific attention when using the RH850/F1L, as well as CPU-related items that apply specifically to the RH850/F1L and are therefore not included in the *RH850 Family Software User's Manual*.

3.1 Overview

3.1.1 Architectural Features

The CPU incorporated in the RH850/F1L has a Harvard architecture and uses a RISC instruction set.

This CPU has a 5-stage pipeline and can execute almost all instructions in a single clock cycle.

The CPU also uses a 32-bit hardware multiplier, enabling high-speed multiplication processing. A range of instructions such as saturating and bit manipulation instructions are also supported.

3.1.1.1 CPU

- Core: G3K CPU (V850E3v5-S architecture class) × 1
  Instruction execution time: 12.5 ns (when operating at 80 MHz)
  10.4 ns (when operating at 96 MHz)
- 5-stage pipeline
- Only native mode is supported as the CPU operating mode

3.1.1.2 Coprocessor

- There is no coprocessor (floating-point coprocessor or fixed-point SIMD coprocessor).

3.1.1.3 Exceptions and Interrupts

- Interrupts of the V850E1 CPU and V850E2 CPU are handled as types of exceptions.
  - Interrupt offset addresses are generated by the CPU (not output from the interrupt controller).
  - FE-level interrupts (FENMI and FEINT) and EI-level interrupts (INT) are input from the interrupt controller.
- The direct vector and table reference methods are both supported.
- Up to 8 priority levels can be assigned to interrupts.

3.1.1.4 Memory Management

- Memory space
  - Program area: Linear 128 MB area
  - Data area: Linear 128 MB area
- Processor protection functions
  - Memory Protection Unit (MPU)
    The MPU protects the CPU's memory space against execution or data manipulation that is not allowed for user programs (up to 4 memory protection areas).
  - There is no memory protection setting check function.
- There is no memory management unit (MMU).
- Cache
  There is no cache operation function (cache instructions are not supported).

3.1.1.5 System Register Protection (SRP)

The SRP function can prevent damage to system registers by non-trusted programs.

3.1.1.6 Functions not Supported

The following functions are not supported by the RH850/F1L:

- Hardware multithreading function
- CPU virtualization support function
- Multiprocessing function

3.1.2 Power Supply and Clock

The CPU subsystem is located on the Isolated-Area (ISO) and is supplied with the CPUCLK clock.

3.1.3 RH850/F1L CPU Subsystem

The following figure shows a block diagram of the RH850/F1L CPU subsystem.

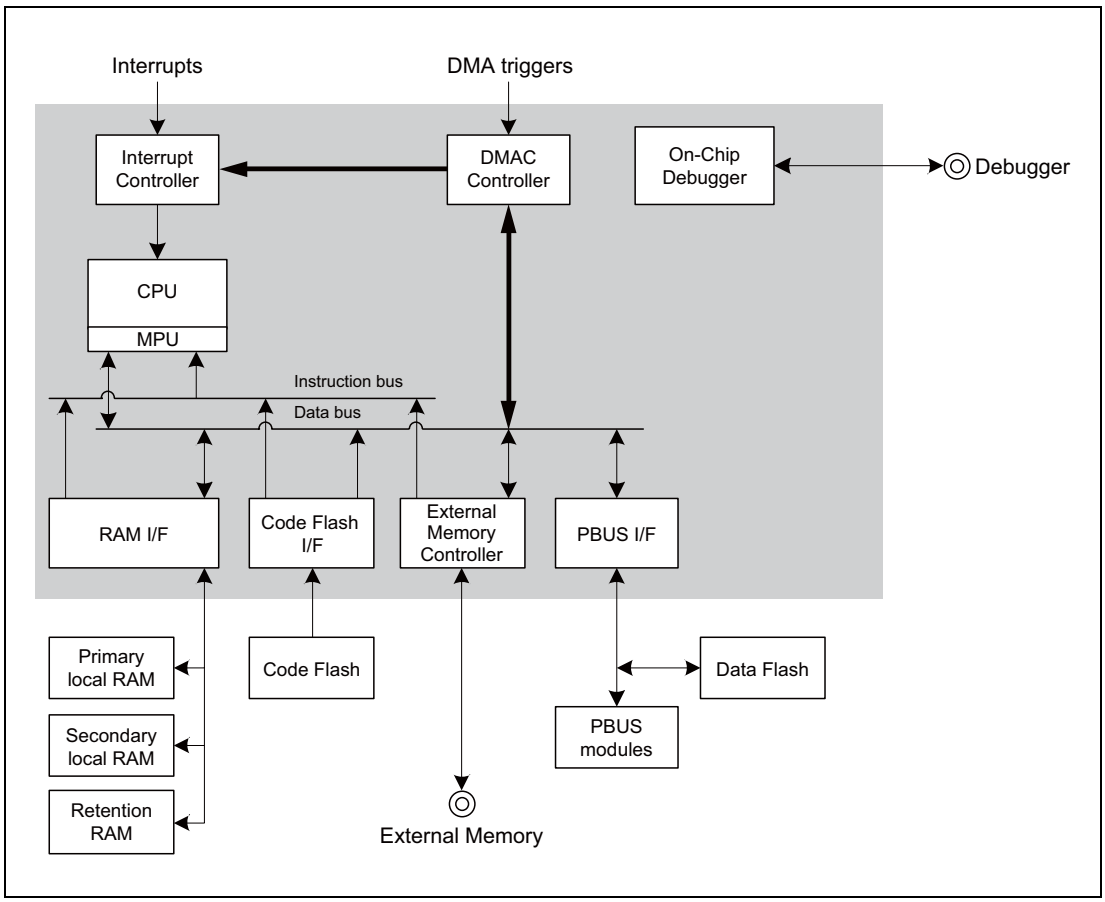


Figure 3.1 RH850/F1L CPU Subsystem

All buses used for data transfer among many peripheral devices are controlled by two master devices.

- CPU
- DMA controller (DMAC)

Table 3.1 RH850/F1L CPU Subsystem Data / Instruction Buses

Master	Bus	Code Flash	Primary Local RAM/ Secondary Local RAM/ Retention RAM	Data Flash	PBUS I/F
CPU	Instruction (32 bits)	R	R	—	—
	Data (32 bits)	R	R/W	R/W*1	R/W
DMA controller	DMA data bus (32 bits)	R	R/W	R	R/W

Note 1. Cannot be written by the store instruction.

3.2 Processor Model

This section describes the items included in CHAPTER 2 of the *RH850 Family User's Manual: Software* that require specific attention when using the RH850/F1L.

3.2.1 CPU Operating Mode

The RH850/F1L only supports native mode as the CPU operating mode. Resources are managed by a 2-layer control system consisting of a supervisor (SV) layer and a user layer.

All instructions can be executed in supervisor mode. The instructions that can be executed in user mode are limited, and an exception occurs if an unauthorized instruction is executed.

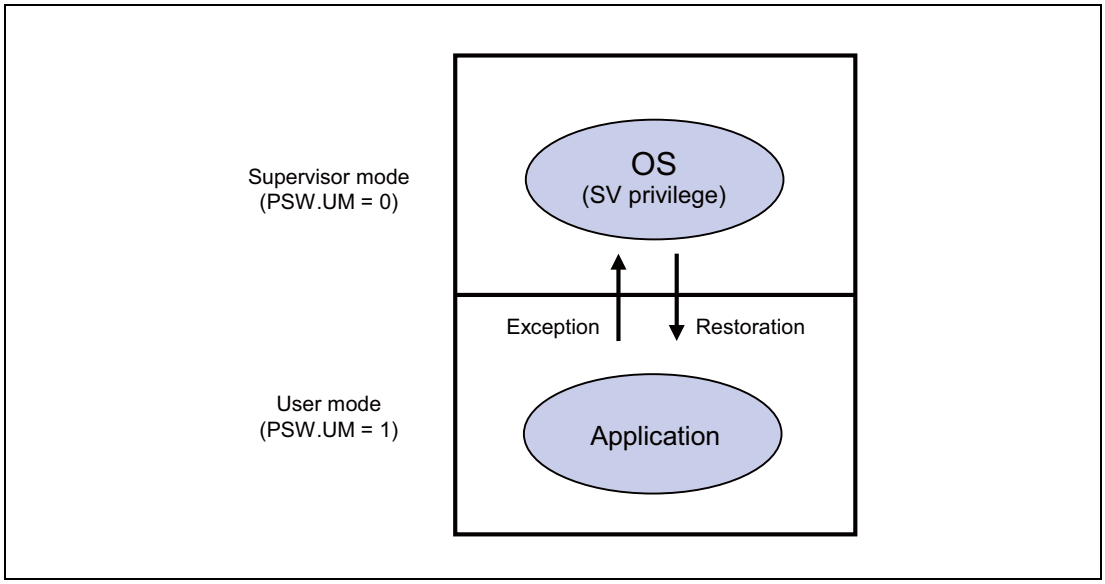


Figure 3.2 Mode Transitions in Native Mode

3.2.2 Hardware Thread

The RH850/F1L only supports single-thread processing. If the HALT instruction is executed, therefore, the CPU stops operating. The HTCT.HLT bit is not set even if the HALT instruction is executed.

Also, when the SNOOZE instruction is executed, the thread that was stopped by the SNOOZE instruction starts operating again after 32 clock cycles have elapsed.

3.2.3 CPU Data Address and Physical Program Address Space

This section describes the CPU's address space, i.e. size and addresses of CPU address space and the physical address space.
The address range of data space and program space together with their wraparound properties are presented.

The CPU supports the following address space.

- 128 MB CPU data address space

  128 MB of a 4 GB (max.) address space can be accessed by using the 32-bit general registers.

- 128 MB physical program address space

  The CPU provides 128 MB of physical address space to access instruction codes in the program memory. This means that up to 128 MB external or internal memory is accessible.

3.2.3.1 Program Space and Data Space

The figure below shows the assignment of data space and program space in the CPU address space.

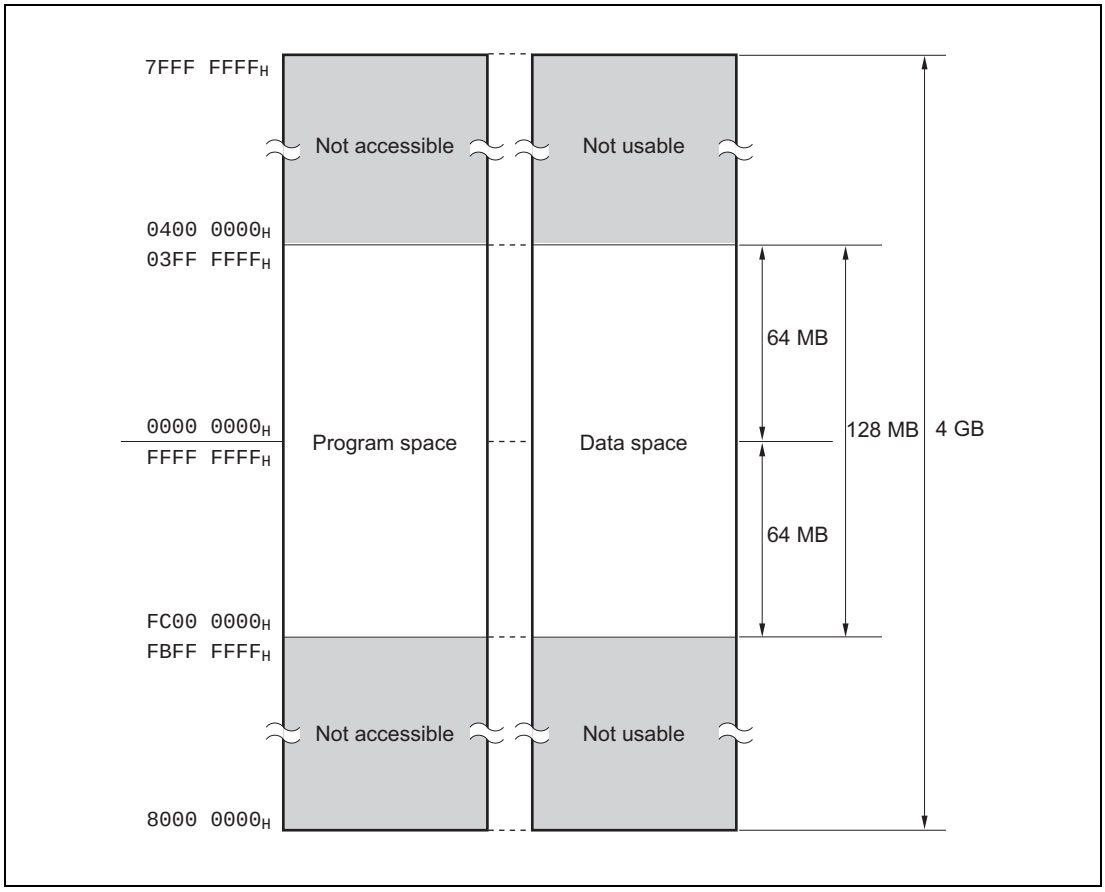


Figure 3.3 CPU Address Space

For instruction address addressing, a value whose 26th bit is sign-extended is automatically specified for the higher-order 5 bits of the register holding the instruction address. Therefore, the addressable range is 0000 0000_H to 03FF $FFFE_H$ and FC00 0000_H to FFFF $FFFE_H$ (and the least significant bit is always 0). Be sure to place the instructions and the tables referenced using the SWITCH, CALLT, and SYSCALL instructions in the instruction addressable address range.

3.2.3.2 Wrap-Around of Data Space

If an operand address calculation exceeds 32 bits, only the lower 32 bits of the result are considered. Therefore, the addresses 0000 0000_H and FFFF $FFFF_H$ are contiguous addresses.
This results in a wrap-around of the data space:

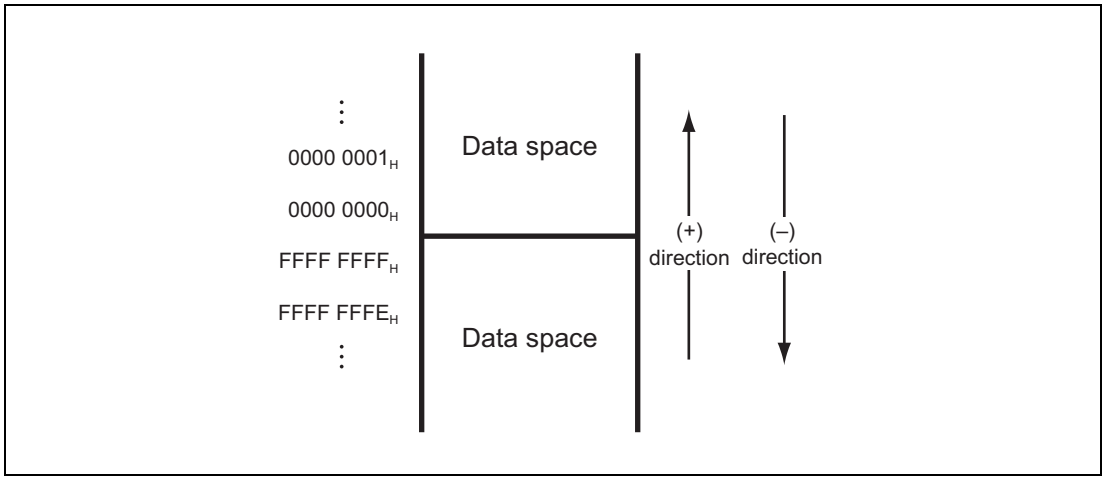


Figure 3.4 Wrap-Around of Data Space

3.2.3.3 Wrap-Around of Program Space

If an instruction address calculation exceeds 26 bits, only the lower 26 bits of the result are considered (a value resulting from a sign-extension of bit 26 is set to the higher-order 5 bits). Therefore, the addresses 03FF $FFFF_H$ and FC00 0000_H, and FFFF $FFFF_H$ and 0000 0000_H are contiguous addresses. This results in a wrap-around of the program space.

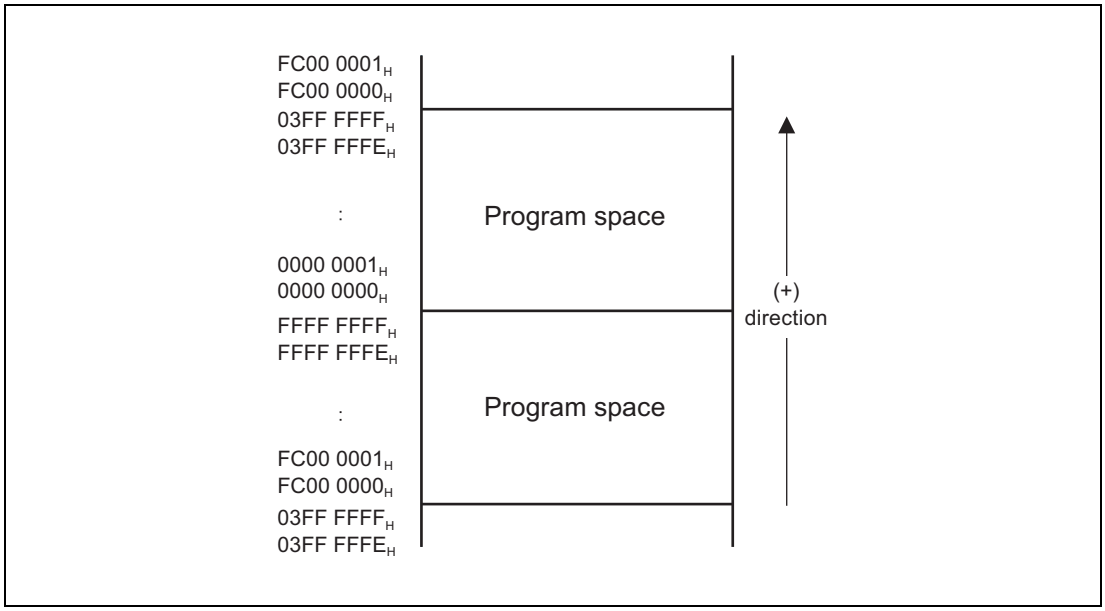
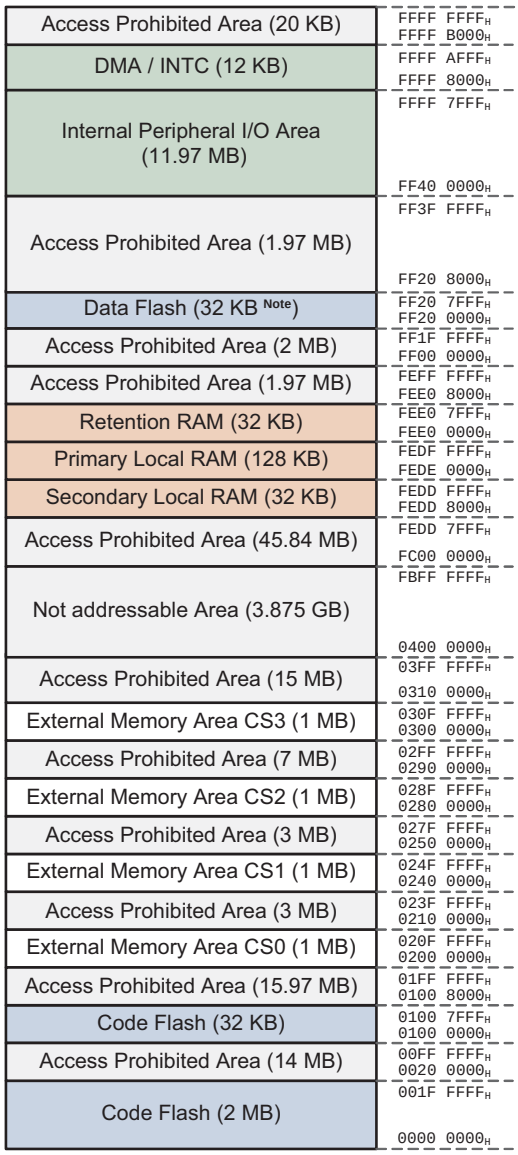


Figure 3.5 Wrap-Around of Program Space

3.2.4 RH850/F1L Memory Map

This section describes the CPU memory map, DMA address map, and specific memory space.



Note: The data flash memory for the products in the PREMIUM range is 64 Kbytes (FF20 0000_H to FF20 $FFFF_H$).

CAUTION: Do not access the address prohibited area, nor the address to which no register has been allocated.

Figure 3.6 RH850/F1L Memory Map (Example of the product with 176 pins and code flash (2 MB))

The size, and thus the address range, on the code flash and RAM differs depending on the product. The external memory area is only included in F1L devices with 176 pins.

3.2.4.1 Code Flash Area

The following table lists the code flash sizes and address ranges.

Table 3.2 Code Flash Memory Area

Code Flash Size	Address Range	Product Name					
		48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
256 KB	0000 0000$_H$ - 0003 FFFF$_H$	R7F7010082AFP R7F7010083AFP R7F7010084AFP	R7F7010112AFP R7F7010113AFP R7F7010114AFP	R7F7010162AFP R7F7010163AFP R7F7010164AFP	R7F7010212AFP R7F7010213AFP R7F7010214AFP	—	—
384 KB	0000 0000$_H$ - 0005 FFFF$_H$	R7F7010092AFP R7F7010093AFP R7F7010094AFP	R7F7010122AFP R7F7010123AFP R7F7010124AFP	R7F7010172AFP R7F7010173AFP R7F7010174AFP	R7F7010222AFP R7F7010223AFP R7F7010224AFP	R7F7010262AFP R7F7010263AFP R7F7010264AFP	—
512 KB	0000 0000$_H$ - 0007 FFFF$_H$	R7F7010102AFP R7F7010103AFP R7F7010104AFP	R7F7010132AFP R7F7010133AFP R7F7010134AFP	R7F7010182AFP R7F7010183AFP R7F7010184AFP	R7F7010232AFP R7F7010233AFP R7F7010234AFP R7F7010022AFP R7F7010023AFP R7F7010024AFP	R7F7010272AFP R7F7010273AFP R7F7010274AFP	—
768 KB	0000 0000$_H$ - 000B FFFF$_H$	—	R7F7010142AFP R7F7010143AFP R7F7010144AFP	R7F7010192AFP R7F7010193AFP R7F7010194AFP	R7F7010242AFP R7F7010243AFP R7F7010244AFP	R7F7010282AFP R7F7010283AFP R7F7010284AFP	R7F7010322AFP R7F7010323AFP R7F7010324AFP
			R7F7010402AFP R7F7010403AFP R7F7010404AFP	R7F7010422AFP R7F7010423AFP R7F7010424AFP	R7F7010442AFP R7F7010443AFP R7F7010444AFP	R7F7010462AFP R7F7010463AFP R7F7010464AFP	R7F7010502AFP R7F7010503AFP R7F7010504AFP
1 MB	0000 0000$_H$ - 000F FFFF$_H$	—	R7F7010152AFP R7F7010153AFP R7F7010154AFP	R7F7010202AFP R7F7010203AFP R7F7010204AFP	R7F7010252AFP R7F7010253AFP R7F7010254AFP R7F7010032AFP R7F7010033AFP R7F7010034AFP	R7F7010292AFP R7F7010293AFP R7F7010294AFP	R7F7010332AFP R7F7010333AFP R7F7010334AFP
			R7F7010412AFP R7F7010413AFP R7F7010414AFP	R7F7010432AFP R7F7010433AFP R7F7010434AFP	R7F7010452AFP R7F7010453AFP R7F7010454AFP	R7F7010472AFP R7F7010473AFP R7F7010474AFP	R7F7010512AFP R7F7010513AFP R7F7010514AFP
1.5 MB	0000 0000$_H$ - 0017 FFFF$_H$	—	—	—	—	R7F7010302AFP R7F7010303AFP R7F7010304AFP	R7F7010342AFP R7F7010343AFP R7F7010344AFP
						R7F7010482AFP R7F7010483AFP R7F7010484AFP	R7F7010522AFP R7F7010523AFP R7F7010524AFP
						R7F7010542AFP R7F7010543AFP R7F7010544AFP	R7F7010562AFP R7F7010563AFP R7F7010564AFP
2 MB	0000 0000$_H$ - 001F FFFF$_H$	—	—	—	—	R7F7010492AFP R7F7010493AFP R7F7010494AFP	R7F7010532AFP R7F7010533AFP R7F7010534AFP
						R7F7010552AFP R7F7010553AFP R7F7010554AFP	R7F7010572AFP R7F7010573AFP R7F7010574AFP

3.2.4.2 Data Flash Area

The following table lists the data flash sizes and address ranges.

Table 3.3 Data Flash Area

Code Flash Size	Data Flash Size	Address Range	Product Name					
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
256 KB	32 KB	FF20 0000$_H$ - FF20 7FFF$_H$	R7F7010082AFP R7F7010083AFP R7F7010084AFP	R7F7010112AFP R7F7010113AFP R7F7010114AFP	R7F7010162AFP R7F7010163AFP R7F7010164AFP	R7F7010212AFP R7F7010213AFP R7F7010214AFP	—	—
384 KB	32 KB	FF20 0000$_H$ - FF20 7FFF$_H$	R7F7010092AFP R7F7010093AFP R7F7010094AFP	R7F7010122AFP R7F7010123AFP R7F7010124AFP	R7F7010172AFP R7F7010173AFP R7F7010174AFP	R7F7010222AFP R7F7010223AFP R7F7010224AFP	R7F7010262AFP R7F7010263AFP R7F7010264AFP	—
512 KB	32 KB	FF20 0000$_H$ - FF20 7FFF$_H$	R7F7010102AFP R7F7010103AFP R7F7010104AFP	R7F7010132AFP R7F7010133AFP R7F7010134AFP	R7F7010182AFP R7F7010183AFP R7F7010184AFP	R7F7010232AFP R7F7010233AFP R7F7010234AFP R7F7010022AFP R7F7010023AFP R7F7010024AFP	R7F7010272AFP R7F7010273AFP R7F7010274AFP	—
768 KB	32 KB	FF20 0000$_H$ - FF20 7FFF$_H$	—	R7F7010142AFP R7F7010143AFP R7F7010144AFP	R7F7010192AFP R7F7010193AFP R7F7010194AFP	R7F7010242AFP R7F7010243AFP R7F7010244AFP	R7F7010282AFP R7F7010283AFP R7F7010284AFP	R7F7010322AFP R7F7010323AFP R7F7010324AFP
				R7F7010402AFP R7F7010403AFP R7F7010404AFP	R7F7010422AFP R7F7010423AFP R7F7010424AFP	R7F7010442AFP R7F7010443AFP R7F7010444AFP	R7F7010462AFP R7F7010463AFP R7F7010464AFP	R7F7010502AFP R7F7010503AFP R7F7010504AFP
1 MB	32 KB	FF20 0000$_H$ - FF20 7FFF$_H$	—	R7F7010152AFP R7F7010153AFP R7F7010154AFP	R7F7010202AFP R7F7010203AFP R7F7010204AFP	R7F7010252AFP R7F7010253AFP R7F7010254AFP R7F7010032AFP R7F7010033AFP R7F7010034AFP	R7F7010292AFP R7F7010293AFP R7F7010294AFP	R7F7010332AFP R7F7010333AFP R7F7010334AFP
				R7F7010412AFP R7F7010413AFP R7F7010414AFP	R7F7010432AFP R7F7010433AFP R7F7010434AFP	R7F7010452AFP R7F7010453AFP R7F7010454AFP	R7F7010472AFP R7F7010473AFP R7F7010474AFP	R7F7010512AFP R7F7010513AFP R7F7010514AFP
1.5 MB	32 KB	FF20 0000$_H$ - FF20 7FFF$_H$	—	—	—	—	R7F7010302AFP R7F7010303AFP R7F7010304AFP	R7F7010342AFP R7F7010343AFP R7F7010344AFP
							R7F7010482AFP R7F7010483AFP R7F7010484AFP	R7F7010522AFP R7F7010523AFP R7F7010524AFP
	64 KB	FF20 0000$_H$ - FF20 FFFF$_H$	—	—	—	—	R7F7010542AFP R7F7010543AFP R7F7010544AFP	R7F7010562AFP R7F7010563AFP R7F7010564AFP
2 MB	32 KB	FF20 0000$_H$ - FF20 7FFF$_H$	—	—	—	—	R7F7010492AFP R7F7010493AFP R7F7010494AFP	R7F7010532AFP R7F7010533AFP R7F7010534AFP
	64 KB	FF20 0000$_H$ - FF20 FFFF$_H$	—	—	—	—	R7F7010552AFP R7F7010553AFP R7F7010554AFP	R7F7010572AFP R7F7010573AFP R7F7010574AFP

3.2.4.3 Primary Local RAM Area

The primary local RAM can be accessed immediately, without requiring the system to wait. Values are not retained in this area in DEEPSTOP mode or any reset.

The following table lists the primary local RAM sizes and address ranges.

Table 3.4 Primary Local RAM Area

Code Flash Size	Primary Local RAM Size	Address Range	Product Name					
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
256 KB	—	—	R7F7010082AFP R7F7010083AFP R7F7010084AFP	R7F7010112AFP R7F7010113AFP R7F7010114AFP	R7F7010162AFP R7F7010163AFP R7F7010164AFP	R7F7010212AFP R7F7010213AFP R7F7010214AFP	—	—
384 KB	16 KB	FEDF $C000_H$ - FEDF $FFFF_H$	R7F7010092AFP R7F7010093AFP R7F7010094AFP	R7F7010122AFP R7F7010123AFP R7F7010124AFP	R7F7010172AFP R7F7010173AFP R7F7010174AFP	R7F7010222AFP R7F7010223AFP R7F7010224AFP	R7F7010262AFP R7F7010263AFP R7F7010264AFP	—
512 KB	32 KB	FEDF 8000_H - FEDF $FFFF_H$	R7F7010102AFP R7F7010103AFP R7F7010104AFP	R7F7010132AFP R7F7010133AFP R7F7010134AFP	R7F7010182AFP R7F7010183AFP R7F7010184AFP	R7F7010232AFP R7F7010233AFP R7F7010234AFP R7F7010022AFP R7F7010023AFP R7F7010024AFP	R7F7010272AFP R7F7010273AFP R7F7010274AFP	—
768 KB	64 KB	FEDF 0000_H - FEDF $FFFF_H$	—	R7F7010142AFP R7F7010143AFP R7F7010144AFP	R7F7010192AFP R7F7010193AFP R7F7010194AFP	R7F7010242AFP R7F7010243AFP R7F7010244AFP	R7F7010282AFP R7F7010283AFP R7F7010284AFP	R7F7010322AFP R7F7010323AFP R7F7010324AFP
				R7F7010402AFP R7F7010403AFP R7F7010404AFP	R7F7010422AFP R7F7010423AFP R7F7010424AFP	R7F7010442AFP R7F7010443AFP R7F7010444AFP	R7F7010462AFP R7F7010463AFP R7F7010464AFP	R7F7010502AFP R7F7010503AFP R7F7010504AFP
1 MB	96 KB	FEDE 8000_H - FEDF $FFFF_H$	—	R7F7010152AFP R7F7010153AFP R7F7010154AFP	R7F7010202AFP R7F7010203AFP R7F7010204AFP	R7F7010252AFP R7F7010253AFP R7F7010254AFP R7F7010032AFP R7F7010033AFP R7F7010034AFP	R7F7010292AFP R7F7010293AFP R7F7010294AFP	R7F7010332AFP R7F7010333AFP R7F7010334AFP
				R7F7010412AFP R7F7010413AFP R7F7010414AFP	R7F7010432AFP R7F7010433AFP R7F7010434AFP	R7F7010452AFP R7F7010453AFP R7F7010454AFP	R7F7010472AFP R7F7010473AFP R7F7010474AFP	R7F7010512AFP R7F7010513AFP R7F7010514AFP
1.5 MB	128 KB	FEDE 0000_H - FEDF $FFFF_H$	—	—	—	—	R7F7010302AFP R7F7010303AFP R7F7010304AFP	R7F7010342AFP R7F7010343AFP R7F7010344AFP
							R7F7010482AFP R7F7010483AFP R7F7010484AFP	R7F7010522AFP R7F7010523AFP R7F7010524AFP
							R7F7010542AFP R7F7010543AFP R7F7010544AFP	R7F7010562AFP R7F7010563AFP R7F7010564AFP
2 MB	128 KB	FEDE 0000_H - FEDF $FFFF_H$	—	—	—	—	R7F7010492AFP R7F7010493AFP R7F7010494AFP	R7F7010532AFP R7F7010533AFP R7F7010534AFP
							R7F7010552AFP R7F7010553AFP R7F7010554AFP	R7F7010572AFP R7F7010573AFP R7F7010574AFP

3.2.4.4 Secondary Local RAM Area

The secondary local RAM can be accessed in a few cycles of the CPU. Values are not retained in this area in DEEPSTOP mode or any reset.

The following table lists the secondary local RAM sizes and address ranges.

Table 3.5 Secondary Local RAM Area

Code Flash Size	Secondary Local RAM Size	Address Range	Product Name					
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
256 KB	—	—	R7F7010082AFP R7F7010083AFP R7F7010084AFP	R7F7010112AFP R7F7010113AFP R7F7010114AFP	R7F7010162AFP R7F7010163AFP R7F7010164AFP	R7F7010212AFP R7F7010213AFP R7F7010214AFP	—	—
384 KB	—	—	R7F7010092AFP R7F7010093AFP R7F7010094AFP	R7F7010122AFP R7F7010123AFP R7F7010124AFP	R7F7010172AFP R7F7010173AFP R7F7010174AFP	R7F7010222AFP R7F7010223AFP R7F7010224AFP	R7F7010262AFP R7F7010263AFP R7F7010264AFP	—
512 KB	—	—	R7F7010102AFP R7F7010103AFP R7F7010104AFP	R7F7010132AFP R7F7010133AFP R7F7010134AFP	R7F7010182AFP R7F7010183AFP R7F7010184AFP	R7F7010232AFP R7F7010233AFP R7F7010234AFP R7F7010022AFP R7F7010023AFP R7F7010024AFP	R7F7010272AFP R7F7010273AFP R7F7010274AFP	—
768 KB	—	—	—	R7F7010142AFP R7F7010143AFP R7F7010144AFP	R7F7010192AFP R7F7010193AFP R7F7010194AFP	R7F7010242AFP R7F7010243AFP R7F7010244AFP	R7F7010282AFP R7F7010283AFP R7F7010284AFP	R7F7010322AFP R7F7010323AFP R7F7010324AFP
				R7F7010402AFP R7F7010403AFP R7F7010404AFP	R7F7010422AFP R7F7010423AFP R7F7010424AFP	R7F7010442AFP R7F7010443AFP R7F7010444AFP	R7F7010462AFP R7F7010463AFP R7F7010464AFP	R7F7010502AFP R7F7010503AFP R7F7010504AFP
1 MB	—	—	—	R7F7010152AFP R7F7010153AFP R7F7010154AFP	R7F7010202AFP R7F7010203AFP R7F7010204AFP	R7F7010252AFP R7F7010253AFP R7F7010254AFP R7F7010032AFP R7F7010033AFP R7F7010034AFP	R7F7010292AFP R7F7010293AFP R7F7010294AFP	R7F7010332AFP R7F7010333AFP R7F7010334AFP
				R7F7010412AFP R7F7010413AFP R7F7010414AFP	R7F7010432AFP R7F7010433AFP R7F7010434AFP	R7F7010452AFP R7F7010453AFP R7F7010454AFP	R7F7010472AFP R7F7010473AFP R7F7010474AFP	R7F7010512AFP R7F7010513AFP R7F7010514AFP
1.5 MB	—	—	—	—	—	—	R7F7010302AFP R7F7010303AFP R7F7010304AFP	R7F7010342AFP R7F7010343AFP R7F7010344AFP
							R7F7010482AFP R7F7010483AFP R7F7010484AFP	R7F7010522AFP R7F7010523AFP R7F7010524AFP
							R7F7010542AFP R7F7010543AFP R7F7010544AFP	R7F7010562AFP R7F7010563AFP R7F7010564AFP
2 MB	32 KB	FEDD 8000_H - FEDD $FFFF_H$	—	—	—	—	R7F7010492AFP R7F7010493AFP R7F7010494AFP	R7F7010532AFP R7F7010533AFP R7F7010534AFP
							R7F7010552AFP R7F7010553AFP R7F7010554AFP	R7F7010572AFP R7F7010573AFP R7F7010574AFP

3.2.4.5 Retention RAM Area

The retention RAM can be accessed after waiting a few cycles. Values are retained in this area in DEEPSTOP mode or any reset. The values are retained when the power supply voltage is at or above RAM retention voltage even though the power supply voltage falls below the POC voltage.

The following table lists the Retention RAM sizes and address ranges.

Table 3.6 Retention RAM Area

Code Flash Size	Retention RAM Size	Address Range	Product Name					
			48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
256 KB	32 KB	FEE0 0000$_H$ - FEE0 7FFF$_H$	R7F7010082AFP R7F7010083AFP R7F7010084AFP	R7F7010112AFP R7F7010113AFP R7F7010114AFP	R7F7010162AFP R7F7010163AFP R7F7010164AFP	R7F7010212AFP R7F7010213AFP R7F7010214AFP	—	—
384 KB	32 KB	FEE0 0000$_H$ - FEE0 7FFF$_H$	R7F7010092AFP R7F7010093AFP R7F7010094AFP	R7F7010122AFP R7F7010123AFP R7F7010124AFP	R7F7010172AFP R7F7010173AFP R7F7010174AFP	R7F7010222AFP R7F7010223AFP R7F7010224AFP	R7F7010262AFP R7F7010263AFP R7F7010264AFP	—
512 KB	32 KB	FEE0 0000$_H$ - FEE0 7FFF$_H$	R7F7010102AFP R7F7010103AFP R7F7010104AFP	R7F7010132AFP R7F7010133AFP R7F7010134AFP	R7F7010182AFP R7F7010183AFP R7F7010184AFP	R7F7010232AFP R7F7010233AFP R7F7010234AFP R7F7010022AFP R7F7010023AFP R7F7010024AFP	R7F7010272AFP R7F7010273AFP R7F7010274AFP	—
768 KB	32 KB	FEE0 0000$_H$ - FEE0 7FFF$_H$	—	R7F7010142AFP R7F7010143AFP R7F7010144AFP	R7F7010192AFP R7F7010193AFP R7F7010194AFP	R7F7010242AFP R7F7010243AFP R7F7010244AFP	R7F7010282AFP R7F7010283AFP R7F7010284AFP	R7F7010322AFP R7F7010323AFP R7F7010324AFP
				R7F7010402AFP R7F7010403AFP R7F7010404AFP	R7F7010422AFP R7F7010423AFP R7F7010424AFP	R7F7010442AFP R7F7010443AFP R7F7010444AFP	R7F7010462AFP R7F7010463AFP R7F7010464AFP	R7F7010502AFP R7F7010503AFP R7F7010504AFP
1 MB	32 KB	FEE0 0000$_H$ - FEE0 7FFF$_H$	—	R7F7010152AFP R7F7010153AFP R7F7010154AFP	R7F7010202AFP R7F7010203AFP R7F7010204AFP	R7F7010252AFP R7F7010253AFP R7F7010254AFP R7F7010032AFP R7F7010033AFP R7F7010034AFP	R7F7010292AFP R7F7010293AFP R7F7010294AFP	R7F7010332AFP R7F7010333AFP R7F7010334AFP
				R7F7010412AFP R7F7010413AFP R7F7010414AFP	R7F7010432AFP R7F7010433AFP R7F7010434AFP	R7F7010452AFP R7F7010453AFP R7F7010454AFP	R7F7010472AFP R7F7010473AFP R7F7010474AFP	R7F7010512AFP R7F7010513AFP R7F7010514AFP
1.5 MB	32 KB	FEE0 0000$_H$ - FEE0 7FFF$_H$	—	—	—	—	R7F7010302AFP R7F7010303AFP R7F7010304AFP	R7F7010342AFP R7F7010343AFP R7F7010344AFP
							R7F7010482AFP R7F7010483AFP R7F7010484AFP	R7F7010522AFP R7F7010523AFP R7F7010524AFP
							R7F7010542AFP R7F7010543AFP R7F7010544AFP	R7F7010562AFP R7F7010563AFP R7F7010564AFP
2 MB	32 KB	FEE0 0000$_H$ - FEE0 7FFF$_H$	—	—	—	—	R7F7010492AFP R7F7010493AFP R7F7010494AFP	R7F7010532AFP R7F7010533AFP R7F7010534AFP
							R7F7010552AFP R7F7010553AFP R7F7010554AFP	R7F7010572AFP R7F7010573AFP R7F7010574AFP

CAUTION

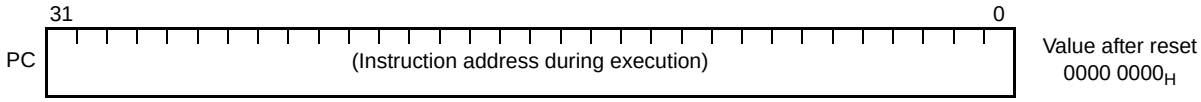
Instruction fetching and data access from an un-initialized local RAM area may generate an ECC error interrupt or an SYSERR exception. Before reading from a local RAM area, initialize the entire local RAM area.

3.3 Register Reference

This section describes the items included in Section 3 of the *RH850 Family Software User's Manual* that require specific attention when using the RH850/F1L.

3.3.1 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.



CAUTION

For a CPU whose instruction addressing range is 128 MB, a value resulting from a sign-extension of bit 26 is automatically set to bits 31 to 27.

3.3.2 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 selection ID.

Table 3.7 Basic System Registers (1/2)

Register No. (regID, selID)	Symbol	Function	Changes from the RH850 Software Manual	Access Permission
SR0, 0	EIPC*1	Status save registers when acknowledging EI level exception	Some changes in specifications	SV
SR1, 0	EIPSW	Status save registers when acknowledging EI level exception	Some changes in specifications	SV
SR2, 0	FEPC*1	Status save registers when acknowledging FE level exception	Some changes in specifications	SV
SR3, 0	FEPSW	Status save registers when acknowledging FE level exception	Some changes in specifications	SV
SR 5, 0	PSW	Program status word	Some changes in specifications	*2
SR 6, 0	FPSR	Not implemented. A value of 0 is returned when read and writing is ignored.		CU0 = 1 and SV*3
SR 7, 0	FPEPC	Not implemented. A value of 0 is returned when read and writing is ignored.		CU0 = 1 and SV*3
SR8, 0	FPST	Not implemented. A value of 0 is returned when read and writing is ignored.		CU0 = 1*3
SR9, 0	FPCC	Not implemented. A value of 0 is returned when read and writing is ignored.		CU0 = 1*3
SR10, 0	FPCFG	Not implemented. A value of 0 is returned when read and writing is ignored.		CU0 = 1*3
SR 11, 0	FPEC	Not implemented. A value of 0 is returned when read and writing is ignored.		CU0 = 1 and SV*3
SR 12, 0	SESR	Not implemented. A value of 0 is returned when read and writing is ignored.		CU0 = 1*4
SR 13, 0	EIIC	EI level exception cause	No change	SV
SR 14, 0	FEIC	FE level exception cause	No change	SV
SR 16, 0	CTPC*1	CALLT execution status save register	Some changes in specifications	UM
SR 17, 0	CTPSW	CALLT execution status save register	No change	UM
SR 20, 0	CTBP*1	CALLT base pointer	Some changes in specifications	UM
SR 28, 0	EIWR	EI level exception working register	No change	SV
SR 29, 0	FEWR	FE level exception working register	No change	SV
SR 31, 0	BSEL	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR0, 1	MCFG0	Machine configuration 0	Some changes in specifications	SV
SR1, 1	MCFG1	Machine configuration 1	Some changes in specifications	SV
SR2, 1	RBASE*1	Reset vector base address	Some changes in specifications	SV
SR3, 1	EBASE*1	Exception handler vector address	Some changes in specifications	SV
SR4, 1	INTBP*1	Base address of the interrupt handler table	Some changes in specifications	SV
SR 5, 1	MCTL	CPU control	Some changes in specifications	SV
SR 6, 1	PID	Processor ID	Some changes in specifications	SV
SR 11, 1	SCCFG	SYSCALL operation setting	No change	SV
SR 12, 1	SCBP*1	SYSCALL base pointer	Some changes in specifications	SV
SR0, 2	HTCFG0	Thread configuration	Some changes in specifications	SV

Table 3.7 Basic System Registers (2/2)

Register No. (regID, selID)	Symbol	Function	Changes from the RH850 Software Manual	Access Permission
SR6, 2	MEA	Memory error address	No change	SV
SR7, 2	ASID	Address space ID	No change	SV
SR8, 2	MEI	Memory error information	Some changes in specifications	SV

Note 1. For a CPU whose instruction addressing range is 128 MB, a value resulting from a sign-extension of bit 26 is automatically set to bits 31 to 27.

Note 2. The access permission differs depending on the bit. For details, see **Table 3.12, Access Permission for PSW Register**.

Note 3. In RH850/F1L, CU0 is fixed to “0”. There is no access right.

Note 4. In RH850/F1L, CU1 is fixed to “0”. There is no access right.

The following registers are described in *Section 3.2.1, Basic registers (specific to hardware threads)*, in the *RH850 Family User's Manual: Software*.

- **Section 3.3.2.1, EIPC — Status Save Register when Acknowledging EI Level Exception**
- **Section 3.3.2.2, EIPSW — Status Save Register when Acknowledging EI Level Exception**
- **Section 3.3.2.3, FEPC — Status Save Register when Acknowledging FE Level Exception**
- **Section 3.3.2.4, FEPSW — Status Save Register when Acknowledging FE Level Exception**
- **Section 3.3.2.5, PSW — Program Status Word**
- **Section 3.3.2.6, CTPC — Status Save Register when Executing CALLT**
- **Section 3.3.2.7, CTBP — CALLT Base Pointer**
- **Section 3.3.2.8, HTCFG0 — Thread Configuration**
- **Section 3.3.2.9, MEI — Memory Error Information**

The following registers are described in *Section 3.2.2, Basic registers (specific to virtual machines)*, in the *RH850 Family User's Manual: Software*.

- **Section 3.3.2.10, RBASE — Reset Vector Base Address**
- **Section 3.3.2.11, EBASE — Exception Handler Vector Address**
- **Section 3.3.2.12, INTBP — Base Address of the Interrupt Handler Table**
- **Section 3.3.2.13, PID — Processor ID**
- **Section 3.3.2.14, SCBP — SYSCALL Base Pointer**
- **Section 3.3.2.15, MCFG0 — Machine Configuration 0**
- **Section 3.3.2.16, MCFG1 — Machine Configuration 1**
- **Section 3.3.2.17, MCTL — Machine Control**

3.3.2.1 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 6.1.3, Types of exceptions* of the *RH850 Family 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 must not be specified.

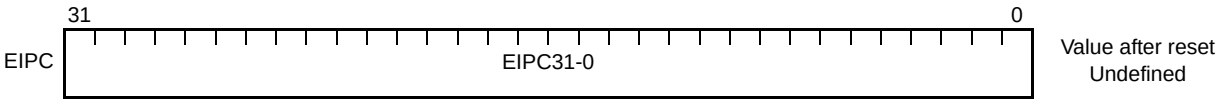


Table 3.8 EIPC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	EIPC31-1	These bits indicate the PC saved when an EI level exception is acknowledged.	R/W	Undefined
0	EIPC0	This bit indicates the PC saved 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

CAUTION

For a CPU whose instruction addressing range is 128 MB, a value resulting from a sign-extension of bit 26 is automatically set to bits 31 to 27 of EIPC.

3.3.2.2 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.

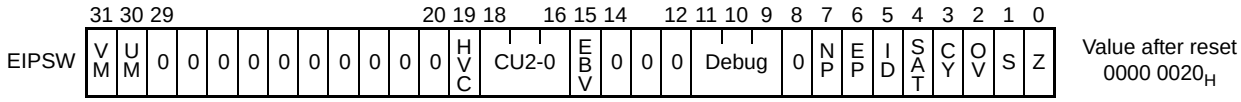


Table 3.9 EIPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31	VM	The RH850/F1L does not support this bit.	R	0
30	UM	This bit stores the PSW.UM bit setting when an EI level exception is acknowledged.	R/W	0
29 to 20	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
19	HVC	The RH850/F1L does not support this bit.	R	0
18 to 16	CU2-0	The RH850/F1L does not support these bits.	R	0
15	EBV	This bit stores 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	0
8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an EI level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an EI level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an EI level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an EI level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an EI level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an EI level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an EI level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an EI level exception is acknowledged.	R/W	0

3.3.2.3 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 6.1.3, Types of exceptions* of the *RH850 Family 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 must not be specified.

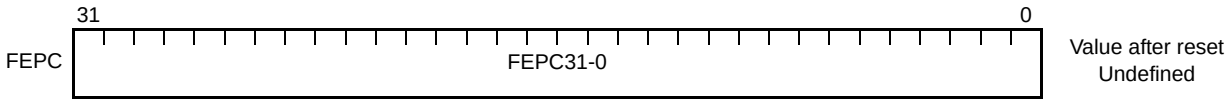


Table 3.10 FEPC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 1	FEPC31-1	These bits indicate the PC saved when an FE level exception is acknowledged.	R/W	Undefined
0	FEPC0	This bit indicates the PC saved 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

CAUTION

For a CPU whose instruction addressing range is 128 MB, a value resulting from a sign-extension of bit 26 is automatically set to bits 31 to 27 of FEPC.

3.3.2.4 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.

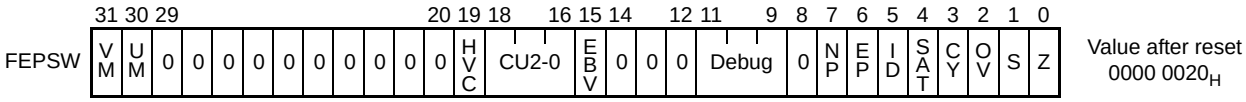


Table 3.11 FEPSW Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31	VM	The RH850/F1L does not support this bit.	R	0
30	UM	This bit stores the PSW.UM bit setting when an FE level exception is acknowledged.	R/W	0
29 to 20	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
19	HVC	The RH850/F1L does not support this bit.	R	0
18 to 16	CU2-0	The RH850/F1L does not support these bits.	R	0
15	EBV	This bit stores 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 store the PSW.Debug field when an FE level exception is acknowledged.	R	0
8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
7	NP	This bit stores the PSW.NP bit setting when an FE level exception is acknowledged.	R/W	0
6	EP	This bit stores the PSW.EP bit setting when an FE level exception is acknowledged.	R/W	0
5	ID	This bit stores the PSW.ID bit setting when an FE level exception is acknowledged.	R/W	1
4	SAT	This bit stores the PSW.SAT bit setting when an FE level exception is acknowledged.	R/W	0
3	CY	This bit stores the PSW.CY bit setting when an FE level exception is acknowledged.	R/W	0
2	OV	This bit stores the PSW.OV bit setting when an FE level exception is acknowledged.	R/W	0
1	S	This bit stores the PSW.S bit setting when an FE level exception is acknowledged.	R/W	0
0	Z	This bit stores the PSW.Z bit setting when an FE level exception is acknowledged.	R/W	0

3.3.2.5 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 a condition instruction (Bcond, CMOV, etc.)).

When the LDSR instruction is used to change the contents of bits 7 to 0 in this register, the changed contents become valid from the instruction following the LDSR instruction.

The access permission for the PSW register differs depending on the bit. All bits can be read, but some bits can only be written under certain conditions. See **Table 3.12** for the access permission for each bit.

Table 3.12 Access Permission for PSW Register

Bit		Access Permission When Reading	Access Permission When Writing
30	UM	UM	SV*1
15	EBV		SV*1
7	NP		SV*1
6	EP		SV*1
5	ID		SV*1
4	SAT		UM
3	CY		UM
2	OV		UM
1	S		UM
0	Z		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.

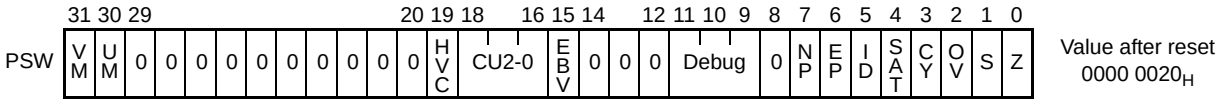


Table 3.13 PSW Register Contents (1/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
31	VM	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0
30	UM	This bit indicates that the CPU is in the user mode (the UM mode) 0: Supervisor mode 1: User mode	R/W	0
29 to 20	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
19	HVC	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0
18 to 16	CU2-0	The RH850/F1L does not support these bits. Be sure to clear to 0.	R	000
15	EBV	This bit indicates the reset vector and exception vector operation. For details, see the RBASE register and EBASE register in **Section 3.3.2.10, RBASE — Reset Vector Base Address**.	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 function for the development tool. In normal cases, set these bits to “0”.	R	0
8	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

Table 3.13 PSW Register Contents (2/2)

Bit Position	Bit Name	Function	R/W	Value After Reset
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. 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 interrupt is 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. This bit is also used to disable EI level exceptions from being acknowledged as a critical section while an ordinary program or interrupt is being serviced. It is set to 1 when the DI instruction is executed, and cleared to 0 when the EI instruction is executed. 0: EI level exception is 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. When only the OV flag is set to 1 during a saturated operation, the SAT flag is set to 1.

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	

3.3.2.6 CTPC — Status Save Register when Executing CALLT

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 must not be specified.

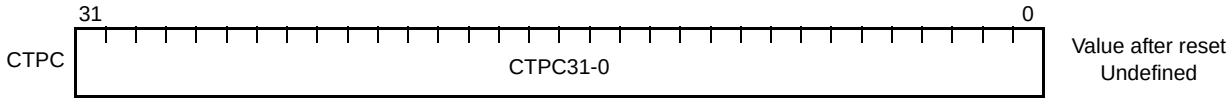


Table 3.14 CTPC Register Contents

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

CAUTION

For a CPU whose instruction addressing range is 128 MB, a value resulting from a sign-extension of bit 26 is automatically set to bits 31 to 27 of CTPC.

3.3.2.7 CTBP — CALLT Base Pointer

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.

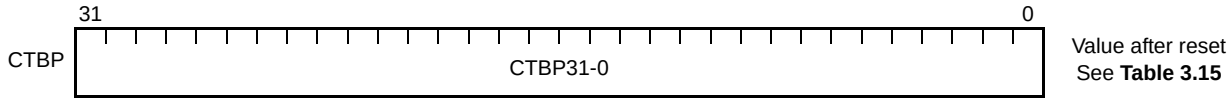


Table 3.15 CTBP Register Contents

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

CAUTION

For a CPU whose instruction addressing range is 128 MB, a value resulting from a sign-extension of bit 26 is automatically set to bits 31 to 27 of CTBP.

3.3.2.8 HTCFG0 — Thread Configuration

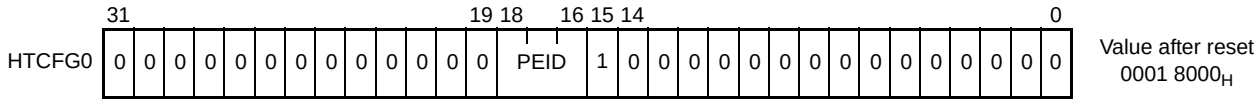


Table 3.16 HTCFG0 Register Contents

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	001_B
15	—	(Be sure to set this bit to 1. A value of 1 will be returned when read.)	R	1
14 to 0	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

3.3.2.9 MEI — Memory Error Information

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

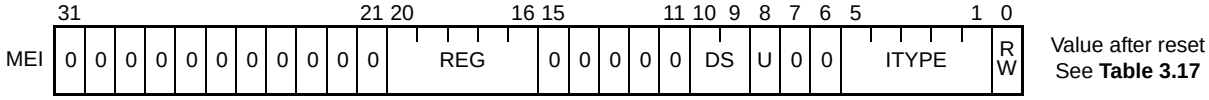


Table 3.17 MEI Register Contents

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 3.18**.	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 type of data handled by 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 3.18**.	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 3.18**.	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 3.18**.	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 3.18**.	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 3.18 Instructions Causing Exceptions and Values of MEI Register

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
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
Interrupt (table reference)*2	—	2 (word)	1 (unsigned)	0 (read)	10101_B

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

Note 2. An exception occurs when the table reference interrupt vector is read.

NOTE

dst: destination register number, src: source register number

Misaligned Exception

Access to data at an address that does not correspond to the required boundary for the unit of access leads to a misaligned exception. For example, reading or writing a word from or to an address that is not a multiple of 4 leads to an exception of this type. However, if the type is double-word, access at any word-form boundary does not lead to an exception. When MCTL.MA is 1, access proceeds without generating misaligned exceptions.

3.3.2.10 RBASE — Reset Vector Base Address

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.

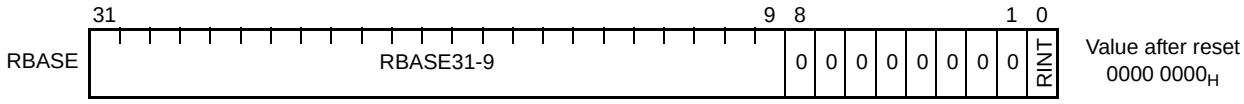


Table 3.19 RBASE Register Contents

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 not assigned as names because these bits are always 0.	R	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 offset address of the exception handler used during interrupt servicing is decremented. This bit is valid when PSW.EBV = 0.	R	0

Note 1. When the reset vector is changed by self programming, the address will be changed.

3.3.2.11 EBASE — Exception Handler Vector Address

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

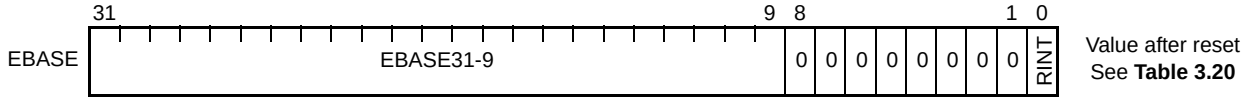


Table 3.20 EBASE Register Contents

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 EBASE8-1 bits are not assigned as names because these bits are always 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.	R/W	Undefined

CAUTION

For a CPU whose instruction addressing range is 128 MB, a value resulting from a sign-extension of bit 26 is automatically set to bits 31 to 27 of EBASE.

3.3.2.12 INTBP — Base Address 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.

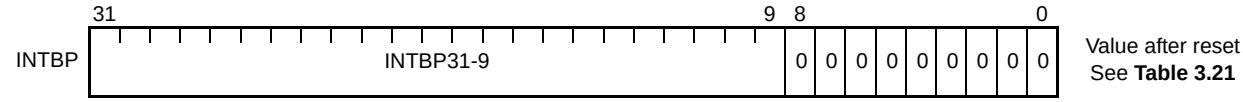


Table 3.21 INTBP Register Contents

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 interrupt prescribed by the expanded specifications (EIINT0 to 287) is acknowledged. The INTBP8-0 bits are not assigned as names because these bits are always 0.	R/W	Undefined
8 to 0	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

CAUTION

For a CPU whose instruction addressing range is 128 MB, a value resulting from a sign-extension of bit 26 is automatically set to bits 31 to 27 of INTBP.

3.3.2.13 PID — Processor ID

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

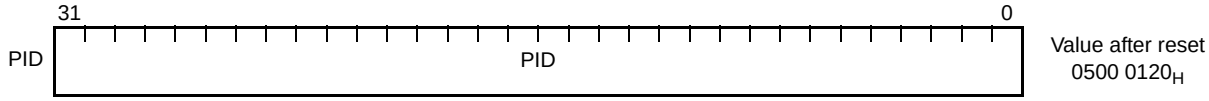


Table 3.22 PID Register Contents

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 9 Reserved Bit 8 Memory protection unit (MPU) function	R	0001_H
7 to 0		Version identifier This identifier indicates the version of the processor.	R	20_H

3.3.2.14 SCBP — SYSCALL Base Pointer

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.

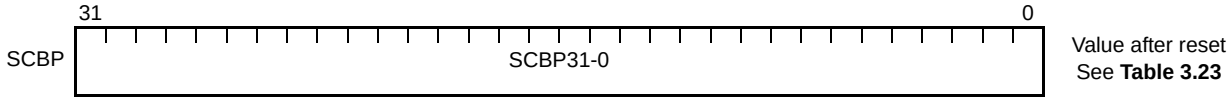


Table 3.23 SCBP Register Contents

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 this bit to 0.	R	0

CAUTION

For a CPU whose instruction addressing range is 128 MB, a value resulting from a sign-extension of bit 26 is automatically set to bits 31 to 27 of SCBP.

3.3.2.15 MCFG0 — Machine Configuration 0

This register indicates the CPU configuration.

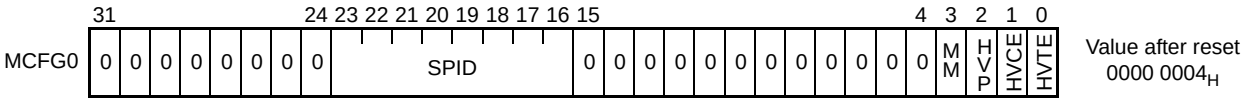


Table 3.24 MCFG0 Register Contents

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	Not supported by the RH850/F1L. When writing, always write 0.	R/W	0000 0000
15 to 4	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
3	MM	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0
2	HVP	The RH850/F1L does not support this bit. Be sure to clear to 1.	R	1
1	HVCE	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0
0	HVTE	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0

3.3.2.16 MCFG1 — Machine Configuration 1

This register indicates the CPU configuration.

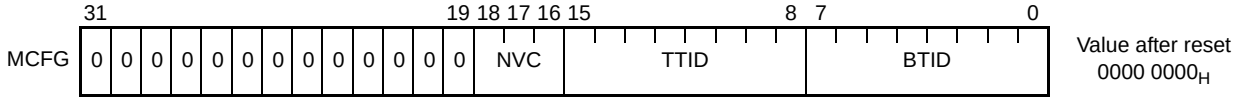


Table 3.25 MCFG1 Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 19	—	(Reserved for future expansion. Be sure to clear these bits to 0.)	R	0
18 to 16	NVC	The RH850/F1L does not support these bits. Be sure to clear to 0.	R	000
15 to 8	TTID	The RH850/F1L does not support these bits. Be sure to clear to 0.	R	0000 0000
7 to 0	BTID	The RH850/F1L does not support these bits. Be sure to clear to 0.	R	0000 0000

3.3.2.17 MCTL — Machine Control

This register is used to control the CPU.

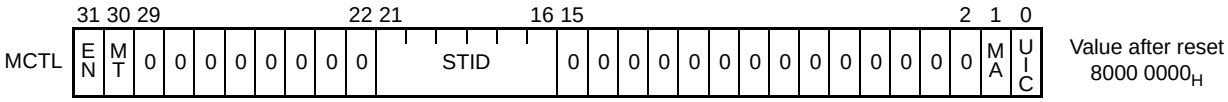


Table 3.26 MCTL Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31	EN	The RH850/F1L does not support this bit. Be sure to set this bit to 1.	R	1
30	MT	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0
29 to 22	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
21 to 16	STID	The RH850/F1L does not support these bits. Be sure to clear to 0.	R	000000
15 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: Hardware operates normally.	R/W	0
0	UIC	This bit is used to control the interrupt enable/disable operation in the user mode. When this bit is set to 1, executing the EI/DI instruction become possible.	R/W	0

3.3.3 Interrupt Function Registers

Table 3.27 Interrupt Function System Registers

Register No. (regID, selID)	Symbol	Function	Changes from the RH850 Software Manual	Access Permission
SR7, 1	FPIPR	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR10, 2	ISPR	Priority of interrupt being serviced	Some changes in specifications	SV
SR11, 2	PMR	Interrupt priority masking	Some changes in specifications	SV
SR12, 2	ICSR	Interrupt control status	Some changes in specifications	SV
SR13, 2	INTCFG	Interrupt function setting	No change	SV

3.3.3.1 ISPR — Priority of Interrupt being Serviced

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

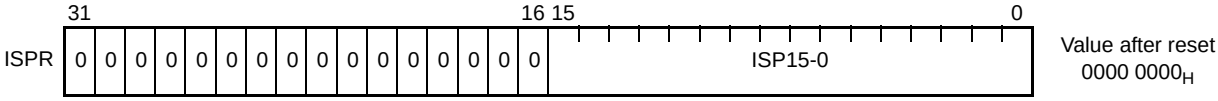


Table 3.28 ISPR Register Contents

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 8	ISP15-8	These bits are not used in the RH850/F1L. Be sure to clear these bits to 0.	R	0
7 to 0	ISP7-0	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 ... / ... 6 / Priority 6 7 / Priority 7 (lowest) 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 ISP7-0 bits that are set (0 is the highest priority) is cleared to 0*2. While a bit in this register is set to 1, lower priority interrupts (EIINTn) 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. For details, see *Section 6.1.5, Interrupt exception priority and priority masking of the RH850 Family User's Manual: Software*. When performing software-based priority control using the PMR register, be sure to clear this register by using the INTCFG.ISPC bit.	R*3	0

Note 1. For details, see *Section 6.1.5, Interrupt exception priority and priority masking* of the *RH850 Family User's Manual: Software*.

Note 2. Interrupt acknowledgment and auto-updating of values when the EIRET instruction is executed are disabled by setting 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. 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.

3.3.3.2 PMR — Interrupt Priority Masking

This register is used to mask the specified interrupt priority.

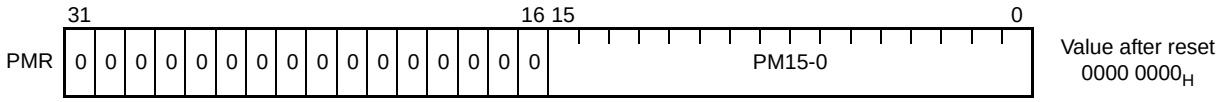


Table 3.29 PMR Register Contents

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 8	PM15-8	These bits are not used in the RH850/F1L. Be sure to clear these bits to 0.	R	0
7 to 0	PM7-0	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 ... 6 — Priority 6 7 — Priority 7 (lowest) While a bit in this register is set to 1, interrupts (EIINTn) 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*1.	R/W	0

Note 1. Specify the masks by setting the bits to 1 in order from the lowest-priority bit. For example, $F0_H$ can be set, but 10_H or $0F_H$ cannot.

3.3.3.3 ICSR — Interrupt Control Status

This register indicates the interrupt control status in the CPU.

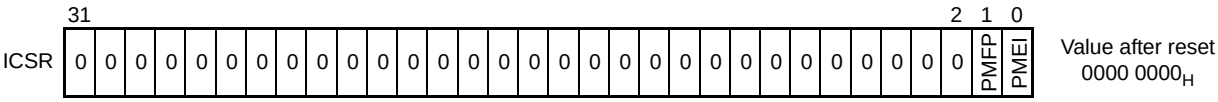


Table 3.30 ICSR Register Contents

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	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0
0	PMEI	This bit indicates that an interrupt (EIINTn) with the priority level masked by the PMR register exists.	R	0

3.3.3.4 INTCFG — Interrupt Function Setting

This register is used to specify the interrupt function in the CPU.

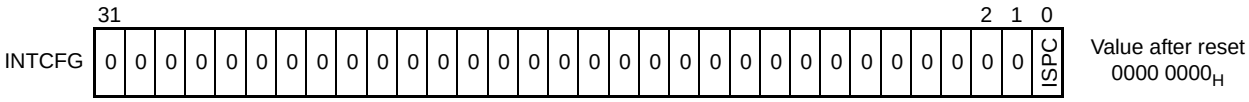


Table 3.31 INTCFG Register Contents

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 the programming to modify the ISPR register. 0: ISPR is automatically updated. Updating by the program (LDSR, LDTC.SR) is ignored. 1: ISPR is not automatically updated. Updating by the program (LDSR, LDTC.SR) is performed. While this bit is cleared to 0, the respective bits in the ISPR register are automatically set to 1 by the acceptance of an interrupt (EIINTn) or automatically cleared to 0 by the execution of the EIRET instruction. In this state, updating by the program (LDSR, LDTC.SR) is ignored. While this bit is set to 1, the respective bits in the ISPR register are not updated by the acceptance of an interrupt (EIINTn) or execution of the EIRET instruction. In this state, updating by the program (LDSR, LDTC.SR) is enabled. Normally, use the ISPC bit being cleared to 0.If the interrupt priority is controlled by software, set this bit to 1 and control the priority by the PMR register.	R/W	0

3.3.4 Hardware Thread Function Registers

Table 3.32 Hardware Thread Function System Registers

Register No. (regID, selID)	Symbol	Function	Changes from the RH850 Software Manual	Access Permission
SR10, 1	TCSEL	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR5, 2	HTCTL	Hardware thread control	Some changes in specifications	SV
SR23, 2	HTSCCTL	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR24, 2	HTSCTBL0	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR25, 2	HTSCTBL1	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR26, 2	HTSCTBL2	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR27, 2	HTSCTBL3	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR28, 2	HTSCTBL4	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR29, 2	HTSCTBL5	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR30, 2	HTSCTBL6	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR31, 2	HTSCTBL7	Not implemented. A value of 0 is returned when read and writing is ignored.	-	SV

3.3.4.1 HTCTL — Hardware Thread Control

	31	30	29																												
HTCTL	EN	HLT	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

Value after reset
$8000\ 0000_H$

Table 3.33 HTCTL Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31	EN	The RH850/F1L does not support this bit. Be sure to set this bit to 1.	R	1
30	HLT	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0
29 to 0	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0

3.3.5 Virtualization Support Function Registers

The RH850/F1L does not include a CPU virtualization support function, so all the following registers return a value of 0 when read, and writing to these registers is ignored.

Table 3.34 Virtual Machine Control System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR13, 1	HVCCFG	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR14, 1	HVCBP	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR15, 1	VCSEL	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR16, 1	VMPRT0	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR17, 1	VMPRT1	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR18, 1	VMPRT2	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR23, 1	VMSCCTL	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR24, 1	VMSCTBL0	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR25, 1	VMSCTBL1	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR26, 1	VMSCTBL2	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR27, 1	VMSCTBL3	Not implemented. A value of 0 is returned when read and writing is ignored.	SV

3.3.6 FPU Function Registers

The RH850/F1L does not include an FPU, so the following registers do not exist. If any of these registers are accessed (read or written), therefore, a UCPOP exception (coprocessor unusable exception) occurs.

Table 3.35 FPU System Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR6, 0	FPSR	Not implemented. A value of 0 is returned when read and writing is ignored.	CU0 = 1 and SV[*1]
SR7, 0	FPEPC	Not implemented. A value of 0 is returned when read and writing is ignored.	CU0 = 1 and SV[*1]
SR8, 0	FPST	Not implemented. A value of 0 is returned when read and writing is ignored.	CU0 = 1[*1]
SR9, 0	FPCC	Not implemented. A value of 0 is returned when read and writing is ignored.	CU0 = 1[*1]
SR10, 0	FPCFG	Not implemented. A value of 0 is returned when read and writing is ignored.	CU0 = 1[*1]
SR11, 0	FPEC	Not implemented. A value of 0 is returned when read and writing is ignored.	CU0 = 1 and SV[*1]

Note 1. In RH850/F1L, CU0 is fixed to “0”. There is no access right.

3.3.7 SIMD Function Registers

The RH850/F1L does not include an SIMD function, so there are no vector registers. If the SESR (SIMD operation setting/status) register is accessed (read or written), therefore, a UCPOP exception (coprocessor unusable exception) occurs.

Table 3.36 SIMD Function Register

Register No. (regID, selID)	Symbol	Function	Access Permission
SR12, 0	SESR	Not implemented. A value of 0 is returned when read and writing is ignored.	CU1 = 1[*1]

Note 1. In RH850/F1L, CU1 is fixed to “0”. There is no access right.

3.3.8 MMU Function Registers

The RH850/F1L does not include an MMU function, so all the following registers return a value of 0 when read, and writing to these registers is ignored.

Table 3.37 MMU Function Register

Register No. (regID, selID)	Symbol	Function	Access Permission
SR16, 2	TLBSCH	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR0, 4	TLBIDX	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR4, 4	TELO0	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR5, 4	TELO1	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR6, 4	TEHI0	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR7, 4	TEHI1	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR10, 4	TLBCFG	Not implemented. A value of 0 is returned when read and writing is ignored.	SV

3.3.9 MPU Function Registers

Table 3.38 MPU Function System Registers (1/2)

Register No. (regID, selID)	Symbol	Function	Changes from the RH850 Software Manual	Access Permission
SR0, 5	MPM	Memory protection operation mode setting	Some changes in specifications	SV
SR1, 5	MPRC	MPU region control	Some changes in specifications	SV
SR4, 5	MPBRGN	MPU base region number	Some changes in specifications	SV
SR5, 5	MPTRGN	MPU end region number	Some changes in specifications	SV
SR8, 5	MCA	Memory protection setting check address	Some changes in specifications	SV
SR9, 5	MCS	Memory protection setting check size	Some changes in specifications	SV
SR10, 5	MCC	Memory protection setting check command	Some changes in specifications	SV
SR11, 5	MCR	Memory protection setting check result	Some changes in specifications	SV
SR20, 5	MPPRT0	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR21, 5	MPPRT1	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR22, 5	MPPRT2	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR0, 6	MPLA0	Protection area minimum address	Some changes in specifications	SV
SR1, 6	MPUA0	Protection area maximum address	Some changes in specifications	SV
SR2, 6	MPAT0	Protection area attribute	Some changes in specifications	SV
SR4, 6	MPLA1	Protection area minimum address	Some changes in specifications	SV
SR5, 6	MPUA1	Protection area maximum address	Some changes in specifications	SV
SR6, 6	MPAT1	Protection area attribute	Some changes in specifications	SV
SR8, 6	MPLA2	Protection area minimum address	Some changes in specifications	SV
SR9, 6	MPUA2	Protection area maximum address	Some changes in specifications	SV
SR10, 6	MPAT2	Protection area attribute	Some changes in specifications	SV
SR12, 6	MPLA3	Protection area minimum address	Some changes in specifications	SV
SR13, 6	MPUA3	Protection area maximum address	Some changes in specifications	SV
SR14, 6	MPAT3	Protection area attribute	Some changes in specifications	SV

Table 3.38 MPU Function System Registers (2/2)

Register No. (regID, selID)	Symbol	Function	Changes from the RH850 Software Manual	Access Permission
SR16, 6	MPLA4	Not implemented. A value of 0 is returned when read and writing is ignored.		SV
SR17, 6	MPUA4			SV
SR18, 6	MPAT4			SV
SR20, 6	MPLA5			SV
SR21, 6	MPUA5			SV
SR22, 6	MPAT5			SV
SR24, 6	MPLA6			SV
SR25, 6	MPUA6			SV
SR26, 6	MPAT6			SV
SR28, 6	MPLA7			SV
SR29, 6	MPUA7			SV
SR30, 6	MPAT7			SV
SR0, 7	MPLA8			SV
SR1, 7	MPUA8			SV
SR2, 7	MPAT8			SV
SR4, 7	MPLA9			SV
SR5, 7	MPUA9			SV
SR6, 7	MPAT9			SV
SR8, 7	MPLA10			SV
SR9, 7	MPUA10			SV
SR10, 7	MPAT10			SV
SR12, 7	MPLA11			SV
SR13, 7	MPUA11			SV
SR14, 7	MPAT11			SV
SR16, 7	MPLA12			SV
SR17, 7	MPUA12			SV
SR18, 7	MPAT12			SV
SR20, 7	MPLA13			SV
SR21, 7	MPUA13			SV
SR22, 7	MPAT13			SV
SR24, 7	MPLA14			SV
SR25, 7	MPUA14			SV
SR26, 7	MPAT14			SV
SR28, 7	MPLA15			SV
SR29, 7	MPUA15			SV
SR30, 7	MPAT15			SV

3.3.9.1 MPM — Memory Protection Operation Mode

The memory protection mode register is used to define the basic operating state of the memory protection function. The settings for this register are normally specified once on startup and are not changed during program operation.

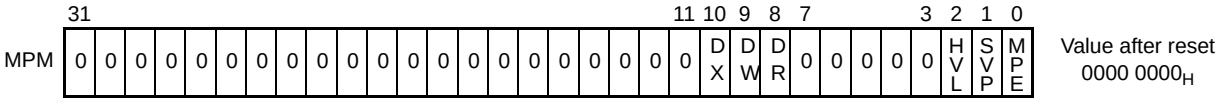


Table 3.39 MPM Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 11	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
10	DX	This bit specifies the default operation when an instruction is executed at an address that does not exist in a protection area. 0: Disable executing an instruction at an address that does not exist in a protection area. 1: Enable executing an instruction at an address that does not exist in a protection area. The setting of this bit affects the access operation when the protection areas overlap.	R/W	0
9	DW	This bit specifies the default operation when writing to an address that does not exist in a protection area. 0: Disable writing to an address that does not exist in a protection area. 1: Enable writing to an address that does not exist in a protection area. The setting of this bit affects the access operation when the protection areas overlap.	R/W	0
8	DR	This bit specifies the default operation when reading from an address that does not exist in a protection area. 0: Disable reading from an address that does not exist in a protection area. 1: Enable reading from an address that does not exist in a protection area. The setting of this bit affects the access operation when the protection areas overlap.	R/W	0
7 to 3	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
2	HVL	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0
1	SVP	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0
0	MPE	This bit is used to specify whether to enable or disable MPU function. 0: Disable 1: Enable	R/W	0

3.3.9.2 MPRC — MPU Region Control

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

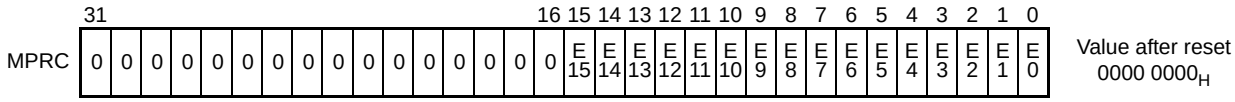


Table 3.40 MPRC Register Contents

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 4	E15-E4	These bits are not used in the RH850/F1L. Be sure to clear these bits to 0.	R	0
3 to 0	E3-E0	These are the enable bits for each protection area. Bit En is a copy of bit MPATn.E (where n = 3 to 0).	R/W	0

3.3.9.3 MPBRGN — MPU Base Region

This register indicates the minimum usable MPU area number.

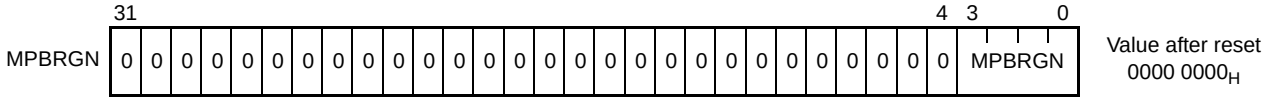


Table 3.41 MPBRGN Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 4	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
3 to 0	MPBRGN	The RH850/F1L does not support these bits. Be sure to clear to 0.	R	0000_B

3.3.9.4 MPTRGN — MPU End Region

This register indicates the maximum usable MPU area number.

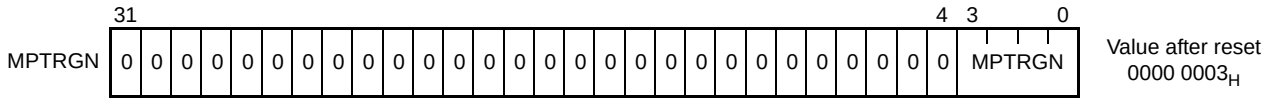


Table 3.42 MPTRGN Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 4	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
3 to 0	MPTRGN	The RH850/F1L does not support these bits. Be sure to set these bits to 0011_B.	R	0011_B

3.3.9.5 MCA — Memory Protection Setting Check Address

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

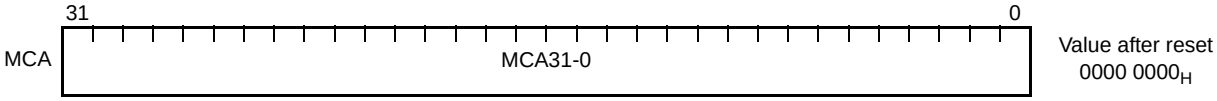


Table 3.43 MCA Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	MCA31-0	The RH850/F1L does not support these bits. Be sure to clear to 0.	R	0

CAUTION

There is no memory protection setting check function, but the MCA register can be manipulated.

However, writing to the MCA register is always ignored, and this register always returns a value of 0000 0000_H when read.

3.3.9.6 MCS — Memory Protection Setting Check Size

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

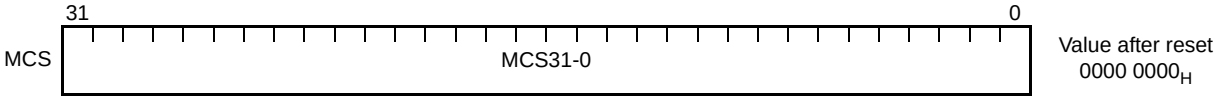


Table 3.44 MCS Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	MCS31-0	The RH850/F1L does not support these bits. Be sure to clear to 0.	R	0

CAUTION

There is no memory protection setting check function, but the MCS register can be manipulated.

However, writing to the MCS register is always ignored, and this register always returns a value of 0000 0000_H when read.

3.3.9.7 MCC — Memory Protection Setting Check Command

This command register is used to start a memory protection setting check.

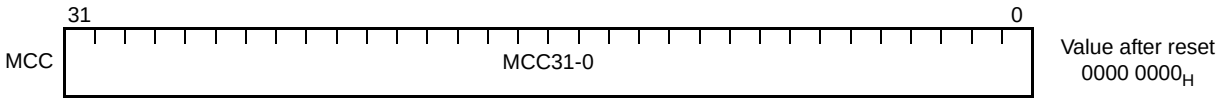


Table 3.45 MCC Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 0	MCC31-0	The RH850/F1L does not support these bits. Be sure to clear to 0.	R	0

CAUTION

There is no memory protection setting check function, but the MCC register can be manipulated.

However, writing to the MCC register is always ignored, and this register always returns a value of 0000 0000_H when read.

3.3.9.8 MCR — Memory Protection Setting Check Result

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

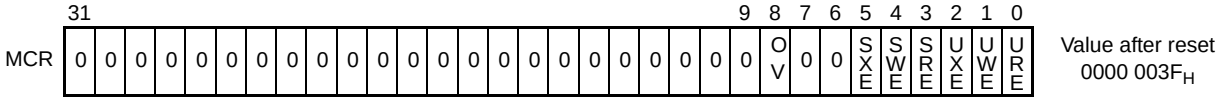


Table 3.46 MCR Register Contents

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	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	0
7, 6	—	(Reserved for future expansion. Be sure to clear to 0.)	R	0
5	SXE	The RH850/F1L does not support this bit. Be sure to set this bit to 1.	R	1
4	SWE	The RH850/F1L does not support this bit. Be sure to set this bit to 1.	R	1
3	SRE	The RH850/F1L does not support this bit. Be sure to set this bit to 1.	R	1
2	UXE	The RH850/F1L does not support this bit. Be sure to set this bit to 1.	R	1
1	UWE	The RH850/F1L does not support this bit. Be sure to set this bit to 1.	R	1
0	URE	The RH850/F1L does not support this bit. Be sure to set this bit to 1.	R	1

CAUTION

There is no memory protection setting check function, but the MCR register can be manipulated.

However, writing to the MCR register is always ignored, and this register always returns a value of 0000 $003F_H$ when read.

3.3.9.9 MPLAn — Protection Area Minimum Address

These registers indicate the minimum address of area n (where n = 0 to 3).

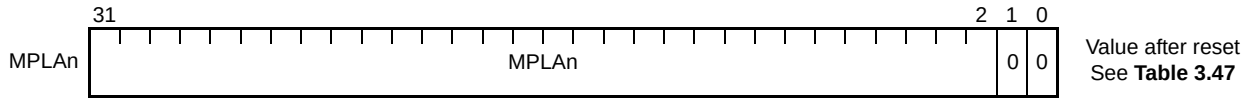


Table 3.47 MPLAn Register Contents

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 these bits to 0.	R	0

CAUTION

For a CPU whose instruction addressing range is 128 MB, a value resulting from a sign-extension of bit 26 is automatically set to bits 31 to 27 of MPLAn.

3.3.9.10 MPUAn — Protection Area Maximum Address

These registers indicate the maximum address of area n (where n = 0 to 3).

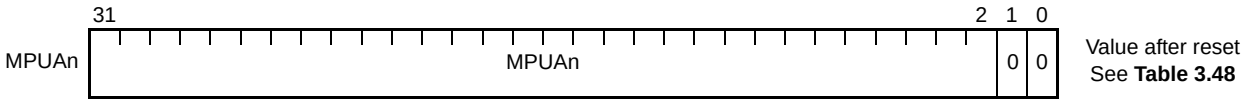


Table 3.48 MPUAn Register Contents

Bit Position	Bit Name	Function	R/W	Value After Reset
31 to 2	MPUA31-2	These bits indicate the maximum 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 these bits to 0.	R	0

CAUTION

For a CPU whose instruction addressing range is 128 MB, a value resulting from a sign-extension of bit 26 is automatically set to bits 31 to 27 of MPUAn.

3.3.9.11 MPATn — Protection Area Attribute

These registers indicate the attributes of area n (where n = 0 to 3).

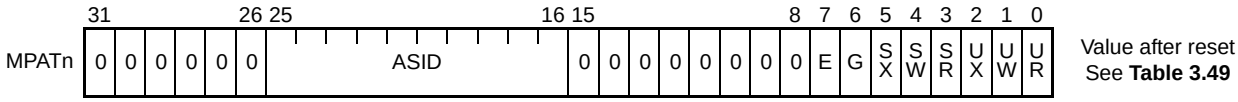


Table 3.49 MPATn Register Contents

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	Not supported by the RH850/F1L. When writing, always write 0.	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: The protection setting for area n is disabled. 1: The protection setting for area n is enabled.	R/W	0
6	G	If this bit is 1, the ASID bit is ignored under the area match condition. Be sure to set this bit to 1.	R/W	Undefined
5	SX	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	Undefined
4	SW	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	Undefined
3	SR	The RH850/F1L does not support this bit. Be sure to clear to 0.	R	Undefined
2	UX	This bit indicates the execution privilege for the user mode. 0: The execution privilege is not granted to the user mode. 1: The execution privilege is granted to the user mode.	R/W	Undefined
1	UW	This bit indicates the write permission for the user mode. 0: The write permission is disabled for the user mode. 1: The write permission is enabled for the user mode.	R/W	Undefined
0	UR	This bit indicates the read permission for the user mode. 0: The read permission is disabled for the user mode. 1: The read permission is enabled for the user mode.	R/W	Undefined

3.3.10 Cache Operation Function Registers

The RH850/F1L does not include a cache operation function, so all the following registers return a value of 0 when read, and writing to these registers is ignored.

Table 3.50 Cache Operation Function Registers

Register No. (regID, selID)	Symbol	Function	Access Permission
SR12, 4	BWERRL	Not implemented. A value of 0 is returned when read and writing is ignored.	SV
SR13, 4	BWERRH		SV
SR14, 4	BRERRL		SV
SR15, 4	BRERRH		SV
SR16, 4	ICTAGL		SV
SR17, 4	ICTAGH		SV
SR18, 4	ICDATL		SV
SR19, 4	ICDATH		SV
SR20, 4	DCTAGL		SV
SR21, 4	DCTAGH		SV
SR22, 4	DCDATL		SV
SR23, 4	DCDATH		SV
SR24, 4	ICCTRL		SV
SR25, 4	DCCTRL		SV
SR26, 4	ICCFG		SV
SR27, 4	DCCFG		SV
SR28, 4	ICERR		SV
SR29, 4	DCERR		SV

3.4 Instruction Reference

The instructions used by the RH850/F1L comply with the V850E3v5-S architecture. For details about the V850E3v5-S architecture, see the *RH850 Family User's Manual: Software (R01US0058E)*.

Although the RH850/F1L complies with the V850E3v5-S architecture, some instructions differ in their specifications. These instructions are shown in the table below. Note that, as described as differences between *V850E3v5* and other architectures in *APPENDIX G*, in the *RH850 Family User's Manual: Software*, there is no RETI instruction.

Table 3.51 Instructions with Specifications that Differ from V850E3v5-S Architecture

Instruction	Operation
CACHE	Functions in the same way as an NOP instruction.
DST	Not available. (A RIE exception occurs.)
EST	Not available. (A RIE exception occurs.)
DBHVTRAP	Functions in the same way as a DBTRAP instruction.
HVCALL	Not available. (A RIE exception occurs.)
HVTRAP	Not available. (A RIE exception occurs.)
LDL.W	Functions in the same way as an LD.W instruction.
LDTC.GR	Not available. (A RIE exception occurs.)
LDTC.VR	Not available. (A RIE exception occurs.)
LDTC.PC	Not available. (A RIE exception occurs.)
LDTC.SR	Not available. (A RIE exception occurs.)
LDVC.SR	Not available. (A RIE exception occurs.)
PREF	Functions in the same way as an NOP instruction.
STC.W	Functions in the same way as an ST.W instruction.
STTC.GR	Not available. (A RIE exception occurs.)
STTC.VR	Not available. (A RIE exception occurs.)
STTC.PC	Not available. (A RIE exception occurs.)
STTC.SR	Not available. (A RIE exception occurs.)
STVC.SR	Not available. (A RIE exception occurs.)
SWICH	The first address in the table when the SWITCH instruction straddles the boundary of a 128 MB space (the next address of the SWITCH instruction) is handled as follows: • If the address of the SWITCH instruction is PC = 03FF FFFE$_H$, the first address in the table is calculated as (PC + 2) = 0400 0000$_H$. • If the address of the SWITCH instruction is PC = 0400 0000$_H$ (physical address = FC00 0000$_H$), the first address in the table is calculated as (PC + 2) = FC00 0002$_H$.
TLBAI	Not available. (A RIE exception occurs.)
TLBR	Not available. (A RIE exception occurs.)
TLBS	Not available. (A RIE exception occurs.)
TLBW	Not available. (A RIE exception occurs.)
TLBVI	Not available. (A RIE exception occurs.)
SYNCE	Functions in the same way as an SYNCM instruction.
Floating-point instructions	Not available. (A UCPOP exception occurs.)
SIMD instructions	Not available. (A UCPOP exception occurs.)

3.4.1 Number of Instruction Clock Cycles

Table 3.52 shows the number of clock cycles used by each instruction.

Table 3.52 Number of Instruction Clock Cycles (1/6)

Type of Instruction		Mnemonic	Operand	Instruction Length (in bytes)	No. of Execution Clock Cycles		
					issue	repeat	latency
Load instructions		LD.B	disp16 [reg1], reg2	4	1	1	2*[1]
			disp23 [reg1], reg3	6	1	1	2*[1]
		LD.BU	disp16 [reg1], reg2	4	1	1	2*[1]
			disp23 [reg1], reg3	6	1	1	2*[1]
		LD.H	disp16 [reg1], reg2	4	1	1	2*[1]
			disp23 [reg1], reg3	6	1	1	2*[1]
		LD.HU	disp16 [reg1], reg2	4	1	1	2*[1]
			disp23 [reg1], reg3	6	1	1	2*[1]
		LD.W	disp16 [reg1], reg2	4	1	1	2*[1]
			disp23 [reg1], reg3	6	1	1	2*[1]
		LD.DW	disp23 [reg1], reg3	6	2	2	4*[1]
	ep-relative	SLD.B	disp7 [ep], reg2	2	1	1	2*[1]
		SLD.BU	disp4 [ep], reg2	2	1	1	2*[1]
		SLD.H	disp8 [ep], reg2	2	1	1	2*[1]
		SLD.HU	disp5 [ep], reg2	2	1	1	2*[1]
		SLD.W	disp8 [ep], reg2	2	1	1	2*[1]
Store instructions		ST.B	reg2, disp16 [reg1]	4	1	1	1
			reg3, disp23 [reg1]	6	1	1	1
		ST.H	reg2, disp16 [reg1]	4	1	1	1
			reg3, disp23 [reg1]	6	1	1	1
		ST.W	reg2, disp16 [reg1]	4	1	1	1
			reg3, disp23 [reg1]	6	1	1	1
		ST.DW	reg3, disp23[reg1]	6	2	2	2
	ep-relative	SST.B	reg2, disp7 [ep]	2	1	1	1
		SST.H	reg2, disp8 [ep]	2	1	1	1
		SST.W	reg2, disp8 [ep]	2	1	1	1
Multiply instructions		MUL	reg1, reg2, reg3	4	1	4	4
			imm9, reg2, reg3	4	1	4	4
		MULH	reg1, reg2	2	1	1	1
			imm5, reg2	2	1	1	1
		MULHI	imm16, reg1, reg2	4	1	1	1
		MULU	reg1, reg2, reg3	4	1	4	4
			imm9, reg2, reg3	4	1	4	4
	Multiply-accumulate instructions	MAC	reg1, reg2, reg3, reg4	4	2	5	5
		MACU	reg1, reg2, reg3, reg4	4	2	5	5

Table 3.52 Number of Instruction Clock Cycles (2/6)

Type of Instruction		Mnemonic	Operand	Instruction Length (in bytes)	No. of Execution Clock Cycles		
					issue	repeat	latency
Arithmetic instructions		ADD	reg1, reg2	2	1	1	1
			imm5, reg2	2	1	1	1
		ADDI	imm16, reg1, reg2	4	1	1	1
		CMP	reg1, reg2	2	1	1	1
			imm5, reg2	2	1	1	1
		MOV	reg1, reg2	2	1	1	1
			imm5, reg2	2	1	1	1
			imm32, reg1	6	1	1	1
		MOVEA	imm16, reg1, reg2	4	1	1	1
		MOVHI	imm16, reg1, reg2	4	1	1	1
		SUB	reg1, reg2	2	1	1	1
		SUBR	reg1, reg2	2	1	1	1
	Conditional arithmetic instructions	ADF	cccc, reg1, reg2, reg3	4	1	1	1
		SBF	cccc, reg1, reg2, reg3	4	1	1	1
	Saturated operation instructions	SATADD	reg1, reg2	2	1	1	1
			imm5, reg2	2	1	1	1
			reg1, reg2, reg3	4	1	1	1
		SATSUB	reg1, reg2	2	1	1	1
			reg1, reg2, reg3	4	1	1	1
		SATSUBI	imm16, reg1, reg2	4	1	1	1
		SATSUBR	reg1, reg2	2	1	1	1
Logical instructions		AND	reg1, reg2	2	1	1	1
		ANDI	imm16, reg1, reg2	4	1	1	1
		NOT	reg1, reg2	2	1	1	1
		OR	reg1, reg2	2	1	1	1
		ORI	imm16, reg1, reg2	4	1	1	1
		TST	reg1, reg2	2	1	1	1
		XOR	reg1, reg2	2	1	1	1
		XORI	imm16, reg1, reg2	4	1	1	1

Table 3.52 Number of Instruction Clock Cycles (3/6)

Type of Instruction		Mnemonic	Operand	Instruction Length (in bytes)	No. of Execution Clock Cycles		
					issue	repeat	latency
Data manipulation instructions		BINS	reg1, pos, width, reg2	4	1	1	1
		BSH	reg2, reg3	4	1	1	1
		BSW	reg2, reg3	4	1	1	1
		CMOV	cccc, reg1, reg2, reg3	4	1	1	1
			cccc, imm5, reg2, reg3	4	1	1	1
		HSH	reg2, reg3	4	1	1	1
		HSW	reg2, reg3	4	1	1	1
		ROTL	imm5, reg2, reg3	4	1	1	1
			reg1, reg2, reg3	4	1	1	1
		SAR	reg1, reg2	4	1	1	1
			imm5, reg2	2	1	1	1
			reg1, reg2, reg3	4	1	1	1
		SASF	cccc, reg2	4	1	1	1
		SETF	cccc, reg2	4	1	1	1
		SHL	reg1, reg2	4	1	1	1
			imm5, reg2	2	1	1	1
			reg1, reg2, reg3	4	1	1	1
		SHR	reg1, reg2	4	1	1	1
			imm5, reg2	2	1	1	1
			reg1, reg2, reg3	4	1	1	1
		SXB	reg1	2	1	1	1
		SXH	reg1	2	1	1	1
		ZXB	reg1	2	1	1	1
		ZXH	reg1	2	1	1	1
Bit search instructions		SCH0L	reg2, reg3	4	1	1	1
		SCH0R	reg2, reg3	4	1	1	1
		SCH1L	reg2, reg3	4	1	1	1
		SCH1R	reg2, reg3	4	1	1	1
Divide instructions		DIV	reg1, reg2, reg3	4	36	36	36
		DIVH	reg1, reg2	2	36	36	36
			reg1, reg2, reg3	4	36	36	36
		DIVHU	reg1, reg2, reg3	4	35	35	35
		DIVU	reg1, reg2, reg3	4	35	35	35
	High-speed divide instructions	DIVQ	reg1, reg2, reg3	4	N + 7*2	N + 7*2	N + 7*2
		DIVQU	reg1, reg2, reg3	4	N + 6*2	N + 6*2	N + 6*2

Table 3.52 Number of Instruction Clock Cycles (4/6)

Type of Instruction	Mnemonic	Operand	Instruction Length (in bytes)	No. of Execution Clock Cycles		
				issue	repeat	latency
Branch instructions	Bcond	disp9 (when condition is not satisfied)	2	1*3	1*3	1*3
		disp9 (when condition is satisfied)	2	2*3	2*3	2*3
	Bcond	disp17 (when condition is not satisfied)	4	1*3	1*3	1*3
		disp17 (when condition is satisfied)	4	2*3	2*3	2*3
	JARL	disp22, reg2	4	2	2	2
		disp32, reg1	6	2	2	2
		[reg1], reg3	4	3	3	3
	JMP	[reg1]	2	3	3	3
		disp32 [reg1]	6	3	3	3
	JR	disp22	4	2	2	2
		disp32	6	2	2	2
Loop instruction	LOOP	reg1, disp16 (After an update, when reg1 = 0)	4	2*3	2*3	2*3
		reg1, disp16 (After an update, when reg1 ≠ 0)	4	5*3	5*3	5*3
Bit manipulation instructions	CLR1	bit#3, disp16 [reg1]	4	3*4	3*4	3*4
		reg2, [reg1]	4	3*4	3*4	3*4
	NOT1	bit#3, disp16 [reg1]	4	3*4	3*4	3*4
		reg2, [reg1]	4	3*4	3*4	3*4
	SET1	bit#3, disp16 [reg1]	4	3*4	3*4	3*4
		reg2, [reg1]	4	3*4	3*4	3*4
	TST1	bit#3, disp16 [reg1]	4	3*4	3*4	3*4
		reg2, [reg1]	4	3*4	3*4	3*4

Table 3.52 Number of Instruction Clock Cycles (5/6)

Type of Instruction	Mnemonic	Operand	Instruction Length (in bytes)	No. of Execution Clock Cycles		
				issue	repeat	latency
Special instructions						
Jump with table look up	SWITCH	reg1	2	5	5	5
Subroutine call	CALLT	imm6	2	5	5	5
	CTRET	—	4	4	4	4
System call exception	SYSCALL	vector8	4	6	6	6
Software exception	FETRAP	vector	2	4	4	4
	TRAP	vector5	4	4	4	4
Return from exception	EIRET	—	4	4	4	4
	FERET	—	4	4	4	4
EI-level interrupt control	DI	—	4	1	1	1
	EI	—	4	1	1	1
Saving/restoring stack frames	DISPOSE	imm5, list12	4	n + 1*5	n + 1*5	n + 1*5
		imm5, list12, [reg1]	4	n + 3*5	n + 3*5	n + 3*5
	PREPARE	list12, imm5	4	n + 1*5	n + 1*5	n + 1*5
		list12, imm5, sp	4	n + 2*5	n + 2*5	n + 2*5
		list12, imm5, imm16	6	n + 2*5	n + 2*5	n + 2*5
		list12, imm5, imm16<<16	6	n + 2*5	n + 2*5	n + 2*5
		list12, imm5, imm32	8	n + 2*5	n + 2*5	n + 2*5
	POPSP	rh-rt	4	n + 1*6	n + 1*6	n + 1*6
	PUSHSP	rh-rt	4	n + 1*6	n + 1*6	n + 1*6
System register manipulation	LDSR	reg2, regID, selID	4	1	1	1
	STSR	regID, reg2, selID	4	1	1	1
Exclusion control	CAXI	[reg1], reg2, reg3	4	5*4	5*4	5*4
	LDL.W*7	LDL.W [reg1], reg3	4	1	1	2*2
	STC.W*7	reg3, [reg1]	4	1	1	1
Stop	HALT	—	4	Undefined	Undefined	Undefined
	SNOOZE	—	4	Undefined	Undefined	Undefined
Synchronization	SYNCE	—	2	Undefined	Undefined	Undefined
	SYNCI	—	2	Undefined	Undefined	Undefined
	SYNCM	—	2	Undefined	Undefined	Undefined
	SYNCP	—	2	Undefined	Undefined	Undefined
Other	NOP	—	2	1	1	1
	RIE	—	4	4	4	4
Cache instruction	CACHE*8	cacheop, [reg1]	4	1	1	1
Prefetch instruction	PREF*8	prefop, [reg1]	4	1	1	1
Debug instruction	DBCP	—	4	1	1	1
	DBHVTRAP *9	—	4	Undefined	Undefined	Undefined
	DBPUSH	rh-rt	4	n*10	n*10	n*10
	DBTAG	imm10	4	1	1	1

Table 3.52 Number of Instruction Clock Cycles (6/6)

Type of Instruction	Mnemonic	Operand	Instruction Length (in bytes)	No. of Execution Clock Cycles		
				issue	repeat	latency
Debug function instructions	DBRET	—	4	Undefined	Undefined	Undefined
	DBTRAP	—	4	Undefined	Undefined	Undefined
	RMTRAP	—	4	Undefined	Undefined	Undefined

Note 1. When the RAM area is accessed with no wait states inserted in the clock cycle: 2 + Number of read access wait states
When an area other than the RAM area is accessed: 3 + Number of read access wait states

Note 2. N = (Effective number of bits of absolute value of dividend) – (Effective number of bits of absolute value of divisor)
However, if N < 1 or the value is divided by 0, this value is N = 1. N is in a range of 1 to 31.

Note 3. If the immediately previous instruction was an instruction that rewrote the contents of the PSW, this is the number of execution clock cycles + 1.

Note 4. When the RAM area is accessed with no wait states inserted in the clock cycle: 3 + Number of read access wait states
When an area other than the RAM area is accessed: 4 + Number of read access wait states
For the CAXI instruction, this is the number of execution clock cycles required for a bit manipulation instruction + 2.

Note 5. *n* depends on the total number of registers specified by *list*.
The value when no wait states are inserted in the clock cycle is as follows:
- PREPARE minimum value: 0; maximum value: 12
- DISPOSE minimum value: 1; maximum value: 12 (+2 clock cycles if DISPOSE is executed with JMP)
When n = 0, this is 1 clock cycle (3 clock cycles if DISPOSE is executed with JMP).

Note 6. *n* depends on the total number of registers specified by *rh-rt*.
The value when no wait states are inserted in the clock cycle is as follows:
- PUSHSP n minimum value: 0; maximum value: 32
- POPSP n minimum value: 0; maximum value: 32
When n = 0, this is 1 clock cycle.

Note 7. Exclusive control of LDL.W and STC.W is not supported. LDL.W functions as the LD.W instruction, and STC.W functions as the ST.W instruction.

Note 8. Functions as a NOP instruction.

Note 9. Functions as the DBTRAP instruction.

Note 10. *n* depends on the total number of registers specified by *rh-rt*.
- DBPUSH n minimum value: 0; maximum value: 32
When the number of registers specified by *rh-rt* ≤ 0, this is 1 clock cycle.

3.5 System Error Notification Setting Registers

3.5.1 SEG_CONT — SYSERR Exception Notification Setting Registers

SEG_CONT is an enable register to set whether to enable or disable error notification of each SYSERR source.

When bit 7, 6, 4, or 1 is set to 1, the corresponding error results in a SYSERR exception. If the value of the corresponding enable bit is 0, the error flag is set when the error occurs but a SYSERR exception is not produced.

Be sure to set bits 15 to 8, 5, 3, 2, and 0 to 0.

Access: SEG_CONT can be read and written in 16-bit units.
SEG_CONTL can be read and written in 8/1-bit units.

Address: SEG_CONT: FFFF 8C00$_H$
SEG_CONTL: FFFF 8C00$_H$

Value after reset: 0000$_H$ (Initialized by any reset source.)

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

Table 3.53 SEG_CONT Register Contents

Bit Position	Bit Name	Function
15 to 8	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
7	SEG_CONT DMAE	DMA transfer error notification enable Sets whether to enable SYSERR notification at a DMA transfer error occurrence.
6	SEG_CONT RMWE	Read modify write error notification enable Sets whether to enable SYSERR notification for byte write or half-word write to the primary local RAM, secondary local RAM, or retention RAM.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4	SEG_CONT RAME	Primary local RAM/secondary local RAM/retention RAM area error notification enable Sets whether to enable SYSERR notification at an error occurring when the CPU accessed data in the primary local RAM, secondary local RAM, or retention RAM area.
3, 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	SEG_CONT ROME	Code flash error notification enable Sets whether to enable SYSERR notification at an error occurring when the CPU accessed data in the code flash area.
0	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.

3.5.2 SEG_FLAG — System Error Source Storing Registers

SEG_FLAG is the flag register to store occurrence state of each SYSERR source. When a SYSERR source occurs, the flag of the corresponding error source is set to 1. Each flag is cleared to 0 by reading the flag state of 1 and then writing 0.

Even if 0 is written to the flag (read as 0) in the subsequent write operation, a flag is set for the error occurring during the time period between reading and writing.

Be sure to set bits 15 to 8, 5, 3, 2, and 0 to 0.

Access: SEG_FLAG can be read and written in 16-bit units.
SEG_FLAGL can be read and written in 8/1-bit units.

Address: SEG_FLAG: FFFF 8C02$_H$
SEG_FLAGL: FFFF 8C02$_H$

Value after reset: 0000$_H$ (Initialized by any reset source.)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2*1	1	0
	—	—	—	—	—	—	—	—	SEG_FLAG DMAF	SEG_FLAG RMWF	—	SEG_FLAG RAMF	—	—	SEG_FLAG ROMF	—
Value after reset	0	0	0	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	R/W	R	R	R/W	R

Note 1. When read, bit 2 in this register returns an undefined value.

Table 3.54 SEG_FLAG Register Contents

Bit Position	Bit Name	Function
15 to 8	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
7	SEG_FLAG DMAF	DMA transfer error flag DMAF is set to 1 when a DMA transfer error is detected during DMA access.
6	SEG_FLAG RMWF	Read modify write error flag RMWF is set to 1 when an error is detected during byte write or half-word write to the primary local RAM, secondary local RAM, or retention RAM with the read modify write error.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4	SEG_FLAG RAMF	Primary local RAM/secondary local RAM/retention RAM area error flag RAMF is set to 1 on detection of an error when the CPU accessed data in the primary local RAM, secondary local RAM, or retention RAM area.
3, 2	Reserved	When writing to these bits, write the value after reset.
3	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
2	Reserved	When read, the undefined value is returned. When writing to this bit, write the value after reset.
1	SEG_FLAG ROMF	Code flash error flag ROMF is set to 1on detection of an error when the CPU accessed data in the code flash area.
0	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.

3.6 Usage Notes

3.6.1 Completion of Store Instruction and Synchronization of Subsequent Instruction

When a control register is updated by the store instruction, there is a time lapse 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 reflected to the instruction following the store instruction, the appropriate synchronization is required. The following shows the execution process to perform the synchronization.

(1) When the updated content of the control register is to be reflected to the execution of the subsequent instruction

Follow the appropriate procedure below when updating of a control register may make a difference to the order of program execution or the correct operation of a subsequent instruction relies on the new value.

(a) Store instructions for writing to peripheral function registers and instructions for access to DMA or INTC related registers

When an instruction for access to a DMA- or INTC-related register (B) follows a store instruction for a peripheral function register (A), (B) may proceed before updating of the peripheral function register by (A) is actually completed. When the correct operation of a subsequent instruction relies on the new value of the control register, the following sequence is required.

- Store instruction for the peripheral function register (A)
- Dummy reading of the above register
- Instruction for access to the DMA- or INTC-related register (B)

(b) Store instructions for writing to DMA-related register or peripheral function register and EI/DI instructions or STSR Instructions of ISPR/ICSR

When an instruction for access to an EI/DI instruction or STSR Instruction of ISPR/ICSR (B) follows a store instruction for a DMA-related register or peripheral function register (A), (B) may proceed before updating of the DMA-related register or peripheral function register by (A) is actually completed. When the correct operation of a subsequent instruction relies on the new value of the control register, the following sequence is required.

- Store instruction for the DMA-related register or peripheral function register (A)
- Dummy reading of the above register
- EI/DI instruction or STSR instruction of ISPR/ICSR (B)

(c) Store instructions for writing to INTC-related registers and EI/DI instructions or STSR Instructions of ISPR/ICSR

When an instruction for access to an EI/DI instruction or STSR Instruction of ISPR/ICSR (B) follows a store instruction for an INTC-related register (A), (B) may proceed before updating of the INTC-related register by (A) is actually completed. When the correct operation of a subsequent instruction relies on the new value of the control register, the following sequence is required.

- Store instruction for the INTC-related register (A)

- Dummy reading of the above register
- SYNCP
- EI/DI instruction or STSR instruction of ISPR/ICSR (B)

(2) When the updated content of the control register or memory is to be reflected to the instruction fetch of the subsequent instruction

For example, when the instruction is to be executed, after it is written to the memory. In this case, execute the following process:

- Store instruction to update the memory (such as ST.W)
- Dummy read of the above memory (such as LD.W)
- SYNCI
- Subsequent instruction (such as a branch instruction)

3.6.2 System Register Hazards

To solve 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, ASID, MPU: All related registers (Register number: sr*, 5-7)
- 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 prohibited state (PSW.ID = 1).
  - PSW.EBV, EBASE, INTBP, ISPR, PMR, ICSR, INTCFG

3.6.3 Access to Registers by Bit-Manipulation Instructions

Processing of writing by bit-manipulation instructions consists of 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.

3.6.4 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 *RH850 Family User's Manual: Software*.

Section 4 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.

4.1 Overview

4.1.1 Functional Overview

The RH850/F1L products require a special procedure using write-protected registers for setting important registers that affect the system, such as clock, reset, and port-related registers. The settings of protected registers are protected against illegal writing by programs by requiring a special procedure. For details about the protected registers, see **Section 4.1.5, Write-Protection Target Registers**. Write-protected registers are managed in units of protected registers called register protection clusters.

4.1.2 Write-Protected Register Write Procedure

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.

In case of any access to another register (second register) between step 1 to step 4 of the above sequence for writing to a write-protected register (first register), the protection mechanism behaves as follows:

- If the second register belongs to the same cluster, the write to the protected register fails (indicated by the error monitor bit 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 can be completed successfully.

4.1.3 Interrupt during Write Protection Unlocking

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 return from the interrupt service routine. **Figure 4.1** shows an execution example.

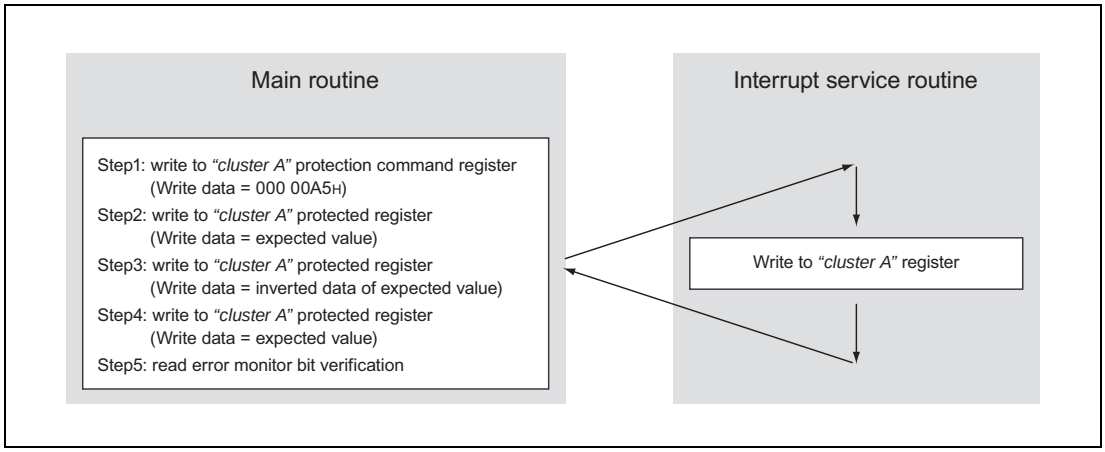


Figure 4.1 Example of Interruption of Register Protection Unlock Sequence

(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 interrupted the write operation to the protected register is completed upon return from the interrupt service routine. **Figure 4.2** shows an execution example.

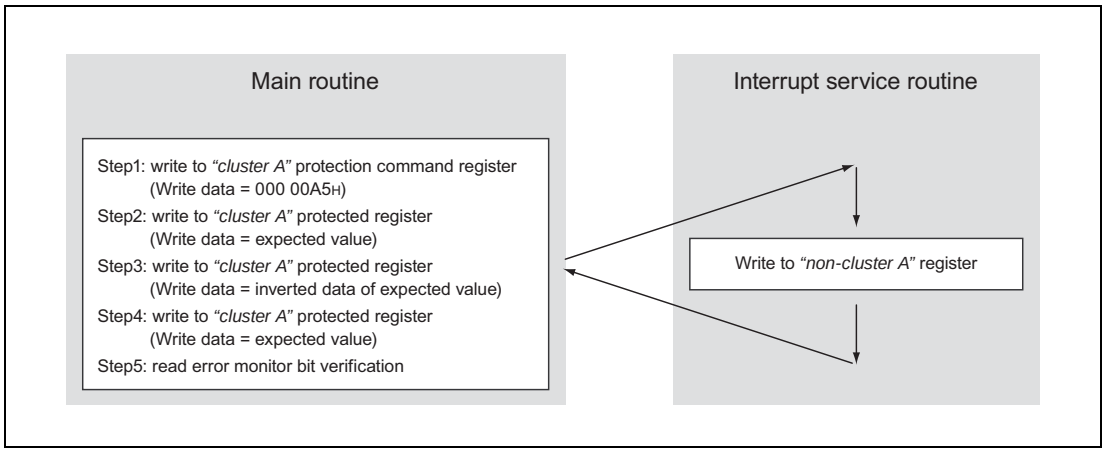


Figure 4.2 Example of Successful Protection Unlock Sequence

For more information on registers of RH850/F1L register protection clusters, see **Section 4.1.5, Write-Protection Target Registers**.

4.1.4 Emulation Break during Write Protection Unlock Sequence

If an emulation break occurs during the above 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.

4.1.5 Write-Protection Target Registers

The registers that are protected through the write-protected registers are listed below.

Table 4.1 Write-Protected Registers (1/2)

Protection Target	Protected Register	Protection 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	PLLE	PROTCMD1	PROTS1	Control protection cluster 1
	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			
Clock Monitors	CLMAnCTL0	CLMAnPCMD	CLMAnPS	Clock Monitor control protection cluster
	CLMATEST	PROTCMDCLMA	PROTSCLMA	Clock Monitor test protection cluster
Port*[1]	PODCn, JPODC0, PDSCn	PPCMDn	PPROTSn	Port protection cluster 1 and 2

Table 4.1 Write-Protected Registers (2/2)

Protection Target	Protected Register	Protection Register		Protection Cluster
		Command Register	Status Register	
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.

(1) Port Protection Clusters

The following port control registers have write protection:

- 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 4.2 Port Protection Clusters

Port Protection Cluster	Port Group
0	JP0, P0, P1, P2, P8
1	P9, P10, P11, P12, P18, 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 PODC2.

4.2 Registers

4.2.1 List of Registers

The following table lists the write-protected registers.

Table 4.3 List of Write-Protected 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 Monitors 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$_H$
	PPCMD8	FFC1 4C20$_H$
Protection status registers	JPPROTS0	FFC2 04B0$_H$
	PPROTS0	FFC1 4B00$_H$
	PPROTS1	FFC1 4B04$_H$
	PPROTS2	FFC1 4B08$_H$
	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$_H$
	PPCMD18	FFC1 4C48$_H$
	PPCMD20	FFC1 4C50$_H$

Table 4.3 List of Write-Protected 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$_H$
	PPROTS18	FFC1 4B48$_H$
	PPROTS20	FFC1 4B50$_H$
Core Voltage Monitor protection cluster		
Protection command register	PROTCMDCVM	FFF5 0100$_H$
Protection status register	PROTSCVM	FFF5 0104$_H$
Self-programming protection cluster		
Protection command register	FLMDPCMD	FFA0 0004$_H$
Protection error status register	FLMDPS	FFA0 0008$_H$

4.2.2 Details of Control Protection Cluster Registers

4.2.2.1 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 4.1, Write-Protected Registers**.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 4.3, List of Write-Protected 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 4.4 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 control protection cluster registers

4.2.2.2 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 4.1, Write-Protected Registers**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 4.3, List of Write-Protected 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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 4.5 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 1: Protection error occurred

4.2.3 Details of Clock Monitor Protection Cluster Registers

4.2.3.1 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 4.1, Write-Protected Registers**.

Access: This register is a write-only register that can be written in 8-bit units.

Address: See **Table 4.3, List of Write-Protected 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 4.6 CLMAnPCMD Register Contents

Bit Position	Bit Name	Function
7 to 0	CLMAnREG[7:0]	Protection command register bits that enable writing to clock monitor control protection cluster registers

4.2.3.2 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 4.1, Write-Protected Registers**.

Access: This register is a read-only register that can be read in 8-bit units.

Address: See **Table 4.3, List of Write-Protected 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 4.7 CLMAnPS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	CLMAnPRERR	Protection write sequence error monitor 0: No protection error occurred 1: Protection error occurred

4.2.3.3 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 4.3, List of Write-Protected 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 4.8 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]	This register is a protection command register for the CLMATEST register.

4.2.3.4 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 4.3, List of Write-Protected 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 4.9 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

4.2.4 Details of Core Voltage Monitor Register

4.2.4.1 PROTCMDCVM — Core Voltage Monitor Protection Command Register

This register is a protection command register for the CVMF register.

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

Address: See **Table 4.3, List of Write-Protected 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 4.10 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]	This register is a protection command register for the CVMF register.

4.2.4.2 PROTSCVM — Core Voltage Monitor Protection Status Register

This register is used to verify whether the write-protected register (CVMF) has been successfully written or not.

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

Address: See **Table 4.3, List of Write-Protected 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 4.11 PROTSCVM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	CVMFPRERR	Writing sequence protection error monitor 0: No protection error occurred 1: Protection error occurred

4.2.5 Details of Port Protection Cluster Registers

4.2.5.1 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 4.1, Write-Protected Registers**.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See **Table 4.3, List of Write-Protected 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 4.12 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

4.2.5.2 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 4.1, Write-Protected Registers**.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See **Table 4.3, List of Write-Protected 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 4.13 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

4.2.6 Details of Self-Programming Protection Cluster Registers

4.2.6.1 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 4.3, List of Write-Protected 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 4.14 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 self-programming protection cluster registers

4.2.6.2 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 4.3, List of Write-Protected 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 4.15 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 5 Operating Modes

This section describes the operating mode and mode selection of the RH850/F1L.

The RH850/F1L has the operating modes as shown below.

- Normal operating mode

  This mode is for execution of the user programming. The on-chip debug capabilities also use this mode. If FLMD0 is pulled up high while the operation is 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 reset is generated, the RH850/F1L latches the state of the FLMD0, FLMD1, MODE0, and MODE1 pins and decides the operating mode after the reset is canceled. **Table 5.1** lists the relationship between the pin settings and the operating mode.

Table 5.1 Selection of Operating Modes

Pins				
FLMD0	**FLMD1 (P10_8)**	**MODE0 (P10_1)**	**MODE1 (P10_2)**	**Operating Mode**
0	x	x	x	Normal operation mode
1	0	x	x	Serial programming mode
1	1	0	1	Boundary scan mode
Other than above				Settings are prohibited

Section 6 Exception/Interrupts

6.1 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 *RH850 Family Users' Manual: Software*.

Table 6.1 List of Exception Sources

Name	Symbol	Source	Priority	Saved to
Reset	RESET	Reset input	High	—
Debug exception (asynchronous)	AsyncDB (NM)	Asynchronous debug event		DB
FE level non-maskable interrupt*1	FENMI (NM)	FENMI input		FE
System error exception	SYSERR (NM)	SYSERR input		FE
FE level maskable interrupt*1	FEINT (NM)	FEINT input		FE
EI level maskable interrupt*1	EIINT	Interrupt controller		EI
Debug exception (synchronous)	SyncDB	Synchronous debug event		DB
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 reserved instruction		FE
Misalign exception	MAE	Generation of misalign access		FE
Memory protection exception (access right)	MDP	Memory protection violation		FE
Debug trap	DBTRAP	Execution of DBTRAP instruction		DB
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 interrupts are described in this section.

(1) Interrupts

The following three exceptions from **Table 6.1** are called interrupts, and are thus 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.
  - FENMI interrupt is acknowledged even if the CPU system register PSW.NP = 1.
  - Return from FENMI interrupt is not possible, recover is disabled.
- 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 enabled, recovery enabled
- 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, INT with higher priority is being processed, or PSW.ID = 1.
  - Return enabled, recovery enabled.
  - Interrupt masking can be specified for each interrupt channel.
  - 8 interrupt priority levels can be specified for each interrupt channel
  - In this section, the INT that corresponds to interrupt channel n is indicated by “INTn”, whereas the INT that corresponds to interrupt source xxx is indicated by “INTxxx”.

  For the PSW register, see **Section 3.3.2.5, PSW — Program Status Word** and *RH850 Family Users’ Manual: Software*.

NOTE

Return: Indicates whether or not the program can resume 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 that was when the program was interrupted.

These interrupt sources are described in the following pages.

6.2 RH850/F1L Interrupt Sources

6.2.1 RH850/F1L Interrupt Sources

6.2.1.1 RH850/F1L FE Level Non-Maskable Interrupts

(1) Priority Group

See **Table 6.1, List of Exception Sources**.

(2) Return PC

An FE non-maskable interrupt does not allow to return or recover.

(3) Status Register

See **Section 6.4.3, FNC — FE Level NMI Status Register**.

(4) Return Instruction

None

Table 6.2 FE Level Non-Maskable Interrupt Requests

Interrupt			Interrupt Request					
	Control Register							
Symbol	Name	Address	Name	Cause	Unit	Priority Group	Exception Request Code	Handler Address 00000...
FENMI	FNC	FFFF 9078$_H$	TNMI	NMI pin	Port	*1	E0$_H$	0E0$_H$
			WDTA0NMI	WDTA0 FENMI interrupt request	WDTA0			
			WDTA1NMI	WDTA1 FENMI interrupt request	WDTA1			

Note 1. See **Table 6.1, List of Exception Sources**.

The source of the FENMI interrupt can be evaluated by a dedicated flag register. See **Section 6.2.2, FE Level Non-Maskable Interrupt Sources** for details.

6.2.1.2 RH850/F1L FE Level Maskable Interrupts

(1) Priority Group

See **Table 6.1, List of Exception Sources**.

(2) Return PC

The PC return from an interrupt handling routine by the FERET instruction is the suspended PC (current PC).

(3) Status Register

See **Section 6.4.4, FIC — FE Level Maskable Interrupt Status Register**.

(4) Return Instruction

FERET

Table 6.3 FE Level Maskable Interrupt Requests

Interrupt			Interrupt Request					
	Control Register							
Symbol	Name	Address	Name	Cause	Unit	Priority Group	Exception Request Code	Handler Address 0000...
FEINT	FIC	FFFF 907A$_H$	INTLVI	LVI voltage detection (falling)	LVI	*1	F0$_H$	0F0$_H$
			INTECCDEEP0	ECC 2-bit error interrupt of Data Flash	Data Flash			
			INTECCDCNRAM	ECC 2-bit error interrupt of RSCAN0	RSCAN0			
			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			
			INTECCSDFLI0	ECC 1-bit error interrupt of Code Flash	Code Flash			
			INTECCRAM	ECC 1-bit error interrupt of RAM	RAM			
			INTOSTM0_FE	OSTM0 interrupt	OSTM0			
			INTLVIH	LVI voltage detection (rising)	LVI			

Note 1. See **Table 6.1, List of Exception Sources**.

6.2.1.3 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 name for **IC**<*name*>, see **Table 6.4**.

- Interrupt request name: **INT<*name*>**
  The prefix **"INT"** is put in front of **<*name*>**.
- Interrupt request control register: **IC<*name*>**
  The prefix **"IC"** is put in front of **<*name*>**.
  The 16-bit register **IC<*name*>** can also be accessed byte-wise with the following names:
  - 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*>**, **P2<*name*>**, **P1<*name*>**, **P0<*name*>**
  The bit prefix "**CT**", "**RF**", "**MK**", "**TB**", "**P2**", "**P1**", or "**P0**" prepends 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*, **P3***TAUD0I2*, **P2***TAUD0I2*, **P1***TAUD0I2*, **P0***TAUD0I2*

(2) Priority Group

See **Table 6.1, List of Exception Sources**.

(3) Return PC

The PC return from an interrupt handling routine by the EIRET instruction is the suspended PC (current PC).

(4) Control Register

EI level maskable interrupt control register

See **Section 6.4.1, ICxxx — EI Level Interrupt Control Registers**.

(5) Return Instruction

EIRET instruction

(6) Configuration

EI level maskable interrupts are supported on total 288 channels with a cascade connection of INTC1 and INTC2.

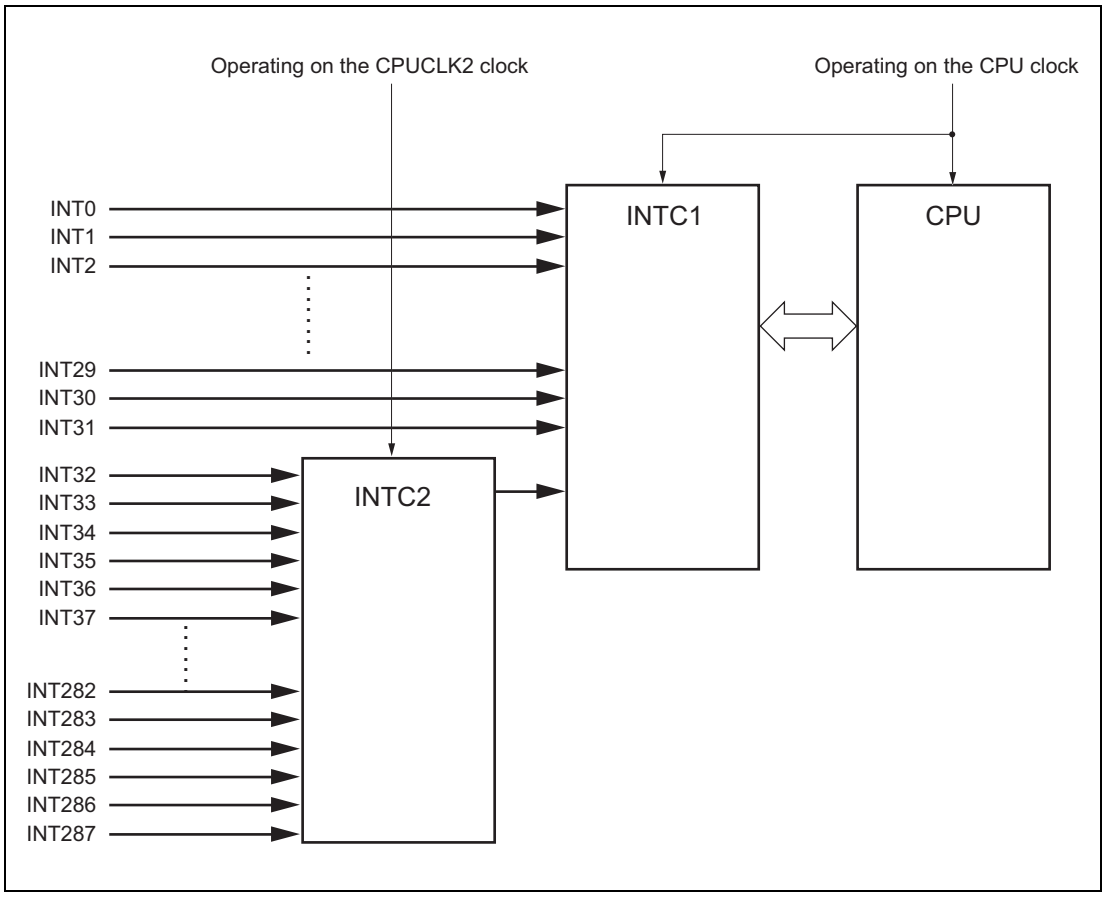


Figure 6.1 Configuration Diagram of EI Level Maskable Interrupt

CAUTION

As CPUCLK2 is the operating clock for INTC2, INT32 to INT287 interrupts, which are connected to INTC2, are delayed more than the interrupts directly connected to INITC1.

Table 6.4 lists EI level maskable interrupts.

Table 6.4 RH850/F1L EI Level Maskable Interrupt Sources (1/8)

Interrupt			Interrupt Request				Exception Request	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset)*8		
	Control Register													Direct Jumping to an Address		
Interrupt No.*1	Name	Address	Name	Source	Unit	Detection Type*2								RINT = 0*3	RINT = 1*4	Reference to a Table*5
0	ICTAUD0I0	FFFF 9000$_H$	INTTAUD0I0*6	Interrupt for CH0 of TAUD0	TAUD0	Edge	1000$_H$	√	√	√	√	√	√	*11	*12	+000$_H$
	ICCSIH2IC_1	FFFF 9000$_H$	INTCSIH2IC_1*6	CSIH2 communication status interrupt	CSIH2	Edge		—	—							
1	ICTAUD0I2	FFFF 9002$_H$	INTTAUD0I2*6	Interrupt for CH2 of TAUD0	TAUD0	Edge	1001$_H$	√	√	√	√	√	√			+004$_H$
	ICCSIH3IC_1	FFFF 9002$_H$	INTCSIH3IC_1*6	CSIH3 communication status interrupt	CSIH3	Edge		—	—	—						
2	ICTAUD0I4	FFFF 9004$_H$	INTTAUD0I4	Interrupt for CH4 of TAUD0	TAUD0	Edge	1002$_H$	√	√	√	√	√	√			+008$_H$
3	ICTAUD0I6	FFFF 9006$_H$	INTTAUD0I6	Interrupt for CH6 of TAUD0	TAUD0	Edge	1003$_H$	√	√	√	√	√	√			+00C$_H$
4	ICTAUD0I8	FFFF 9008$_H$	INTTAUD0I8	Interrupt for CH8 of TAUD0	TAUD0	Edge	1004$_H$	√	√	√	√	√	√			+010$_H$
5	ICTAUD0I10	FFFF 900A$_H$	INTTAUD0I10*6	Interrupt for CH10 of TAUD0	TAUD0	Edge	1005$_H$	√	√	√	√	√	√			+014$_H$
	ICCSIH3IR_1	FFFF 900A$_H$	INTCSIH3IR_1*6	CSIH3 receive status interrupt	CSIH3	Edge		—	—	—						
6	ICTAUD0I12	FFFF 900C$_H$	INTTAUD0I12*6	Interrupt for CH12 of TAUD0	TAUD0	Edge	1006$_H$	√	√	√	√	√	√			+018$_H$
	ICCSIH3IRE_1	FFFF 900C$_H$	INTCSIH3IRE_1*6	CSIH3 communication error interrupt	CSIH3	Edge		—	—	—						
7	ICTAUD0I14	FFFF 900E$_H$	INTTAUD0I14*6	Interrupt for CH14 of TAUD0	TAUD0	Edge	1007$_H$	√	√	√	√	√	√			+01C$_H$
	ICCSIH3IJC_1	FFFF 900E$_H$	INTCSIH3IJC_1*6	CSIH3 job completion interrupt	CSIH3	Edge		—	—	—						
8	ICTAPA0IPEK0	FFFF 9010$_H$	INTTAPA0IPEK0*6	TAPA0 peak interrupt 0	TAPA0	Edge	1008$_H$	√	√	√	√	√	√			+020$_H$
	ICCSIH1IC_1	FFFF 9010$_H$	INTCSIH1IC_1*6	CSIH1 communication status interrupt	CSIH1	Edge		—	—							
9	ICTAPA0IVLY0	FFFF 9012$_H$	INTTAPA0IVLY0*6	TAPA0 valley interrupt 0	TAPA0	Edge	1009$_H$	√	√	√	√	√	√			+024$_H$
	ICCSIH1IR_1	FFFF 9012$_H$	INTCSIH1IR_1*6	CSIH1 receive status interrupt	CSIH1	Edge		—	—							
10	ICADCA0I0	FFFF 9014$_H$	INTADCA0I0	ADCA0 SG1 end interrupt	ADCA0	Edge	100A$_H$	√	√	√	√	√	√			+028$_H$
11	ICADCA0I1	FFFF 9016$_H$	INTADCA0I1	ADCA0 SG2 end interrupt	ADCA0	Edge	100B$_H$	√	√	√	√	√	√			+02C$_H$
12	ICADCA0I2	FFFF 9018$_H$	INTADCA0I2	ADCA0 SG3 end interrupt	ADCA0	Edge	100C$_H$	√	√	√	√	√	√			+030$_H$
13	ICDCUTDI	FFFF 901A$_H$	INTDCUTDI	Dedicated interrupt for on-chip debug function	Port	Edge	100D$_H$	√	√	√	√	√	√			+034$_H$
14	ICRCANGERR	FFFF 901C$_H$	INTRCANGERR	CAN global error interrupt	RSCAN0	Level	100E$_H$	√	√	√	√	√	√			+038$_H$
15	ICRCANGRECC	FFFF 901E$_H$	INTRCANGRECC	CAN receive FIFO interrupt	RSCAN0	Level	100F$_H$	√	√	√	√	√	√			+03C$_H$
16	ICRCAN0ERR	FFFF 9020$_H$	INTRCAN0ERR	CAN0 error interrupt	RSCAN0	Level	1010$_H$	√	√	√	√	√	√			+040$_H$
17	ICRCAN0REC	FFFF 9022$_H$	INTRCAN0REC	CAN0 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	1011$_H$	√	√	√	√	√	√			+044$_H$
18	ICRCAN0TRX	FFFF 9024$_H$	INTRCAN0TRX	CAN0 transmit interrupt	RSCAN0	Level	1012$_H$	√	√	√	√	√	√			+048$_H$
19	ICCSIG0IC	FFFF 9026$_H$	INTCSIG0IC*6	CSIG0 communication status interrupt	CSIG0	Edge	1013$_H$	√	√	√	√	√	√			+04C$_H$
	ICCSIH1IRE_1	FFFF 9026$_H$	INTCSIH1IRE_1*6	CSIH1 communication error interrupt	CSIH1	Edge		—	—							
20	ICCSIG0IR	FFFF 9028$_H$	INTCSIG0IR*6	CSIG0 receive status interrupt	CSIG0	Edge	1014$_H$	√	√	√	√	√	√			+050$_H$
	ICCSIH1IJC_1	FFFF 9028$_H$	INTCSIH1IJC_1*6	CSIH1 job complete interrupt	CSIH1	Edge		—	—							
21	ICCSIH0IC	FFFF 902A$_H$	INTCSIH0IC	CSIH0 communication status interrupt	CSIH0	Edge	1015$_H$	√	√	√	√	√	√			+054$_H$
22	ICCSIH0IR	FFFF 902C$_H$	INTCSIH0IR	CSIH0 receive status interrupt	CSIH0	Edge	1016$_H$	√	√	√	√	√	√			+058$_H$
23	ICCSIH0IRE	FFFF 902E$_H$	INTCSIH0IRE	CSIH0 communication error interrupt	CSIH0	Edge	1017$_H$	√	√	√	√	√	√			+05C$_H$
24	ICCSIH0IJC	FFFF 9030$_H$	INTCSIH0IJC	CSIH0 job completion interrupt	CSIH0	Edge	1018$_H$	√	√	√	√	√	√			+060$_H$
25	ICRLIN30	FFFF 9032$_H$	INTRLIN30	RLIN30 interrupt	RLIN30	Edge	1019$_H$	√	√	√	√	√	√			+064$_H$
26	ICRLIN30UR0	FFFF 9034$_H$	INTRLIN30UR0	RLIN30 transmit interrupt	RLIN30	Edge	101A$_H$	√	√	√	√	√	√			+068$_H$
27	ICRLIN30UR1	FFFF 9036$_H$	INTRLIN30UR1	RLIN30 reception complete interrupt	RLIN30	Edge	101B$_H$	√	√	√	√	√	√			+06C$_H$
28	ICRLIN30UR2	FFFF 9038$_H$	INTRLIN30UR2	RLIN30 status interrupt	RLIN30	Edge	101C$_H$	√	√	√	√	√	√			+070$_H$
29	ICP0	FFFF 903A$_H$	INTP0*6	External interrupt	Port	Edge	101D$_H$	√	√	√	√	√	√			+074$_H$
	ICCSIH2IR_1	FFFF 903A$_H$	INTCSIH2IR_1*6	CSIH2 reception status interrupt	CSIH2	Edge		—	—							

Table 6.4 RH850/F1L EI Level Maskable Interrupt Sources (2/8)

Interrupt			Interrupt Request				Exception Request	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset)*8		
	Control Register													Direct Jumping to an Address		
Interrupt No.*1	Name	Address	Name	Source	Unit	Detection Type*2								RINT = 0*3	RINT = 1*4	Reference to a Table*5
30	ICP1	FFFF 903C$_H$	INTP1*6	External interrupt	Port	Edge	101E$_H$	√	√	√	√	√	√	*11	*12	+078$_H$
	ICCSIH2IRE_1	FFFF 903C$_H$	INTCSIH2IRE_1*6	CSIH2 communication error interrupt	CSIH2	Edge		—	—							
31	ICP2	FFFF 903E$_H$	INTP2*6	External interrupt	Port	Edge	101F$_H$	√	√	√	√	√	√			+07C$_H$
	ICCSIH2IJC_1	FFFF 903E$_H$	INTCSIH2IJC_1*6	CSIH2 job completion interrupt	CSIH2	Edge		—	—							
32	ICWDTA0	FFFF A040$_H$	INTWDTA0	WDTA0 interrupt	WDTA0	Edge	1020$_H$	√	√	√	√	√	√			+080$_H$
33	ICWDTA1	FFFF A042$_H$	INTWDTA1	WDTA1 interrupt	WDTA1	Edge	1021$_H$	√	√	√	√	√	√			+084$_H$
34	ICP3	FFFF A044$_H$	INTP3	External interrupt	Port	Edge	1022$_H$	√	√	√	√	√	√			+088$_H$
35	ICP4	FFFF A046$_H$	INTP4	External interrupt	Port	Edge	1023$_H$	√	√	√	√	√	√			+08C$_H$
36	ICP5	FFFF A048$_H$	INTP5	External interrupt	Port	Edge	1024$_H$	√	√	√	√	√	√			+090$_H$
37	ICP10	FFFF A04A$_H$	INTP10	External interrupt	Port	Edge	1025$_H$	√	√	√	√	√	√			+094$_H$
38	ICP11	FFFF A04C$_H$	INTP11	External interrupt	Port	Edge	1026$_H$	√	√	√	√	√	√			+098$_H$
39	ICTAUD0I1	FFFF A04E$_H$	INTTAUD0I1	Interrupt for CH1 of TAUD0	TAUD0	Edge	1027$_H$	√	√	√	√	√	√			+09C$_H$
40	ICTAUD0I3	FFFF A050$_H$	INTTAUD0I3	Interrupt for CH3 of TAUD0	TAUD0	Edge	1028$_H$	√	√	√	√	√	√			+0A0$_H$
41	ICTAUD0I5	FFFF A052$_H$	INTTAUD0I5	Interrupt for CH5 of TAUD0	TAUD0	Edge	1029$_H$	√	√	√	√	√	√			+0A4$_H$
42	ICTAUD0I7	FFFF A054$_H$	INTTAUD0I7	Interrupt for CH7 of TAUD0	TAUD0	Edge	102A$_H$	√	√	√	√	√	√			+0A8$_H$
43	ICTAUD0I9	FFFF A056$_H$	INTTAUD0I9	Interrupt for CH9 of TAUD0	TAUD0	Edge	102B$_H$	√	√	√	√	√	√			+0AC$_H$
44	ICTAUD0I11	FFFF A058$_H$	INTTAUD0I11	Interrupt for CH11 of TAUD0	TAUD0	Edge	102C$_H$	√	√	√	√	√	√			+0B0$_H$
45	ICTAUD0I13	FFFF A05A$_H$	INTTAUD0I13	Interrupt for CH13 of TAUD0	TAUD0	Edge	102D$_H$	√	√	√	√	√	√			+0B4$_H$
46	ICTAUD0I15	FFFF A05C$_H$	INTTAUD0I15	Interrupt for CH15 of TAUD0	TAUD0	Edge	102E$_H$	√	√	√	√	√	√			+0B8$_H$
47	ICADCA0ERR	FFFF A05E$_H$	INTADCA0ERR	ADCA0 error interrupt	ADCA0	Edge	102F$_H$	√	√	√	√	√	√			+0BC$_H$
48	Reserved						1030$_H$	—	—	—	—	—	—			+0C0$_H$
49	ICCSIG0IRE	FFFF A062$_H$	INTCSIG0IRE	CSIG0 communication error interrupt	CSIG0	Edge	1031$_H$	√	√	√	√	√	√			+0C4$_H$
50	ICRLIN20	FFFF A064$_H$	INTRLIN20	RLIN20 interrupt	RLIN240	Edge	1032$_H$	√	√	√	√	√	√			+0C8$_H$
51	ICRLIN21	FFFF A066$_H$	INTRLIN21	RLIN21 interrupt	RLIN240	Edge	1033$_H$	√	√	√	√	√	√			+0CC$_H$
52	ICDMA0	FFFF A068$_H$	INTDMA0	DMA0 transfer completion	DMAC	Edge	1034$_H$	√	√	√	√	√	√			+0D0$_H$
53	ICDMA1	FFFF A06A$_H$	INTDMA1	DMA1 transfer completion	DMAC	Edge	1035$_H$	√	√	√	√	√	√			+0D4$_H$
54	ICDMA2	FFFF A06C$_H$	INTDMA2	DMA2 transfer completion	DMAC	Edge	1036$_H$	√	√	√	√	√	√			+0D8$_H$
55	ICDMA3	FFFF A06E$_H$	INTDMA3	DMA3 transfer completion	DMAC	Edge	1037$_H$	√	√	√	√	√	√			+0DC$_H$
56	ICDMA4	FFFF A070$_H$	INTDMA4	DMA4 transfer completion	DMAC	Edge	1038$_H$	√	√	√	√	√	√			+0E0$_H$
57	ICDMA5	FFFF A072$_H$	INTDMA5	DMA5 transfer completion	DMAC	Edge	1039$_H$	√	√	√	√	√	√			+0E4$_H$
58	ICDMA6	FFFF A074$_H$	INTDMA6	DMA6 transfer completion	DMAC	Edge	103A$_H$	√	√	√	√	√	√			+0E8$_H$
59	ICDMA7	FFFF A076$_H$	INTDMA7	DMA7 transfer completion	DMAC	Edge	103B$_H$	√	√	√	√	√	√			+0EC$_H$
60	ICDMA8	FFFF A078$_H$	INTDMA8	DMA8 transfer completion	DMAC	Edge	103C$_H$	√	√	√	√	√	√			+0F0$_H$
61	ICDMA9	FFFF A07A$_H$	INTDMA9	DMA9 transfer completion	DMAC	Edge	103D$_H$	√	√	√	√	√	√			+0F4$_H$
62	ICDMA10	FFFF A07C$_H$	INTDMA10	DMA10 transfer completion	DMAC	Edge	103E$_H$	√	√	√	√	√	√			+0F8$_H$
63	ICDMA11	FFFF A07E$_H$	INTDMA11	DMA11 transfer completion	DMAC	Edge	103F$_H$	√	√	√	√	√	√			+0FC$_H$
64	ICDMA12	FFFF A080$_H$	INTDMA12	DMA12 transfer completion	DMAC	Edge	1040$_H$	√	√	√	√	√	√			+100$_H$
65	ICDMA13	FFFF A082$_H$	INTDMA13	DMA13 transfer completion	DMAC	Edge	1041$_H$	√	√	√	√	√	√			+104$_H$
66	ICDMA14	FFFF A084$_H$	INTDMA14	DMA14 transfer completion	DMAC	Edge	1042$_H$	√	√	√	√	√	√			+108$_H$
67	ICDMA15	FFFF A086$_H$	INTDMA15	DMA15 transfer completion	DMAC	Edge	1043$_H$	√	√	√	√	√	√			+10C$_H$
68	ICRIIC0TI	FFFF A088$_H$	INTRIIC0TI	RIIC transmit end interrupt	RIIC0	Edge	1044$_H$	√	√	√	√	√	√			+110$_H$
69	ICRIIC0TEI	FFFF A08A$_H$	INTRIIC0TEI	RIIC transmit data empty interrupt	RIIC0	Level	1045$_H$	√	√	√	√	√	√			+114$_H$
70	ICRIIC0RI	FFFF A08C$_H$	INTRIIC0RI	RIIC receive end interrupt	RIIC0	Edge	1046$_H$	√	√	√	√	√	√			+118$_H$
71	ICRIIC0EE	FFFF A08E$_H$	INTRIIC0EE	RIIC receive error/event interrupt	RIIC0	Level	1047$_H$	√	√	√	√	√	√			+11C$_H$
72	ICTAUJ0I0	FFFF A090$_H$	INTTAUJ0I0	Interrupt for CH0 of TAUJ0	TAUJ0	Edge	1048$_H$	√	√	√	√	√	√			+120$_H$
73	ICTAUJ0I1	FFFF A092$_H$	INTTAUJ0I1	Interrupt for CH1 of TAUJ0	TAUJ0	Edge	1049$_H$	√	√	√	√	√	√			+124$_H$

Table 6.4 RH850/F1L EI Level Maskable Interrupt Sources (3/8)

Interrupt			Interrupt Request				Exception Request	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset)*8		
	Control Register													Direct Jumping to an Address		
Interrupt No.*1	Name	Address	Name	Source	Unit	Detection Type*2								RINT = 0*3	RINT = 1*4	Reference to a Table*5
74	ICTAUJ0I2	FFFF A094H	INTTAUJ0I2	Interrupt for CH2 of TAUJ0	TAUJ0	Edge	104AH	√	√	√	√	√	√	*11	*12	+128H
75	ICTAUJ0I3	FFFF A096H	INTTAUJ0I3	Interrupt for CH3 of TAUJ0	TAUJ0	Edge	104BH	√	√	√	√	√	√			+12CH
76	ICOSTM0	FFFF A098H	INTOSTM0	OSTM0 interrupt	OSTM0	Edge	104CH	√	√	√	√	√	√			+130H
77	ICENCA0IOV	FFFF A09AH	INTENCA0IOV*7	ENCA0 overflow interrupt	ENCA0	Edge	104DH	√	√	√	√	√	√			+134H
	ICPWGA4	FFFF A09AH	INTPWGA4*7	PWGA4 interrupt	PWGA4	Edge										
78	ICENCA0IUD	FFFF A09CH	INTENCA0IUD*7	ENCA0 underflow interrupt	ENCA0	Edge	104EH	√	√	√	√	√	√			+138H
	ICPWGA5	FFFF A09CH	INTPWGA5*7	PWGA5 interrupt	PWGA5	Edge										
79	ICENCA0I0	FFFF A09EH	INTENCA0I0*7	ENCA0 capture/compare match interrupt 0	ENCA0	Edge	104FH	√	√	√	√	√	√			+13CH
	ICPWGA6	FFFF A09EH	INTPWGA6*7	PWGA6 interrupt	PWGA6	Edge										
80	ICENCA0I1	FFFF A0A0H	INTENCA0I1*7	ENCA0 capture/compare match interrupt 1	ENCA0	Edge	1050H	√	√	√	√	√	√			+140H
	ICPWGA7	FFFF A0A0H	INTPWGA7*7	PWGA7 interrupt	PWGA7	Edge										
81	ICENCA0IEC	FFFF A0A2H	INTENCA0IEC	Encoder clear interrupt	ENCA0	Edge	1051H	√	√	√	√	√	√			+144H
82	ICKR0	FFFF A0A4H	INTKR0	Key interrupt	KR0	Edge	1052H	√	√	√	√	√	√			+148H
83	ICQFULL	FFFF A0A6H	INTQFULL	PWSA queue full interrupt	PWSA	Edge	1053H	√	√	√	√	√	√			+14CH
84	ICPWGA0	FFFF A0A8H	INTPWGA0	PWGA0 interrupt	PWGA0	Edge	1054H	√	√	√	√	√	√			+150H
85	ICPWGA1	FFFF A0AAH	INTPWGA1	PWGA1 interrupt	PWGA1	Edge	1055H	√	√	√	√	√	√			+154H
86	ICPWGA2	FFFF A0ACH	INTPWGA2	PWGA2 interrupt	PWGA2	Edge	1056H	√	√	√	√	√	√			+158H
87	ICPWGA3	FFFF A0AEH	INTPWGA3	PWGA3 interrupt	PWGA3	Edge	1057H	√	√	√	√	√	√			+15CH
88	ICPWGA8	FFFF A0B0H	INTPWGA8	PWGA8 interrupt	PWGA8	Edge	1058H	√	√	√	√	√	√			+160H
89	ICPWGA9	FFFF A0B2H	INTPWGA9	PWGA9 interrupt	PWGA9	Edge	1059H	√	√	√	√	√	√			+164H
90	ICPWGA10	FFFF A0B4H	INTPWGA10	PWGA10 interrupt	PWGA10	Edge	105AH	√	√	√	√	√	√			+168H
91	ICPWGA11	FFFF A0B6H	INTPWGA11	PWGA11 interrupt	PWGA11	Edge	105BH	√	√	√	√	√	√			+16CH
92	ICPWGA12	FFFF A0B8H	INTPWGA12	PWGA12 interrupt	PWGA12	Edge	105CH	√	√	√	√	√	√			+170H
93	ICPWGA13	FFFF A0BAH	INTPWGA13	PWGA13 interrupt	PWGA13	Edge	105DH	—	√	√	√	√	√			+174H
94	ICPWGA14	FFFF A0BCH	INTPWGA14	PWGA14 interrupt	PWGA14	Edge	105EH	—	√	√	√	√	√			+178H
95	ICPWGA15	FFFF A0BEH	INTPWGA15	PWGA15 interrupt	PWGA15	Edge	105FH	—	√	√	√	√	√			+17CH
96	Reserved						1060H	—	—	—	—	—	—			+180H
97	Reserved						1061H	—	—	—	—	—	—			+184H
98	Reserved						1062H	—	—	—	—	—	—			+188H
99	Reserved						1063H	—	—	—	—	—	—			+18CH
100	Reserved						1064H	—	—	—	—	—	—			+190H
101	Reserved						1065H	—	—	—	—	—	—			+194H
102	ICFLERR	FFFF A0CCH	INTFLERR	Flash sequencer end error interrupt	Flash	Level	1066H	√	√	√	√	√	√			+198H
103	ICFLENDNM	FFFF A0CEH	INTFLENDNM	Flash sequencer end interrupt	Flash	Edge	1067H	√	√	√	√	√	√			+19CH
104	ICCWEND	FFFF A0D0H	INTCWEND	Port polling end interrupt	LPS	Edge	1068H	√	√	√	√	√	√			+1A0H
105	ICRCAN1ERR	FFFF A0D2H	INTRCAN1ERR	CAN1 error interrupt	RSCAN0	Level	1069H	—	√	√	√	√	√			+1A4H
106	ICRCAN1REC	FFFF A0D4H	INTRCAN1REC	CAN1 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	106AH	—	√	√	√	√	√			+1A8H
107	ICRCAN1TRX	FFFF A0D6H	INTRCAN1TRX	CAN1 transmit interrupt	RSCAN0	Level	106BH	—	√	√	√	√	√			+1ACH
108	ICCSIH1IC	FFFF A0D8H	INTCSIH1IC*6	CSIH1 communication status interrupt	CSIH1	Edge	106CH	—	—	√	√	√	√			+1B0H
	ICTAPA0IPEK0_2	FFFF A0D8H	INTTAPA0IPEK0_2*6	TAPA0 peak interrupt 0	TAPA0	Edge										
109	ICCSIH1IR	FFFF A0DAH	INTCSIH1IR*6	CSIH1 receive status interrupt	CSIH1	Edge	106DH	—	—	√	√	√	√			+1B4H
	ICTAPA0IVLY0_2	FFFF A0DAH	INTTAPA0IVLY0_2*6	TAPA0 valley interrupt 0	TAPA0	Edge										

Table 6.4 RH850/F1L EI Level Maskable Interrupt Sources (4/8)

Interrupt			Interrupt Request				Exception Request	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset)*8		
														Direct Jumping to an Address		
Interrupt No.*1	Control Register					Detection Type*2										Reference to a Table*5
	Name	Address	Name	Source	Unit									RINT = 0*3	RINT = 1*4	
110	ICCSIH1IRE	FFFF A0DC$_H$	INTCSIH1IRE*6	CSIH1 communication error interrupt	CSIH1	Edge	106E$_H$	—	—	√	√	√	√	*11	*12	+1B8$_H$
	ICCSIG0IC_2	FFFF A0DC$_H$	INTCSIG0IC_2*6	CSIG0 communication status interrupt	CSIG0	Edge										
111	ICCSIH1IJC	FFFF A0DE$_H$	INTCSIH1IJC*6	CSIH1 job completion interrupt	CSIH1	Edge	106F$_H$	—	—	√	√	√	√			+1BC$_H$
	ICCSIG0IR_2	FFFF A0DE$_H$	INTCSIG0IR_2*6	CSIG0 receive status interrupt	CSIG0	Edge										
112	ICRLIN31	FFFF A0E0$_H$	INTRLIN31	RLIN31 interrupt	RLIN31	Edge	1070$_H$	—	√	√	√	√	√			+1C0$_H$
113	ICRLIN31UR0	FFFF A0E2$_H$	INTRLIN31UR0	RLIN31 transmit interrupt	RLIN31	Edge	1071$_H$	—	√	√	√	√	√			+1C4$_H$
114	ICRLIN31UR1	FFFF A0E4$_H$	INTRLIN31UR1	RLIN31 reception complete interrupt	RLIN31	Edge	1072$_H$	—	√	√	√	√	√			+1C8$_H$
115	ICRLIN31UR2	FFFF A0E6$_H$	INTRLIN31UR2	RLIN31 status interrupt	RLIN31	Edge	1073$_H$	—	√	√	√	√	√			+1CC$_H$
116	ICPWGA20	FFFF A0E8$_H$	INTPWGA20	PWGA20 interrupt	PWGA20	Edge	1074$_H$	—	√	√	√	√	√			+1D0$_H$
117	ICPWGA21	FFFF A0EA$_H$	INTPWGA21	PWGA21 interrupt	PWGA21	Edge	1075$_H$	—	√	√	√	√	√			+1D4$_H$
118	ICPWGA22	FFFF A0EC$_H$	INTPWGA22	PWGA22 interrupt	PWGA22	Edge	1076$_H$	—	√	√	√	√	√			+1D8$_H$
119	ICPWGA23	FFFF A0EE$_H$	INTPWGA23	PWGA23 interrupt	PWGA23	Edge	1077$_H$	—	√	√	√	√	√			+1DC$_H$
120	ICP6	FFFF A0F0$_H$	INTP6	External interrupt	Port	Edge	1078$_H$	—	—	√	√	√	√			+1E0$_H$
121	ICP7	FFFF A0F2$_H$	INTP7	External interrupt	Port	Edge	1079$_H$	—	—	√	√	√	√			+1E4$_H$
122	ICP8	FFFF A0F4$_H$	INTP8	External interrupt	Port	Edge	107A$_H$	—	—	√	√	√	√			+1E8$_H$
123	ICP12	FFFF A0F6$_H$	INTP12	External interrupt	Port	Edge	107B$_H$	—	—	√	√	√	√			+1EC$_H$
124	ICCSIH2IC	FFFF A0F8$_H$	INTCSIH2IC*6	CSIH2 communication status interrupt	CSIH2	Edge	107C$_H$	—	—	√	√	√	√			+1F0$_H$
	ICTAUD0I0_2	FFFF A0F8$_H$	INTTAUD0I0_2*6	TAUD0 CH0 Interrupt	TAUD0	Edge										
125	ICCSIH2IR	FFFF A0FA$_H$	INTCSIH2IR*6	CSIH2 receive status interrupt	CSIH2	Edge	107D$_H$	—	—	√	√	√	√			+1F4$_H$
	ICP0_2	FFFF A0FA$_H$	INTP0_2*6	External interrupt	Port	Edge										
126	ICCSIH2IRE	FFFF A0FC$_H$	INTCSIH2IRE*6	CSIH2 communication error interrupt	CSIH2	Edge	107E$_H$	—	—	√	√	√	√			+1F8$_H$
	ICP1_2	FFFF A0FC$_H$	INTP1_2*6	External interrupt	Port	Edge										
127	ICCSIH2IJC	FFFF A0FE$_H$	INTCSIH2IJC*6	CSIH2 job completion interrupt	CSIH2	Edge	107F$_H$	—	—	√	√	√	√			+1FC$_H$
	ICP2_2	FFFF A0FE$_H$	INTP2_2*6	External interrupt	Port	Edge										
128	Reserved						1080$_H$	—	—	—	—	—	—			+200$_H$
129	Reserved						1081$_H$	—	—	—	—	—	—			+204$_H$
130	Reserved						1082$_H$	—	—	—	—	—	—			+208$_H$
131	Reserved						1083$_H$	—	—	—	—	—	—			+20C$_H$
132	Reserved						1084$_H$	—	—	—	—	—	—			+210$_H$
133	Reserved						1085$_H$	—	—	—	—	—	—			+214$_H$
134	ICTAUB0I0	FFFF A10C$_H$	INTTAUB0I0	Interrupt for CH0 of TAUB0	TAUB0	Edge	1086$_H$	—	—	—	√	√	√			+218$_H$
135	ICTAUB0I1	FFFF A10E$_H$	INTTAUB0I1	Interrupt for CH1 of TAUB0	TAUB0	Edge	1087$_H$	—	—	—	√	√	√			+21C$_H$
136	ICTAUB0I2	FFFF A110$_H$	INTTAUB0I2	Interrupt for CH2 of TAUB0	TAUB0	Edge	1088$_H$	—	—	—	√	√	√			+220$_H$
137	ICTAUB0I3	FFFF A112$_H$	INTTAUB0I3*7	Interrupt for CH3 of TAUB0	TAUB0	Edge	1089$_H$	—	—	—	√	√	√			+224$_H$
	ICPWGA16	FFFF A112$_H$	INTPWGA16*7	PWGA16 interrupt	PWGA16	Edge		—	√	√						
138	ICTAUB0I4	FFFF A114$_H$	INTTAUB0I4	Interrupt for CH4 of TAUB0	TAUB0	Edge	108A$_H$	—	—	—	√	√	√			+228$_H$
139	ICTAUB0I5	FFFF A116$_H$	INTTAUB0I5*7	Interrupt for CH5 of TAUB0	TAUB0	Edge	108B$_H$	—	—	—	√	√	√			+22C$_H$
	ICPWGA17	FFFF A116$_H$	INTPWGA17*7	PWGA17 interrupt	PWGA17	Edge		—	√	√						
140	ICTAUB0I6	FFFF A118$_H$	INTTAUB0I6	Interrupt for CH6 of TAUB0	TAUB0	Edge	108C$_H$	—	—	—	√	√	√			+230$_H$
141	ICTAUB0I7	FFFF A11A$_H$	INTTAUB0I7*7	Interrupt for CH7 of TAUB0	TAUB0	Edge	108D$_H$	—	—	—	√	√	√			+234$_H$
	ICPWGA18	FFFF A11A$_H$	INTPWGA18*7	PWGA18 interrupt	PWGA18	Edge		—	√	√						
142	ICTAUB0I8	FFFF A11C$_H$	INTTAUB0I8	Interrupt for CH8 of TAUB0	TAUB0	Edge	108E$_H$	—	—	—	√	√	√			+238$_H$
143	ICTAUB0I9	FFFF A11E$_H$	INTTAUB0I9*7	Interrupt for CH9 of TAUB0	TAUB0	Edge	108F$_H$	—	—	—	√	√	√			+23C$_H$
	ICPWGA19	FFFF A11E$_H$	INTPWGA19*7	PWGA19 interrupt	PWGA19	Edge		—	√	√						
144	ICTAUB0I10	FFFF A120$_H$	INTTAUB0I10	Interrupt for CH10 of TAUB0	TAUB0	Edge	1090$_H$	—	—	—	√	√	√			+240$_H$

Table 6.4 RH850/F1L EI Level Maskable Interrupt Sources (5/8)

Interrupt			Interrupt Request				Exception Request	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset)*8		
	Control Register													Direct Jumping to an Address		
Interrupt No.*1	Name	Address	Name	Source	Unit	Detection Type*2								RINT = 0*3	RINT = 1*4	Reference to a Table*5
145	ICTAUB0I11	FFFF A122$_H$	INTTAUB0I11*7	Interrupt for CH11 of TAUB0	TAUB0	Edge	1091$_H$	—	—	—	√	√	√	*11	*12	+244$_H$
	ICPWGA26	FFFF A122$_H$	INTPWGA26*7	PWGA26 interrupt	PWGA26	Edge										
146	ICTAUB0I12	FFFF A124$_H$	INTTAUB0I12	Interrupt for CH12 of TAUB0	TAUB0	Edge	1092$_H$	—	—	—	√	√	√			+248$_H$
147	ICTAUB0I13	FFFF A126$_H$	INTTAUB0I13*7	Interrupt for CH13 of TAUB0	TAUB0	Edge	1093$_H$	—	—	—	√	√	√			+24C$_H$
	ICPWGA30	FFFF A126$_H$	INTPWGA30*7	PWGA30 interrupt	PWGA30	Edge										
148	ICTAUB0I14	FFFF A128$_H$	INTTAUB0I14	Interrupt for CH14 of TAUB0	TAUB0	Edge	1094$_H$	—	—	—	√	√	√			+250$_H$
149	ICTAUB0I15	FFFF A12A$_H$	INTTAUB0I15*7	Interrupt for CH15 of TAUB0	TAUB0	Edge	1095$_H$	—	—	—	√	√	√			+254$_H$
	ICPWGA31	FFFF A12A$_H$	INTPWGA31*7	PWGA31 interrupt	PWGA31	Edge										
150	ICCSIH3IC	FFFF A12C$_H$	INTCSIH3IC*6	CSIH3 communication status interrupt	CSIH3	Edge	1096$_H$	—	—	—	√	√	√			+258$_H$
	ICTAUD0I2_2	FFFF A12C$_H$	INTTAUD0I2_2*6	Interrupt for CH2 of TAUD0	TAUD0	Edge										
151	ICCSIH3IR	FFFF A12E$_H$	INTCSIH3IR*6	CSIH3 receive status interrupt	CSIH3	Edge	1097$_H$	—	—	—	√	√	√			+25C$_H$
	ICTAUD0I10_2	FFFF A12E$_H$	INTTAUD0I10_2*6	Interrupt for CH10 of TAUD0	TAUD0	Edge										
152	ICCSIH3IRE	FFFF A130$_H$	INTCSIH3IRE*6	CSIH3 communication error interrupt	CSIH3	Edge	1098$_H$	—	—	—	√	√	√			+260$_H$
	ICTAUD0I12_2	FFFF A130$_H$	INTTAUD0I12_2*6	Interrupt for CH12 of TAUD0	TAUD0	Edge										
153	ICCSIH3IJC	FFFF A132$_H$	INTCSIH3IJC*6	CSIH3 job completion interrupt	CSIH3	Edge	1099$_H$	—	—	—	√	√	√			+264$_H$
	ICTAUD0I14_2	FFFF A132$_H$	INTTAUD0I14_2*6	Interrupt for CH14 of TAUD0	TAUD0	Edge										
154	ICRLIN22	FFFF A134$_H$	INTRLIN22	RLIN22 interrupt	RLIN240	Edge	109A$_H$	—	—	—	√	√	√			+268$_H$
155	ICRLIN23	FFFF A136$_H$	INTRLIN23	RLIN23 interrupt	RLIN240	Edge	109B$_H$	—	—	—	—	√	√			+26C$_H$
156	ICRLIN32	FFFF A138$_H$	INTRLIN32	RLIN32 interrupt	RLIN32	Edge	109C$_H$	—	—	√	√	√	√			+270$_H$
157	ICRLIN32UR0	FFFF A13A$_H$	INTRLIN32UR0	RLIN32 transmit interrupt	RLIN32	Edge	109D$_H$	—	—	√	√	√	√			+274$_H$
158	ICRLIN32UR1	FFFF A13C$_H$	INTRLIN32UR1	RLIN32 reception complete interrupt	RLIN32	Edge	109E$_H$	—	—	√	√	√	√			+278$_H$
159	ICRLIN32UR2	FFFF A13E$_H$	INTRLIN32UR2	RLIN32 status interrupt	RLIN32	Edge	109F$_H$	—	—	√	√	√	√			+27C$_H$
160	ICTAUJ1I0	FFFF A140$_H$	INTTAUJ1I0	Interrupt for CH0 of TAUJ1	TAUJ1	Edge	10A0$_H$	—	—	—	√	√	√			+280$_H$
161	ICTAUJ1I1	FFFF A142$_H$	INTTAUJ1I1	Interrupt for CH1 of TAUJ1	TAUJ1	Edge	10A1$_H$	—	—	—	√	√	√			+284$_H$
162	ICTAUJ1I2	FFFF A144$_H$	INTTAUJ1I2	Interrupt for CH2 of TAUJ1	TAUJ1	Edge	10A2$_H$	—	—	—	√	√	√			+288$_H$
163	ICTAUJ1I3	FFFF A146$_H$	INTTAUJ1I3	Interrupt for CH3 of TAUJ1	TAUJ1	Edge	10A3$_H$	—	—	—	√	√	√			+28C$_H$
164	Reserved						10A4$_H$	—	—	—	—	—	—			+290$_H$
165	Reserved						10A5$_H$	—	—	—	—	—	—			+294$_H$
166	Reserved						10A6$_H$	—	—	—	—	—	—			+298$_H$
167	Reserved						10A7$_H$	—	—	—	—	—	—			+29C$_H$
168	Reserved						10A8$_H$	—	—	—	—	—	—			+2A0$_H$
169	Reserved						10A9$_H$	—	—	—	—	—	—			+2A4$_H$
170	Reserved						10AA$_H$	—	—	—	—	—	—			+2A8$_H$
171	Reserved						10AB$_H$	—	—	—	—	—	—			+2AC$_H$
172	Reserved						10AC$_H$	—	—	—	—	—	—			+2B0$_H$
173	Reserved						10AD$_H$	—	—	—	—	—	—			+2B4$_H$
174	Reserved						10AE$_H$	—	—	—	—	—	—			+2B8$_H$
175	Reserved						10AF$_H$	—	—	—	—	—	—			+2BC$_H$
176	ICPWGA24	FFFF A160$_H$	INTPWGA24	PWGA24 interrupt	PWGA24	Edge	10B0$_H$	—	—	—	√	√	√			+2C0$_H$
177	ICPWGA25	FFFF A162$_H$	INTPWGA25	PWGA25 interrupt	PWGA25	Edge	10B1$_H$	—	—	—	√	√	√			+2C4$_H$
178	ICPWGA27	FFFF A164$_H$	INTPWGA27	PWGA27 interrupt	PWGA27	Edge	10B2$_H$	—	—	—	√	√	√			+2C8$_H$
179	ICPWGA28	FFFF A166$_H$	INTPWGA28	PWGA28 interrupt	PWGA28	Edge	10B3$_H$	—	—	—	√	√	√			+2CC$_H$
180	ICPWGA29	FFFF A168$_H$	INTPWGA29	PWGA29 interrupt	PWGA29	Edge	10B4$_H$	—	—	—	√	√	√			+2D0$_H$
181	ICPWGA32	FFFF A16A$_H$	INTPWGA32	PWGA32 interrupt	PWGA32	Edge	10B5$_H$	—	—	—	√	√	√			+2D4$_H$
182	ICPWGA33	FFFF A16C$_H$	INTPWGA33	PWGA33 interrupt	PWGA33	Edge	10B6$_H$	—	—	—	√	√	√			+2D8$_H$
183	ICPWGA34	FFFF A16E$_H$	INTPWGA34	PWGA34 interrupt	PWGA34	Edge	10B7$_H$	—	—	—	√	√	√			+2DC$_H$

Table 6.4 RH850/F1L EI Level Maskable Interrupt Sources (6/8)

Interrupt			Interrupt Request				Exception Request	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset)*8		
	Control Register													Direct Jumping to an Address		
Interrupt No.*1	Name	Address	Name	Source	Unit	Detection Type*2								RINT = 0*3	RINT = 1*4	Reference to a Table*5
184	ICPWGA35	FFFF A170$_H$	INTPWGA35	PWGA35 interrupt	PWGA35	Edge	10B8$_H$	—	—	—	√	√	√	*11	*12	+2E0$_H$
185	ICPWGA36	FFFF A172$_H$	INTPWGA36	PWGA36 interrupt	PWGA36	Edge	10B9$_H$	—	—	—	√	√	√			+2E4$_H$
186	ICPWGA37	FFFF A174$_H$	INTPWGA37	PWGA37 interrupt	PWGA37	Edge	10BA$_H$	—	—	—	√	√	√			+2E8$_H$
187	ICPWGA38	FFFF A176$_H$	INTPWGA38	PWGA38 interrupt	PWGA38	Edge	10BB$_H$	—	—	—	√	√	√			+2EC$_H$
188	ICPWGA39	FFFF A178$_H$	INTPWGA39	PWGA39 interrupt	PWGA39	Edge	10BC$_H$	—	—	—	√	√	√			+2F0$_H$
189	ICPWGA40	FFFF A17A$_H$	INTPWGA40	PWGA40 interrupt	PWGA40	Edge	10BD$_H$	—	—	—	√	√	√			+2F4$_H$
190	ICPWGA41	FFFF A17C$_H$	INTPWGA41	PWGA41 interrupt	PWGA41	Edge	10BE$_H$	—	—	—	√	√	√			+2F8$_H$
191	ICPWGA42	FFFF A17E$_H$	INTPWGA42	PWGA42 interrupt	PWGA42	Edge	10BF$_H$	—	—	—	√	√	√			+2FC$_H$
192	ICPWGA43	FFFF A180$_H$	INTPWGA43	PWGA43 interrupt	PWGA43	Edge	10C0$_H$	—	—	—	√	√	√			+300$_H$
193	ICPWGA44	FFFF A182$_H$	INTPWGA44	PWGA44 interrupt	PWGA44	Edge	10C1$_H$	—	—	—	√	√	√			+304$_H$
194	ICPWGA45	FFFF A184$_H$	INTPWGA45	PWGA45 interrupt	PWGA45	Edge	10C2$_H$	—	—	—	√	√	√			+308$_H$
195	ICPWGA46	FFFF A186$_H$	INTPWGA46	PWGA46 interrupt	PWGA46	Edge	10C3$_H$	—	—	—	√	√	√			+30C$_H$
196	ICPWGA47	FFFF A188$_H$	INTPWGA47	PWGA47 interrupt	PWGA47	Edge	10C4$_H$	—	—	—	√	√	√			+310$_H$
197	ICP9	FFFF A18A$_H$	INTP9	External interrupt	Port	Edge	10C5$_H$	—	—	—	—	√	√			+314$_H$
198	ICP13	FFFF A18C$_H$	INTP13	External interrupt	Port	Edge	10C6$_H$	—	—	—	√	√	√			+318$_H$
199	ICP14	FFFF A18E$_H$	INTP14	External interrupt	Port	Edge	10C7$_H$	—	—	—	—	√	√			+31C$_H$
200	ICP15	FFFF A190$_H$	INTP15	External interrupt	Port	Edge	10C8$_H$	—	—	—	—	√	√			+320$_H$
201	ICRTCA01S	FFFF A192$_H$	INTRTCA01S	1 second interval interrupt	RTCA	Edge	10C9$_H$	—	—	—	—	√	√			+324$_H$
202	ICRTCA0AL	FFFF A194$_H$	INTRTCA0AL	Alarm interrupt	RTCA	Edge	10CA$_H$	—	—	—	—	√	√			+328$_H$
203	ICRTCA0R	FFFF A196$_H$	INTRTCA0R	Periodic interrupt	RTCA	Edge	10CB$_H$	—	—	—	—	√	√			+32C$_H$
204	ICADCA1ERR	FFFF A198$_H$	INTADCA1ERR	ADCA1 error interrupt	ADCA1	Edge	10CC$_H$	—	—	—	—	√	√			+330$_H$
205	ICADCA1I0	FFFF A19A$_H$	INTADCA1I0	ADCA1 SG1 end interrupt	ADCA1	Edge	10CD$_H$	—	—	—	—	√	√			+334$_H$
206	ICADCA1I1	FFFF A19C$_H$	INTADCA1I1	ADCA1 SG2 end interrupt	ADCA1	Edge	10CE$_H$	—	—	—	—	√	√			+338$_H$
207	ICADCA1I2	FFFF A19E$_H$	INTADCA1I2	ADCA1 SG3 end interrupt	ADCA1	Edge	10CF$_H$	—	—	—	—	√	√			+33C$_H$
208	Reserved						10D0$_H$	—	—	—	—	—	—			+340$_H$
209	ICRCAN2ERR	FFFF A1A2$_H$	INTRCAN2ERR	CAN2 error interrupt	RSCAN0	Level	10D1$_H$	—	√	√	√	√	√			+344$_H$
210	ICRCAN2REC	FFFF A1A4$_H$	INTRCAN2REC	CAN2 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	10D2$_H$	—	√	√	√	√	√			+348$_H$
211	ICRCAN2TRX	FFFF A1A6$_H$	INTRCAN2TRX	CAN2 transmit interrupt	RSCAN0	Level	10D3$_H$	—	√	√	√	√	√			+34C$_H$
212	ICRCAN3ERR	FFFF A1A8$_H$	INTRCAN3ERR	CAN3 error interrupt	RSCAN0	Level	10D4$_H$	—	—	—	*9	√	√			+350$_H$
213	ICRCAN3REC	FFFF A1AA$_H$	INTRCAN3REC	CAN3 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	10D5$_H$	—	—	—	*9	√	√			+354$_H$
214	ICRCAN3TRX	FFFF A1AC$_H$	INTRCAN3TRX	CAN3 transmit interrupt	RSCAN0	Level	10D6$_H$	—	—	—	*9	√	√			+358$_H$
215	ICCSIG1IC	FFFF A1AE$_H$	INTCSIG1IC	CSIG1 communication status interrupt	CSIG1	Edge	10D7$_H$	—	—	—	—	√	√			+35C$_H$
216	ICCSIG1IR	FFFF A1B0$_H$	INTCSIG1IR	CSIG1 receive status interrupt	CSIG1	Edge	10D8$_H$	—	—	—	—	√	√			+360$_H$
217	ICCSIG1IRE	FFFF A1B2$_H$	INTCSIG1IRE	CSIG1 receive error interrupt	CSIG1	Edge	10D9$_H$	—	—	—	—	√	√			+364$_H$
218	ICRLIN24	FFFF A1B4$_H$	INTRLIN24	RLIN24 interrupt	RLIN241	Edge	10DA$_H$	—	—	—	—	√	√			+368$_H$
219	ICRLIN25	FFFF A1B6$_H$	INTRLIN25	RLIN25 interrupt	RLIN241	Edge	10DB$_H$	—	—	—	—	√	√			+36C$_H$
220	ICRLIN33	FFFF A1B8$_H$	INTRLIN33	RLIN33 interrupt	RLIN33	Edge	10DC$_H$	—	—	—	√	√	√			+370$_H$
221	ICRLIN33UR0	FFFF A1BA$_H$	INTRLIN33UR0	RLIN33 transmit interrupt	RLIN33	Edge	10DD$_H$	—	—	—	√	√	√			+374$_H$
222	ICRLIN33UR1	FFFF A1BC$_H$	INTRLIN33UR1	RLIN33 reception complete interrupt	RLIN33	Edge	10DE$_H$	—	—	—	√	√	√			+378$_H$
223	ICRLIN33UR2	FFFF A1BE$_H$	INTRLIN33UR2	RLIN33 status interrupt	RLIN33	Edge	10DF$_H$	—	—	—	√	√	√			+37C$_H$
224	ICRLIN34	FFFF A1C0$_H$	INTRLIN34	RLIN34 interrupt	RLIN34	Edge	10E0$_H$	—	—	—	—	√	√			+380$_H$
225	ICRLIN34UR0	FFFF A1C2$_H$	INTRLIN34UR0	RLIN34 transmit interrupt	RLIN34	Edge	10E1$_H$	—	—	—	—	√	√			+384$_H$
226	ICRLIN34UR1	FFFF A1C4$_H$	INTRLIN34UR1	RLIN34 reception complete interrupt	RLIN34	Edge	10E2$_H$	—	—	—	—	√	√			+388$_H$
227	ICRLIN34UR2	FFFF A1C6$_H$	INTRLIN34UR2	RLIN34 status interrupt	RLIN34	Edge	10E3$_H$	—	—	—	—	√	√			+38C$_H$
228	ICRLIN35	FFFF A1C8$_H$	INTRLIN35	RLIN35 interrupt	RLIN35	Edge	10E4$_H$	—	—	—	—	√	√			+390$_H$

Table 6.4 RH850/F1L EI Level Maskable Interrupt Sources (7/8)

Interrupt			Interrupt Request				Exception Request	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset)*8		
	Control Register													Direct Jumping to an Address		
Interrupt No.*1	Name	Address	Name	Source	Unit	Detection Type*2								RINT = 0*3	RINT = 1*4	Reference to a Table*5
229	ICRLIN35UR0	FFFF A1CA$_H$	INTRLIN35UR0	RLIN35 transmit interrupt	RLIN35	Edge	10E5$_H$	—	—	—	—	√	√	*11	*12	+394$_H$
230	ICRLIN35UR1	FFFF A1CC$_H$	INTRLIN35UR1	RLIN35 reception complete interrupt	RLIN35	Edge	10E6$_H$	—	—	—	—	√	√			+398$_H$
231	ICRLIN35UR2	FFFF A1CE$_H$	INTRLIN35UR2	RLIN35 status interrupt	RLIN35	Edge	10E7$_H$	—	—	—	—	√	√			+39C$_H$
232	ICPWGA48	FFFF A1D0$_H$	INTPWGA48	PWGA48 interrupt	PWGA48	Edge	10E8$_H$	—	—	—	—	√	√			+3A0$_H$
233	ICPWGA49	FFFF A1D2$_H$	INTPWGA49	PWGA49 interrupt	PWGA49	Edge	10E9$_H$	—	—	—	—	√	√			+3A4$_H$
234	ICPWGA50	FFFF A1D4$_H$	INTPWGA50	PWGA50 interrupt	PWGA50	Edge	10EA$_H$	—	—	—	—	√	√			+3A8$_H$
235	ICPWGA51	FFFF A1D6$_H$	INTPWGA51	PWGA51 interrupt	PWGA51	Edge	10EB$_H$	—	—	—	—	√	√			+3AC$_H$
236	ICPWGA52	FFFF A1D8$_H$	INTPWGA52	PWGA52 interrupt	PWGA52	Edge	10EC$_H$	—	—	—	—	√	√			+3B0$_H$
237	ICPWGA53	FFFF A1DA$_H$	INTPWGA53	PWGA53 interrupt	PWGA53	Edge	10ED$_H$	—	—	—	—	√	√			+3B4$_H$
238	ICPWGA54	FFFF A1DC$_H$	INTPWGA54	PWGA54 interrupt	PWGA54	Edge	10EE$_H$	—	—	—	—	√	√			+3B8$_H$
239	ICPWGA55	FFFF A1DE$_H$	INTPWGA55	PWGA55 interrupt	PWGA55	Edge	10EF$_H$	—	—	—	—	√	√			+3BC$_H$
240	ICPWGA56	FFFF A1E0$_H$	INTPWGA56	PWGA56 interrupt	PWGA56	Edge	10F0$_H$	—	—	—	—	√	√			+3C0$_H$
241	ICPWGA57	FFFF A1E2$_H$	INTPWGA57	PWGA57 interrupt	PWGA57	Edge	10F1$_H$	—	—	—	—	√	√			+3C4$_H$
242	ICPWGA58	FFFF A1E4$_H$	INTPWGA58	PWGA58 interrupt	PWGA58	Edge	10F2$_H$	—	—	—	—	√	√			+3C8$_H$
243	ICPWGA59	FFFF A1E6$_H$	INTPWGA59	PWGA59 interrupt	PWGA59	Edge	10F3$_H$	—	—	—	—	√	√			+3CC$_H$
244	ICPWGA60	FFFF A1E8$_H$	INTPWGA60	PWGA60 interrupt	PWGA60	Edge	10F4$_H$	—	—	—	—	√	√			+3D0$_H$
245	ICPWGA61	FFFF A1EA$_H$	INTPWGA61	PWGA61 interrupt	PWGA61	Edge	10F5$_H$	—	—	—	—	√	√			+3D4$_H$
246	ICPWGA62	FFFF A1EC$_H$	INTPWGA62	PWGA62 interrupt	PWGA62	Edge	10F6$_H$	—	—	—	—	√	√			+3D8$_H$
247	ICPWGA63	FFFF A1EE$_H$	INTPWGA63	PWGA63 interrupt	PWGA63	Edge	10F7$_H$	—	—	—	—	√	√			+3DC$_H$
248	ICTAUB1I0	FFFF A1F0$_H$	INTTAUB1I0	Interrupt for CH0 of TAUB1	TAUB1	Edge	10F8$_H$	—	—	—	—	—	√			+3E0$_H$
249	ICTAUB1I1	FFFF A1F2$_H$	INTTAUB1I1	Interrupt for CH1 of TAUB1	TAUB1	Edge	10F9$_H$	—	—	—	—	—	√			+3E4$_H$
250	ICTAUB1I2	FFFF A1F4$_H$	INTTAUB1I2	Interrupt for CH2 of TAUB1	TAUB1	Edge	10FA$_H$	—	—	—	—	—	√			+3E8$_H$
251	ICTAUB1I3	FFFF A1F6$_H$	INTTAUB1I3	Interrupt for CH3 of TAUB1	TAUB1	Edge	10FB$_H$	—	—	—	—	—	√			+3EC$_H$
252	ICTAUB1I4	FFFF A1F8$_H$	INTTAUB1I4	Interrupt for CH4 of TAUB1	TAUB1	Edge	10FC$_H$	—	—	—	—	—	√			+3F0$_H$
253	ICTAUB1I5	FFFF A1FA$_H$	INTTAUB1I5	Interrupt for CH5 of TAUB1	TAUB1	Edge	10FD$_H$	—	—	—	—	—	√			+3F4$_H$
254	ICTAUB1I6	FFFF A1FC$_H$	INTTAUB1I6	Interrupt for CH6 of TAUB1	TAUB1	Edge	10FE$_H$	—	—	—	—	—	√			+3F8$_H$
255	ICTAUB1I7	FFFF A1FE$_H$	INTTAUB1I7	Interrupt for CH7 of TAUB1	TAUB1	Edge	10FF$_H$	—	—	—	—	—	√			+3FC$_H$
256	ICTAUB1I8	FFFF A200$_H$	INTTAUB1I8	Interrupt for CH8 of TAUB1	TAUB1	Edge	1100$_H$	—	—	—	—	—	√			+400$_H$
257	ICTAUB1I9	FFFF A202$_H$	INTTAUB1I9	Interrupt for CH9 of TAUB1	TAUB1	Edge	1101$_H$	—	—	—	—	—	√			+404$_H$
258	ICTAUB1I10	FFFF A204$_H$	INTTAUB1I10	Interrupt for CH10 of TAUB1	TAUB1	Edge	1102$_H$	—	—	—	—	—	√			+408$_H$
259	ICTAUB1I11	FFFF A206$_H$	INTTAUB1I11	Interrupt for CH11 of TAUB1	TAUB1	Edge	1103$_H$	—	—	—	—	—	√			+40C$_H$
260	ICTAUB1I12	FFFF A208$_H$	INTTAUB1I12	Interrupt for CH12 of TAUB1	TAUB1	Edge	1104$_H$	—	—	—	—	—	√			+410$_H$
261	ICTAUB1I13	FFFF A20A$_H$	INTTAUB1I13	Interrupt for CH13 of TAUB1	TAUB1	Edge	1105$_H$	—	—	—	—	—	√			+414$_H$
262	ICTAUB1I14	FFFF A20C$_H$	INTTAUB1I14	Interrupt for CH14 of TAUB1	TAUB1	Edge	1106$_H$	—	—	—	—	—	√			+418$_H$
263	ICTAUB1I15	FFFF A20E$_H$	INTTAUB1I15	Interrupt for CH15 of TAUB1	TAUB1	Edge	1107$_H$	—	—	—	—	—	√			+41C$_H$
264	ICRCAN4ERR	FFFF A210$_H$	INTRCAN4ERR	CAN4 error interrupt	RSCAN0	Level	1108$_H$	—	—	—	*9	*10	√			+420$_H$
265	ICRCAN4REC	FFFF A212$_H$	INTRCAN4REC	CAN4 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	1109$_H$	—	—	—	*9	*10	√			+424$_H$
266	ICRCAN4TRX	FFFF A214$_H$	INTRCAN4TRX	CAN4 transmit interrupt	RSCAN0	Level	110A$_H$	—	—	—	*9	*10	√			+428$_H$
267	ICRLIN26	FFFF A216$_H$	INTRLIN26	RLIN26 interrupt	RLIN241	Edge	110B$_H$	—	—	—	—	—	√			+42C$_H$
268	ICRLIN27	FFFF A218$_H$	INTRLIN27	RLIN27 interrupt	RLIN241	Edge	110C$_H$	—	—	—	—	—	√			+430$_H$
269	ICPWGA64	FFFF A21A$_H$	INTPWGA64	PWGA64 interrupt	PWGA64	Edge	110D$_H$	—	—	—	—	—	√			+434$_H$
270	ICPWGA65	FFFF A21C$_H$	INTPWGA65	PWGA65 interrupt	PWGA65	Edge	110E$_H$	—	—	—	—	—	√			+438$_H$
271	ICPWGA66	FFFF A21E$_H$	INTPWGA66	PWGA66 interrupt	PWGA66	Edge	110F$_H$	—	—	—	—	—	√			+43C$_H$
272	ICPWGA67	FFFF A220$_H$	INTPWGA67	PWGA67 interrupt	PWGA67	Edge	1110$_H$	—	—	—	—	—	√			+440$_H$
273	ICPWGA68	FFFF A222$_H$	INTPWGA68	PWGA68 interrupt	PWGA68	Edge	1111$_H$	—	—	—	—	—	√			+444$_H$
274	ICPWGA69	FFFF A224$_H$	INTPWGA69	PWGA69 interrupt	PWGA69	Edge	1112$_H$	—	—	—	—	—	√			+448$_H$

Table 6.4 RH850/F1L EI Level Maskable Interrupt Sources (8/8)

Interrupt			Interrupt Request				Exception Request	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins	Handler Address (Offset)*8	
	Control Register													Direct Jumping to an Address	
Interrupt No.*1	Name	Address	Name	Source	Unit	Detection Type*2								RINT = 0*3	RINT = 1*4	Reference to a Table*5
275	ICPWGA70	FFFF A226$_H$	INTPWGA70	PWGA70 interrupt	PWGA70	Edge	1113$_H$	—	—	—	—	—	√	*11	*12	+44C$_H$
276	ICPWGA71	FFFF A228$_H$	INTPWGA71	PWGA71 interrupt	PWGA71	Edge	1114$_H$	—	—	—	—	—	√			+450$_H$
277	ICRLIN28	FFFF A22A$_H$	INTRLIN28	RLIN28 interrupt	RLIN210	Edge	1115$_H$	—	—	—	—	—	√			+454$_H$
278	ICRLIN29	FFFF A22C$_H$	INTRLIN29	RLIN29 interrupt	RLIN211	Edge	1116$_H$	—	—	—	—	—	√			+458$_H$
279	ICRCAN5ERR	FFFF A22E$_H$	INTRCAN5ERR	CAN5 error interrupt	RSCAN0	Level	1117$_H$	—	—	—	*9	*10	√			+45C$_H$
280	ICRCAN5REC	FFFF A230$_H$	INTRCAN5REC	CAN5 transmit/receive FIFO receive complete interrupt	RSCAN0	Level	1118$_H$	—	—	—	*9	*10	√			+460$_H$
281	ICRCAN5TRX	FFFF A232$_H$	INTRCAN5TRX	CAN5 transmit interrupt	RSCAN0	Level	1119$_H$	—	—	—	*9	*10	√			+464$_H$
282	Reserved						111A$_H$	—	—	—	—	—	—			+468$_H$
283	Reserved						111B$_H$	—	—	—	—	—	—			+46C$_H$
284	Reserved						111C$_H$	—	—	—	—	—	—			+470$_H$
285	Reserved						111D$_H$	—	—	—	—	—	—			+474$_H$
286	Reserved						111E$_H$	—	—	—	—	—	—			+478$_H$
287	Reserved						111F$_H$	—	—	—	—	—	—			+47C$_H$

Note 1. Each interrupt is connected to INTC1 channel 0 to 31 and INTC2 channel 32 to 287.

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 6.4.1, ICxxx — EI Level Interrupt Control Registers**.

Note 3. Irrespective of interrupt channels, an offset address is determined in the range from +100$_H$ to 1F0$_H$ according to the priority order (0 to 7).

Note 4. Irrespective of the priority order, 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 6.5.1, SELB_INTC1 — INTC1 Interrupt Select Register**.

Note 7. Two interrupt sources are assigned to the same interrupt channel. For details, see **Section 6.5.2, SELB_INTC2 — INTC2 Interrupt Select Register**.

Note 8. For details, see **Section 6.10, Exception Handler Address**.

Note 9. Included only in the F1L for Gateway.

Note 10. Only available in devices with 1.5- and 2-MB code flash memories.

Note 11. The offset address does not differ from channel to channel; it is determined in the range from +100$_H$ to 1F0$_H$ according to the priority order (0 to 7).

Note 12. The offset address is uniformly +100$_H$ irrespective of the priority order.

6.2.2 FE Level Non-Maskable Interrupt Sources

6.2.2.1 WDTNMIF — FENMI Factor Register

This register contains information about which interrupt has generated the FE level non-maskable interrupt (FENMI). This register is initialized by any reset.

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

Address: FFF8 0200$_H$

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 6.5 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

6.2.2.2 WDTNMIFC — WDTNMI Factor Clear Register

This register clears the FE level non-maskable interrupt flags of the WDTNMIF register.

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

Address: FFF8 0208$_H$

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 6.6 WDTNMIFC Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When writing to these bits, write the value after reset.
2	WDTA1NMIFC	WDTA1NMIF flag clear 0: — 1: WDTA1NMIF is cleared.
1	WDTA0NMIFC	WDTA0NMIF flag clear 0: — 1: WDTA0NMIF is cleared.
0	TNMIFC	TNMIF flag clear 0: — 1: TNMIF is cleared.

6.2.3 FE Level Maskable Interrupt Sources

6.2.3.1 FEINTF — FEINT Factor Register

This register contains information about which interrupt has generated the FE level maskable interrupt (FEINT). This register is initialized by any reset.

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

Address: FFF8 0300$_H$

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
	LVIH FEIF	OSTM0 FEIF	ECCRAM FEIF	ECCSD FLI0 FEIF	—	ECCDC SIH3 FEIF	ECCDC SIH2 FEIF	ECCDC SIH1 FEIF	ECCDC SIH0 FEIF	ECCDC NRAM FEIF	—	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 6.7 FEINTF Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	LVIHFEIF	NTLVIH 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	ECCSDFLI0 FEIF	INTECCSDFLI0 interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
11	Reserved	When read, the value after reset is returned.
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	ECCDCNRAM FEIF	INTECCDCNRAM interrupt occurrence 0: No interrupt occurred 1: Interrupt has occurred
5	Reserved	When read, the value after reset is returned.

Table 6.7 FEINTF Register Contents (2/2)

Bit Position	Bit Name	Function
4	ECCDEEP0FEIF	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

6.2.3.2 FEINTFMSK — FEINT Factor Mask Register

This register masks the FE level maskable interrupt (FEINT). This register is initialized by any reset.

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

Address: FFF8 0304$_H$

Value after reset: FFFF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	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
	LVIHFEIFMSK	OSTM0FEIFMSK	ECCRAMFEIFMSK	ECCSDFLI0FEIFMSK	—	ECCDCSIH3FEIFMSK	ECCDCSIH2FEIFMSK	ECCDCSIH1FEIFMSK	ECCDCSIH0FEIFMSK	ECCDCNRAMFEIFMSK	—	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	R/W	R	R	R	R/W

Table 6.8 FEINTFMSK Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
15	LVIHFEIFMSK	INTLVIH interrupt mask 0: Not masked 1: Masked
14	OSTM0FEIFMSK	INTOSTM0_FE interrupt mask 0: Not masked 1: Masked
13	ECCRAMFEIFMSK	INTECCRAM interrupt mask 0: Not masked 1: Masked
12	ECCSDFLI0FEIFMSK	INTECCSDFLI0 interrupt mask 0: Not masked 1: Masked
11	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
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	ECCDCNRAMFEIFMSK	INTECCDCNRAM interrupt mask 0: Not masked 1: Masked
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4	ECCDEEP0FEIFMSK	INTECCDEEP0 interrupt mask 0: Not masked 1: Masked

Table 6.8 FEINTFMSK Register Contents (2/2)

Bit Position	Bit Name	Function
3 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	LVILFEIFMSK	INTLVIL interrupt mask 0: Not masked 1: Masked

6.2.3.3 FEINTFC — FEINT Factor Clear Register

This register clears the bits of the FEINT factor register (FEINTF).

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

Address: FFF8 0308$_H$

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
	LVIH FEIFC	OSTM0 FEIFC	ECC RAM FEIFC	ECCSD FLI0 FEIFC	—	ECCDC SIH3 FEIFC	ECCDC SIH2 FEIFC	ECCDC SIH1 FEIFC	ECCDC SIH0 FEIFC	ECCDC NRAM FEIFC	—	ECC DEEP0 FEIFC	—	—	—	LVIL FEIFC
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	W	W	W	W	R	W	W	W	W	W	R	W	R	R	R	W

Table 6.9 FEINTFC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing to these bits, write 0.
15	LVIHFEIFC	LVIHFEIF flag clear 0: — 1: Cleared
14	OSTM0 FEIFC	OSTM0FEIF flag clear 0: — 1: Cleared
13	ECCRAM FEIFC	ECCRAMFEIF flag clear 0: — 1: Cleared
12	ECCSDFLI0 FEIFC	ECCSDFLI0FEIF flag clear 0: — 1: Cleared
11	Reserved	When writing to this bit, write 0.
10	ECCDCSIH3 FEIFC	ECCDCSIH3FEIF flag clear 0: — 1: Cleared
9	ECCDCSIH2 FEIFC	ECCDCSIH2FEIF flag clear 0: — 1: Cleared
8	ECCDCSIH1 FEIFC	ECCDCSIH1FEIF flag clear 0: — 1: Cleared
7	ECCDCSIH0 FEIFC	ECCDCSIH0FEIF flag clear 0: — 1: Cleared
6	ECCDCNRAM FEIFC	ECCDCNRAMFEIFC flag clear 0: — 1: Cleared
5	Reserved	When writing to this bit, write 0.
4	ECCDEEP0 FEIFC	ECCDEEP0FEIF flag clear 0: — 1: Cleared
3 to 1	Reserved	When writing to these bits, write 0.

Table 6.9 FEINTFC Register Contents (2/2)

Bit Position	Bit Name	Function
0	LVILFEIFC	LVILFEIF flag clear 0: — 1: Cleared

6.3 Edge Detection

The external interrupts (TNMI and INTPm) can be configured to generate an interrupt request upon a rising or falling edge, upon both edges, or upon low or high level of the external pin.

The following registers are used to specify the edge and level of each interrupt:

Table 6.10 External Interrupt Edge 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_INTPLH
INTP9	FCLA0CTL1_INTPLH
INTP10	FCLA0CTL2_INTPH
INTP11	FCLA0CTL3_INTPH
INTP12	FCLA0CTL4_INTPH
INTP13	FCLA0CTL5_INTPLH
INTP14	FCLA0CTL6_INTPLH
INTP15	FCLA0CTL7_INTPLH

See **Section 2, Pin Function** for details of these registers.

6.4 Interrupt Controller Control Registers

Writing to the ICxxx, IMRm (m = 0 to 8), FNC, and FIC registers is enabled only in supervisor mode (PSW.UM = 0).

6.4.1 ICxxx — EI Level Interrupt Control Registers

These registers, each of which is for a channel of EI level maskable interrupt (INT), are used to set the conditions to control each channel. This register is initialized by any reset.

Access: ICxxx can be read/written in 16-bit units.
ICxxxH and ICxxxL can be read/written in 8- or 1-bit units.
Access to the bit 14, 13, 11 to 8, 5 to 3 by using a SET1, CLR1, or NOT1 instruction is prohibited.

Address: See **Table 6.4, RH850/F1L EI Level Maskable Interrupt Sources**.

Value after reset: 0087_H (edge detection), 8087_H (high level detection)*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CTxxx	—	—	RFxxx	—	—	—	—	MKxxx	TBxxx	—	—	—	P2xxx	P1xxx	P0xxx
Value after reset	0/1*1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	1
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R	R	R	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 6.4, RH850/F1L EI Level Maskable Interrupt Sources**.

Table 6.11 ICxxx Register Contents (1/2)

Bit Position	Bit Name	Function
15	CTxxx	This bit indicates an interrupt detection. This bit is read only. 0: Detection of the edge is currently selected. 1: Detection of the level is currently selected.
14, 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
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 (INTn), just as when an interrupt request is acknowledged. 0: No interrupt request is made. 1: Interrupt request is made. **Input Interface** / **Operation** Detection in synchronization with an edge (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. Detection of the high level (CTxxx = 1) / This bit cannot be set or cleared by software. It is readable only. It is not cleared when an interrupt request is acknowledged by the CPU core.
11 to 8	Reserved	When read, the value after reset is returned. When writing to these bits, 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, therefore, 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

Table 6.11 ICxxx Register Contents (2/2)

Bit Position	Bit Name	Function
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: Reference to a table For details on the way to determine the interrupt vector, see the RH850 Family Users' Manual: Software.
5 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2 to 0	P2xxx to P0xxx	These bits specify the interrupt priority as one of 8 levels, with 0 as the highest and 7 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 P2xxx 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 6.4, RH850/F1L EI Level Maskable Interrupt Sources** and of the channels which are not incorporated in the product.

6.4.2 IMRm — EI Level Interrupt Mask Registers (m = 0 to 8)

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. This setting for IMRm is reflected in the corresponding MKxxx bit. This register is initialized by any reset.

Access: IMRm can be read/written in 32-bit units.
IMRmH and IMRmL can be read/written in 16-bit units.
MRmHH, IMRmHL, IMRmLH, and IMRmLL can be read/written in 8- or 1-bit units.

Address: IMR0: FFFF $90F0_H$
IMRm (m = 1 to 8): FFFF $A400_H$ + (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 6.12 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 (INT) channels 0 to 287. 0: Enables interrupt servicing 1: Disables interrupt servicing

CAUTION

MKxxx bits which correspond to channels listed as “Reserved” in **Table 6.4, RH850/F1L EI Level Maskable Interrupt Sources**, and to channels which are not incorporated in the product must be set to “1”.

6.4.3 FNC — FE Level NMI Status Register

This register indicates the status of an FE level non-maskable interrupt (FENMI).

Access: FNC can be read in 16-bit units.
FNCH can be read in 8- or 1-bit units.

Address: FNC: FFFF 9078$_H$
FNCH: FFFF 9079$_H$

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 6.13 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.

6.4.4 FIC — FE Level Maskable Interrupt Status Register

This register indicates the status of an FE level maskable interrupt (FEINT).

Access: FIC can be read in 16-bit units.
FICH can be read in 8- or 1-bit units.

Address: FIC: FFFF 907A$_H$
FICH: FFFF 907B$_H$

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 6.14 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 is readable only.
11 to 0	Reserved	When read, the value after reset is returned.

6.5 EI Level Maskable Interrupt Select Registers

The following registers are used to select an EI level maskable interrupt.

6.5.1 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 6.4, RH850/F1L EI Level Maskable Interrupt Sources**.

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

Address: FFBC 0300_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_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	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 6.15 SELB_INTC1 Register Contents (1/2)

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
11	SELB_INTC1_11	Interrupt channel selection 0: INTTAUD0I14 (Channel 7) INTCSIH3IJC (Channel 153) 1: INTCSIH3IJC_1 (Channel 7) INTTAUD0I14_2 (Channel 153)
10	SELB_INTC1_10	Interrupt channel selection 0: INTTAUD0I12 (Channel 6) INTCSIH3IRE (Channel 152) 1: INTCSIH3IRE_1 (Channel 6) INTTAUD0I12_2 (Channel 152)
9	SELB_INTC1_9	Interrupt channel selection 0: INTTAUD0I10 (Channel 5) INTCSIH3IR (Channel 151) 1: INTCSIH3IR_1 (Channel 5) INTTAUD0I10_2 (Channel 151)
8	SELB_INTC1_8	Interrupt channel selection 0: INTTAUD0I2 (Channel 1) INTCSIH3IC (Channel 150) 1: INTCSIH3IC_1 (Channel 1) INTTAUD0I2_2 (Channel 150)
7	SELB_INTC1_7	Interrupt channel selection 0: INTP2 (Channel 31) INTCSIH2IJC (Channel 127) 1: INTCSIH2IJC_1 (Channel 31) INTP2_2 (Channel 127)
6	SELB_INTC1_6	Interrupt channel selection 0: INTP1 (Channel 30) INTCSIH2IRE (Channel 126) 1: INTCSIH2IRE_1 (Channel 30) INTP1_2 (Channel 126)

Table 6.15 SELB_INTC1 Register Contents (2/2)

Bit Position	Bit Name	Function
5	SELB_INTC1_5	Interrupt channel selection 0: INTP0 (Channel 29) INTCSIH2IR (Channel 125) 1: INTCSIH2IR_1 (Channel 29) INTP0_2 (Channel 125)
4	SELB_INTC1_4	Interrupt channel selection 0: INTTAUD0I0 (Channel 0) INTCSIH2IC (Channel 124) 1: INTCSIH2IC_1 (Channel 0) INTTAUD0I0_2 (Channel 124)
3	SELB_INTC1_3	Interrupt channel selection 0: INTCSIG0IR (Channel 20) INTCSIH1IJC (Channel 111) 1: INTCSIH1IJC_1 (Channel 20) INTCSIG0IR_2 (Channel 111)
2	SELB_INTC1_2	Interrupt channel selection 0: INTCSIG0IC (Channel 19) INTCSIH1IRE (Channel 110) 1: INTCSIH1IRE_1 (Channel 19) INTCSIG0IC_2 (Channel 110)
1	SELB_INTC1_1	Interrupt channel selection 0: INTTAPA0IVLY0 (Channel 9) INTCSIH1IR (Channel 109) 1: INTCSIH1IR_1 (Channel 9) INTTAPA0IVLY0_2 (Channel 109)
0	SELB_INTC1_0	Interrupt channel selection 0: INTTAPA0IPEK0 (Channel 8) INTCSIH1IC (Channel 108) 1: INTCSIH1IC_1 (Channel 8) INTTAPA0IPEK0_2 (Channel 108)

CAUTION

The operation of peripheral functions should be enabled after setting the corresponding interrupt source by SELB_INTC1.

6.5.2 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/written in 16-bit units.

Address: FFBC 0304_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 6.16 SELB_INTC2 Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When writing to these bits, write the value after reset.
10	SELB_INTC2_10	Interrupt channel 149 selection 0: INTTAUB0I15 1: INTPWGA31
9	SELB_INTC2_9	Interrupt channel 147 selection 0: INTTAUB0I13 1: INTPWGA30
8	SELB_INTC2_8	Interrupt channel 145 selection 0: INTTAUB0I11 1: INTPWGA26
7	SELB_INTC2_7	Interrupt channel 143 selection 0: INTTAUB0I9 1: INTPWGA19
6	SELB_INTC2_6	Interrupt channel 141 selection 0: INTTAUB0I7 1: INTPWGA18
5	SELB_INTC2_5	Interrupt channel 139 selection 0: INTTAUB0I5 1: INTPWGA17
4	SELB_INTC2_4	Interrupt channel 137 selection 0: INTTAUB0I3 1: INTPWGA16
3	SELB_INTC2_3	Interrupt channel 80 selection 0: INTENCA0I1 1: INTPWGA7
2	SELB_INTC2_2	Interrupt channel 79 selection 0: INTENCA0I0 1: INTPWGA6
1	SELB_INTC2_1	Interrupt channel 78 selection 0: INTENCA0IUD 1: INTPWGA5
0	SELB_INTC2_0	Interrupt channel 77 selection 0: INTENCA0IOV 1: INTPWGA4

CAUTION

After setting each interrupt source by SELB_INTC2, enable the operation of the corresponding peripheral functions.

6.6 Interrupt Function System Registers

See **Section 3.3.3, Interrupt Function Registers**.

6.6.1 ISPR — Priority of Interrupt being Serviced

See **Section 3.3.3.1, ISPR — Priority of Interrupt being Serviced**.

6.6.2 PMR — Interrupt Priority Masking

See **Section 3.3.3.2, PMR — Interrupt Priority Masking**.

6.6.3 ICSR — Interrupt Control Status

See **Section 3.3.3.3, ICSR — Interrupt Control Status**.

6.6.4 INTCFG — Interrupt Function Setting

See **Section 3.3.3.4, INTCFG — Interrupt Function Setting**.

6.7 Operation when Acknowledging an Interrupt

Check whether each exception that is reported during instruction execution is acknowledged according to the priority. The procedure for acknowledgment operation of each interrupt is shown below.

(1) Check whether the acknowledgment conditions are satisfied and whether exceptions 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 code in the FEIC.
- Update the PSW and MCTL.*2
- Store the exception handler address calculated in (b) 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 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 6.10, Exception Handler Address**.

Note 2. For the values to be updated, see Table 6.1 List of Exception Sources in the *RH850 Family Users' Manual: Software*.

The following figure shows steps (1) to (4).

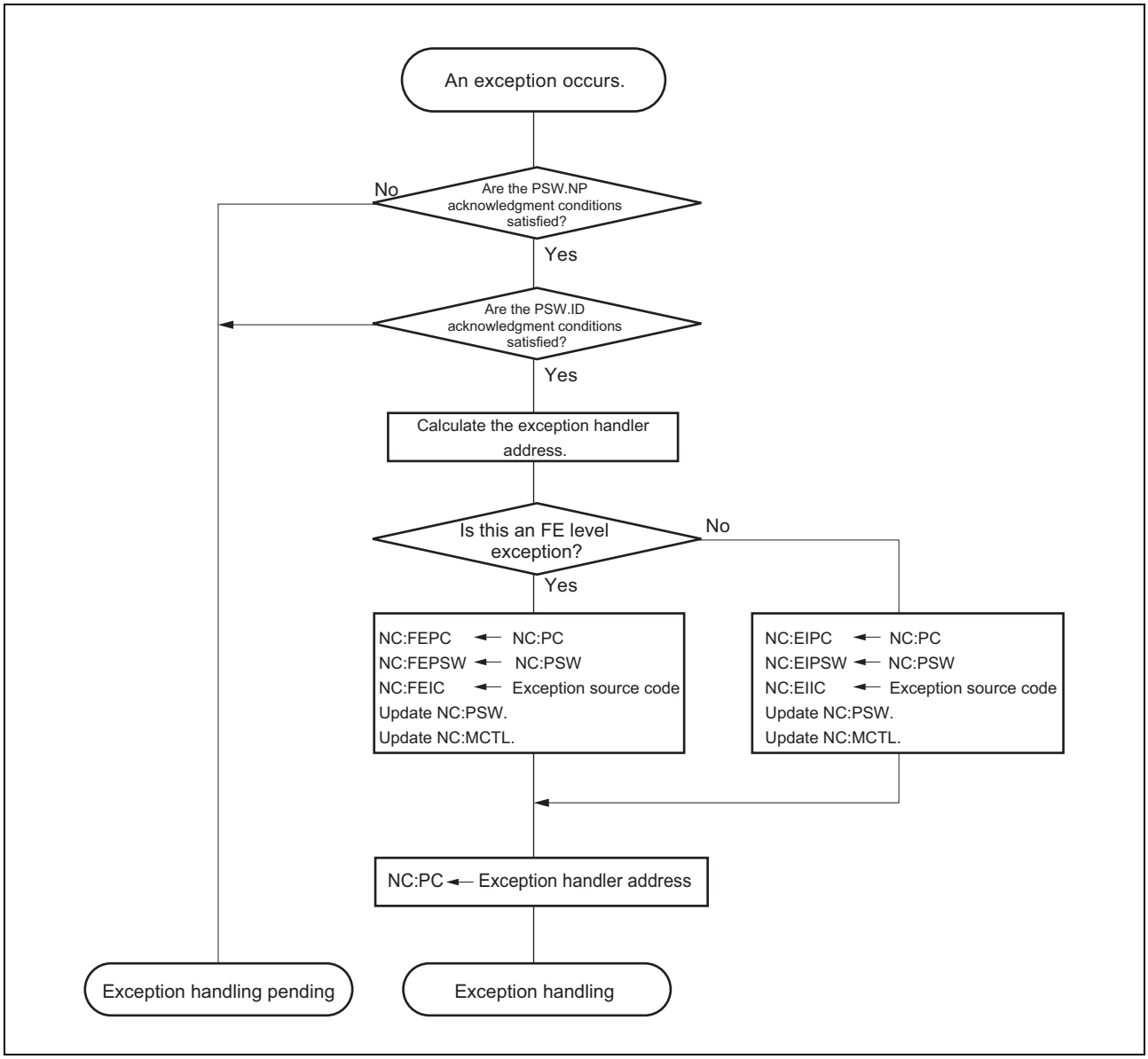


Figure 6.2 Processing upon Occurrence of FENMI Interrupt Request

6.8 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) Return PC and PSW are loaded from the EIPC and EIPSW 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.

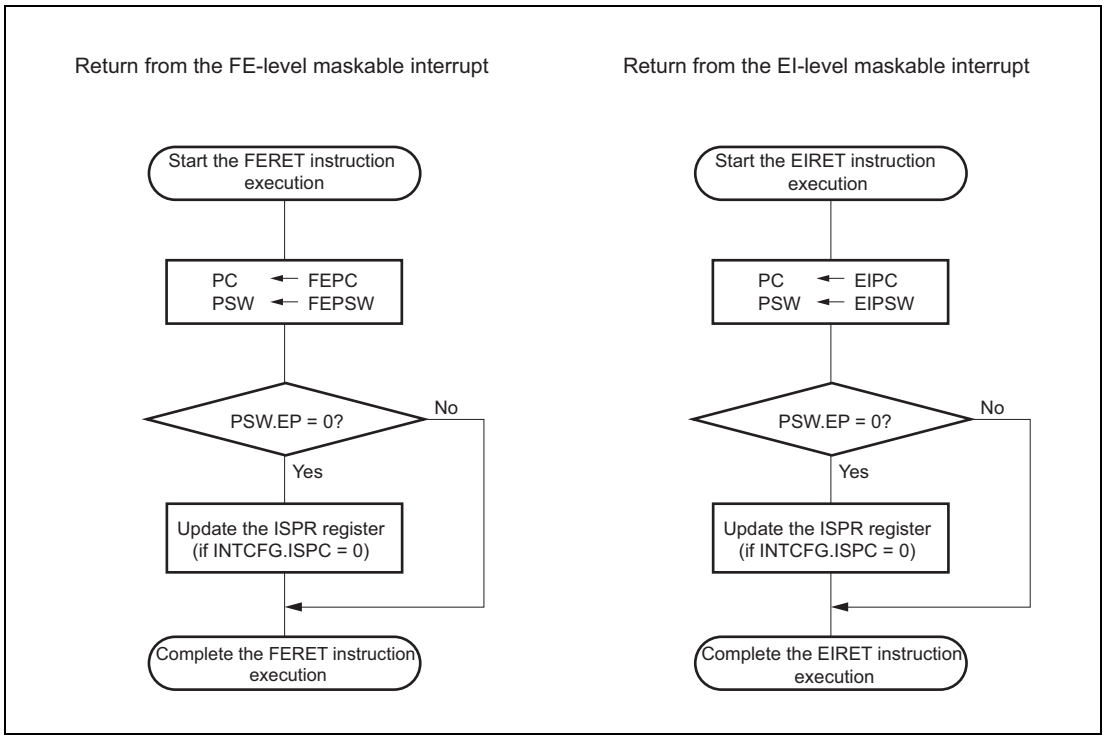


Figure 6.3 Flow of Return from Interrupts

6.9 Interrupt Operation

6.9.1 Interrupt Mask Function of EI Level Maskable Interrupt (INTn)

Interrupt masking can be specified for each respective interrupt channel of INTn. Interrupt masking is performed by the following register settings.

Table 6.17 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 MKn 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.EIMKn bit, interrupts are prohibited for the corresponding channel.

(2) When the corresponding ICxxx.MKxxx bit is read, a 1 is returned.

CAUTION

If the MKxxx bit is set to 0 while an interrupt request is withheld (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 remove an interrupt request that is withheld, clear the corresponding RFxxx bit in software.

6.9.2 Interrupt Priority Level Judgment

When FE level non-maskable interrupts (FENMI), FE level maskable interrupts (FEINT), and EI level maskable interrupts (INT) are input, priorities including other exceptions are determined, and the exception with the highest priority (including interrupts) is requested. Exceptions requested 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 INT interrupts are as follows.

FENMI > FEINT > INT

See the *RH850 Family Users' Manual: Software* for other exceptions.

For INTn interrupts, the priority level can be set independently for each interrupt source. Specify the priority level with the bits P2xxx to P0xxx. The interrupt priority levels can be set from 0 to 7: 0 is the highest and 7 the lowest. Among multiple INTn interrupts with the same priority level, the interrupt with the lowest interrupt channel number has priority.

Table 6.18 Example of INTn Interrupt Priority Level Settings and Priority Levels

INTn	ICxxx.P[2:0]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 has been executed. When multiple INT interrupts are requested at the same time, the interrupt to be acknowledged is determined by the following procedure.

6.9.2.1 Comparison with the Priority Level as 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 shown in the ISPR register.

Interrupts with a higher priority level than the interrupt currently being handled proceed to the next priority judgment stage.

6.9.2.2 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 **Section 3.3.3.2, PMR — Interrupt Priority Masking**, or the *RH850 Family Users' Manual: Software*.

6.9.2.3 The Requested Interrupt Source with the Highest Priority Level is Selected

When interrupts are being simultaneously requested from multiple sources, the interrupt source from the highest priority level, with the smallest interrupt channel number is selected.

6.9.2.4 Interrupt Hold by CPU

Interrupt acknowledgment is pending according to the state of the NP and ID bits of the PSW register. At this time, priority judgment among INT interrupts, and priority judgment among INT, FEINT and FENMI interrupts is performed even while interrupt acknowledgment is pending, and the interrupt with the highest priority is selected upon realization of the acknowledgment condition.

Example

An INT 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 INT interrupt with the priority level 3 is requested and the PSW.ID bit is cleared to 0, the latter INT interrupt (with the priority level 3) will be generated.

Figure 6.4 shows an example of multiple interrupt handling when another interrupt request is acknowledged while an interrupt processing has been 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.

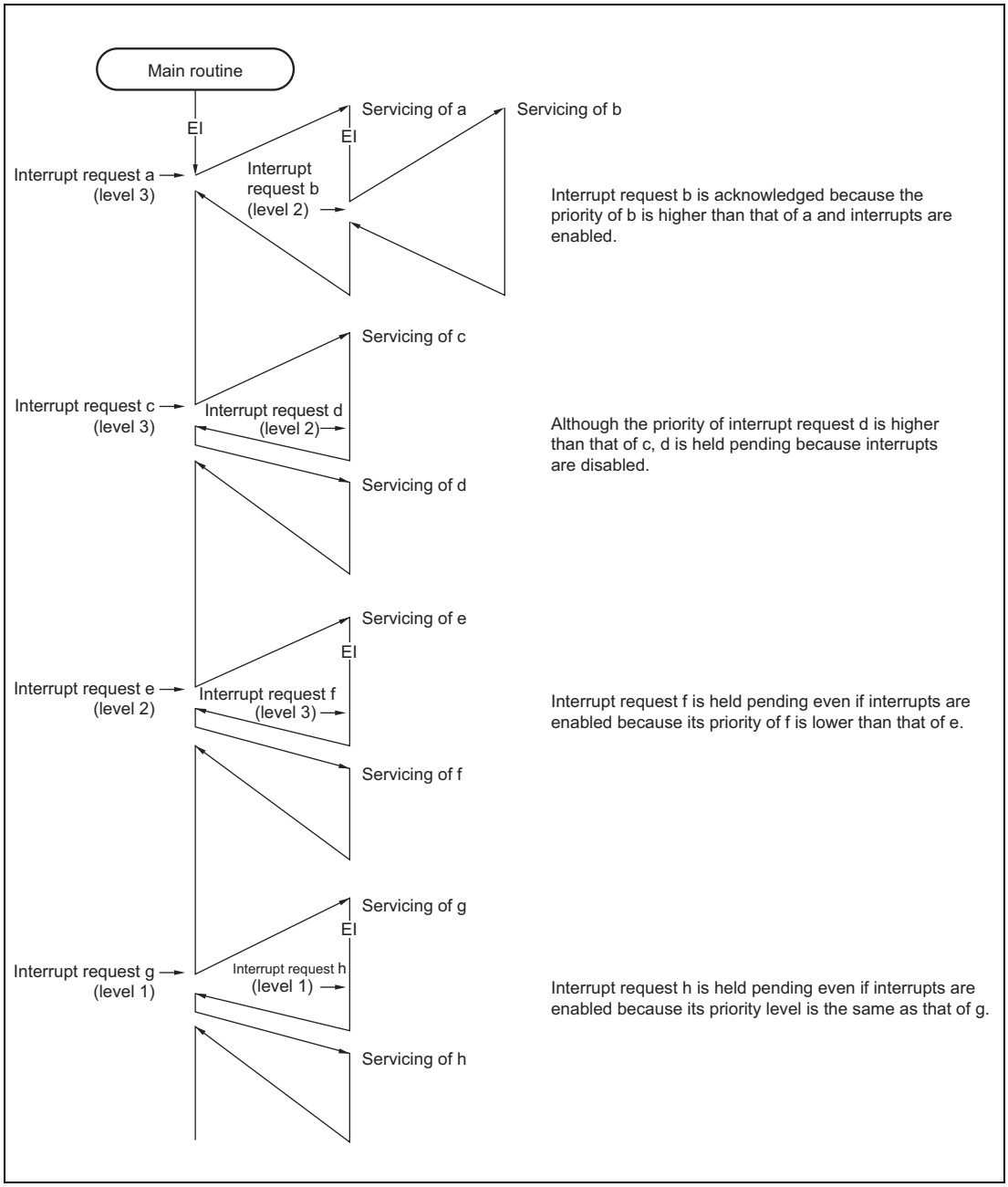
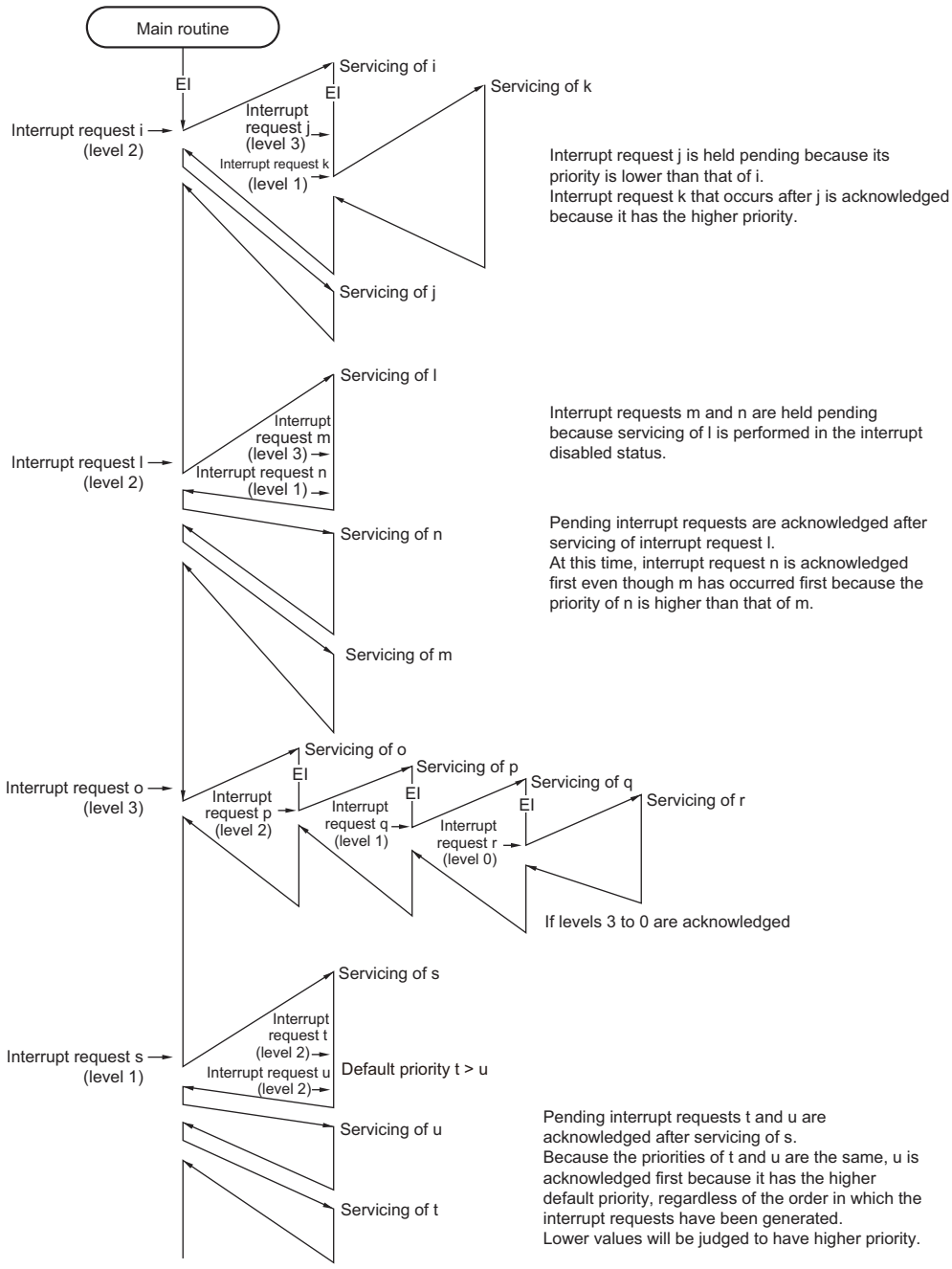


Figure 6.4 Example of Processing in which an Interrupt Request Signal is Issued while Another Interrupt is being Handled (1)



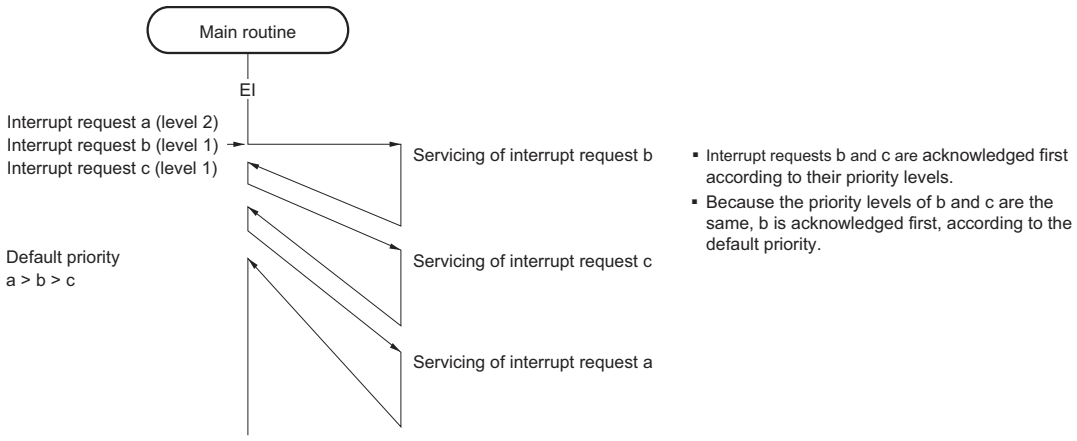
Note 1. “a” to “u” in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

Note 2. The default priority in the figure indicates the relative priority between two interrupt request signals.

Figure 6.5 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 in the figure indicates the relative priority between two interrupt request signals.

Figure 6.6 Example of Servicing Simultaneously Generated Interrupt Requests

6.9.3 Interrupt Request Acknowledgement Conditions and the Priority

See the *RH850 Family User's Manual: Software*.

6.9.4 Exception Priority of Interrupts and the Priority Mask

See the *RH850 Family User's Manual: Software*.

6.9.5 Interrupt Priority Mask

See the *RH850 Family User's Manual: Software*.

6.9.6 Priority Mask Function

The priority mask function prohibits in batch INT interrupts of the specified interrupt priority levels.

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 6.19

PMR.PMRm	Operation
0	Acknowledges requests from priority level m interrupt source.
1	Masks requests from priority level m interrupt source.

Note: m = 0 to 7

The presence of INT interrupts held pending with this function can be checked with **Section 6.9.7, Exception Management**.

For details on the PMR register, see **Section 3.3.3.2, PMR — Interrupt Priority Masking**, or the *RH850 Family User's Manual: Software*.

6.9.7 Exception Management

The suspended interrupt can be checked in the RH850/F1L. For details, see the *RH850 Family User's Manual: Software*.

6.10 Exception Handler Address

For the RH850/F1L the exception handler address used for execution during reset input, exception acknowledgment, or interrupt acknowledgment can be changed according to the settings.

The exception handler address for reset and exception (including interrupt) 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 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 (INTn). In the RH850/F1L, one exception handler address is assigned to each interrupt priority. Consequently, interrupts that have the same priority level branch to the same exception handler.

6.10.1 Direct Vector Method

The CPU uses the result of adding the offset shown in **Table 6.20, 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 in 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 6.20, 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 value after updated.

Note 2. The exceptions that always see the RBASE register are determined according to the hardware specifications.

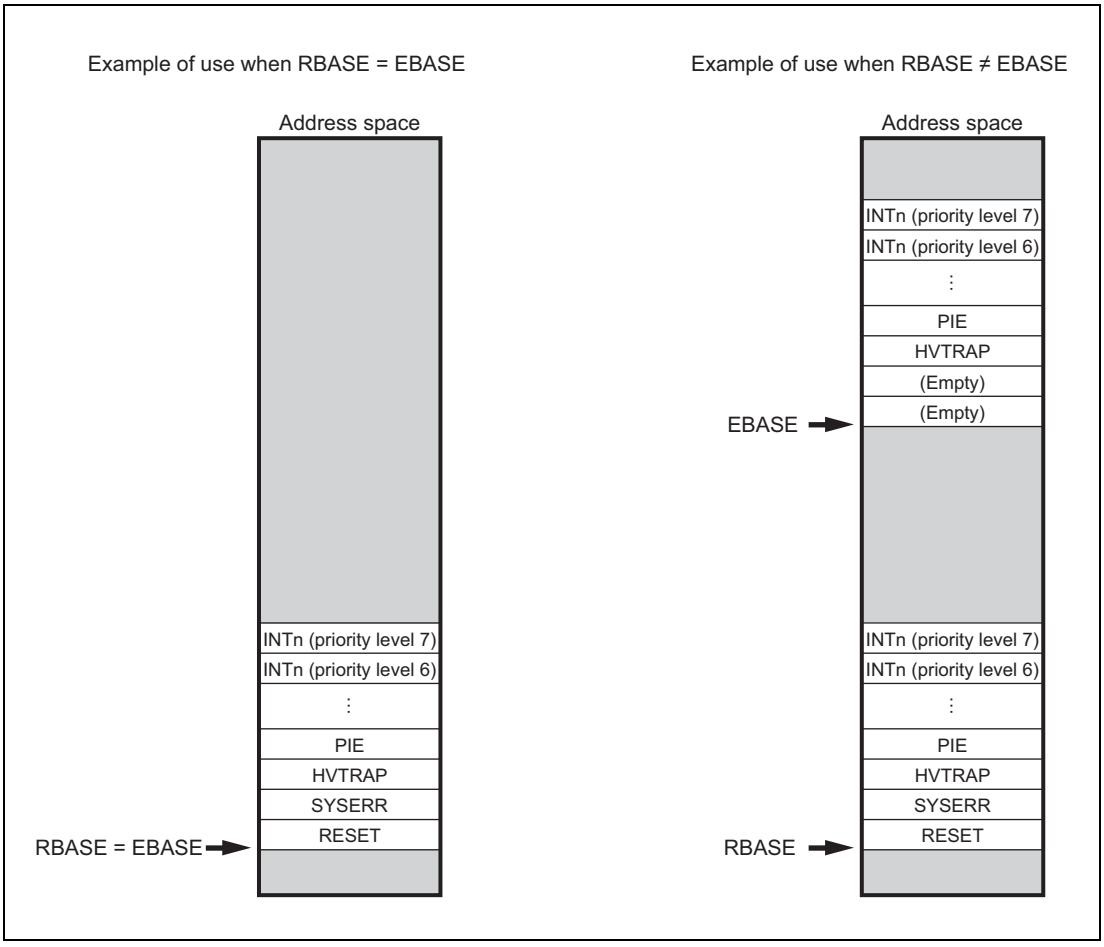


Figure 6.7 Direct Vector Method

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 on the basis of the value after updated by the acknowledgment of an exception.

Table 6.20 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			010_H	010_H
HVTRAP		EBASE	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/ITLBE/DTLBE			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	

Base register selection is used to execute the exception handling for reset 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.

6.10.2 Table Reference Method

In 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/F1L defines the table reference method for interrupts that assume the above usage.

For the table reference method, 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 6.21 shows the exception handler address read positions corresponding to each interrupt channel and **Figure 6.8** shows an overview of the placement in memory.

Table 6.21 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 286	INTBP register value + 286 × 4
EI level maskable interrupt channel 287	INTBP register value + 287 × 4

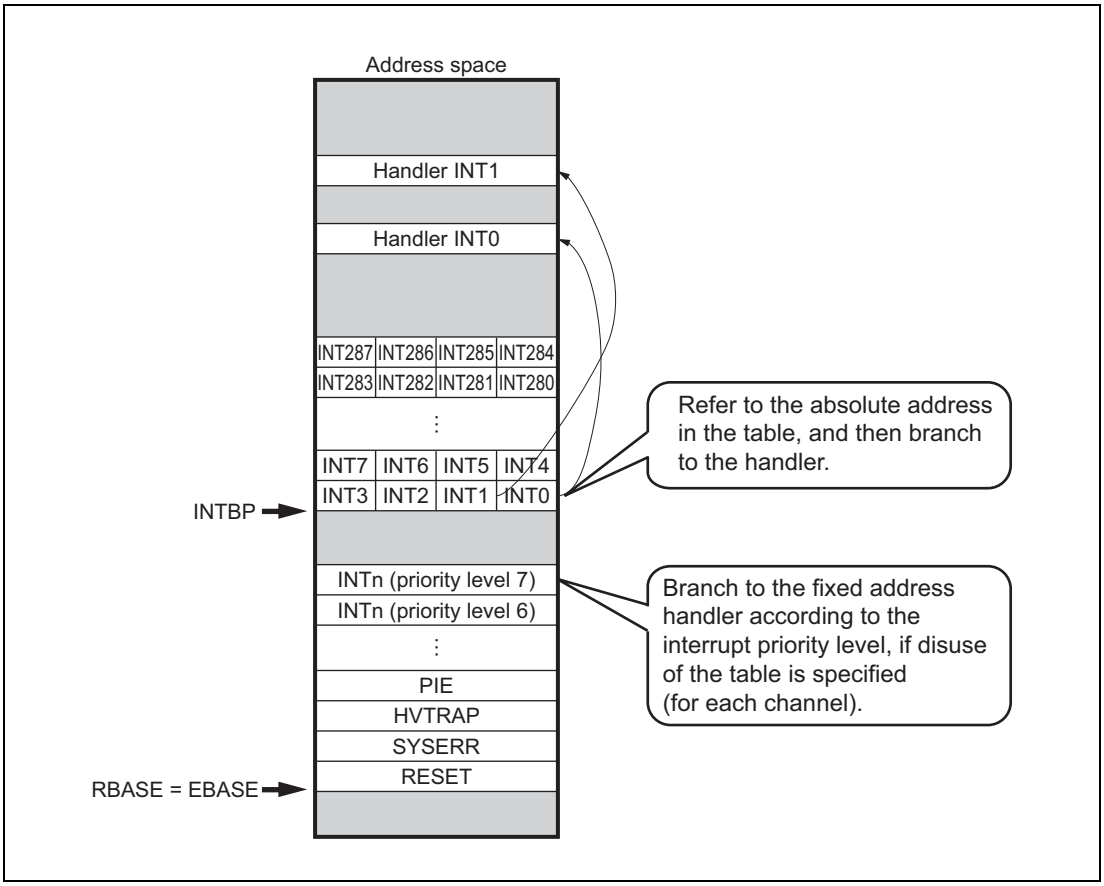


Figure 6.8 Overview of the Table Reference Method

Section 7 DMA

This section contains a generic description of the DMA controller (DMA).

The first part in this section describes the features specific to the RH850/F1L, including the number of channels and register base addresses. The ensuing sections describe the DMA functions and registers.

7.1 Features of RH850/F1L DMA

7.1.1 Number of Channels

The products of the RH850/F1L series incorporate a DMA with the number of channels indicated below.

Table 7.1 Number of Channels

Product	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of channels	16 ch (8 ch + 8 ch)	16 ch (8 ch + 8 ch)	16 ch (8 ch + 8 ch)	16 ch (8 ch + 8 ch)	16 ch (8 ch + 8 ch)	16 ch (8 ch + 8 ch)

Table 7.2 Index

Index	Meaning
m	In this section, the DMA channels are identified with an index “m”. For example, the DMA source address register is denominated DSAm.

7.1.2 Register Base Addresses

The DMA base addresses are indicated in the following table.

The DMA register addresses are expressed as an offset of the base address.

Table 7.3 Register Base Addresses

Base Address Name	Base Address
<DMA_base>	FFFF 8300_H

7.1.3 Interrupt Requests

The DMAC interrupt requests are listed in the following table.

Table 7.4 Interrupt Requests

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number
INTDMA0	DMA0 transfer completion	52	—
INTDMA1	DMA1 transfer completion	53	—
INTDMA2	DMA2 transfer completion	54	—
INTDMA3	DMA3 transfer completion	55	—
INTDMA4	DMA4 transfer completion	56	—
INTDMA5	DMA5 transfer completion	57	—
INTDMA6	DMA6 transfer completion	58	—
INTDMA7	DMA7 transfer completion	59	—
INTDMA8	DMA8 transfer completion	60	—
INTDMA9	DMA9 transfer completion	61	—
INTDMA10	DMA10 transfer completion	62	—
INTDMA11	DMA11 transfer completion	63	—
INTDMA12	DMA12 transfer completion	64	—
INTDMA13	DMA13 transfer completion	65	—
INTDMA14	DMA14 transfer completion	66	—
INTDMA15	DMA15 transfer completion	67	—

Table 7.5 Interrupt Requests (System Error Interrupts)

Unit Interrupt Name	Description	Interrupt Number	DMA Trigger Number
—	DMA transfer error occurrence	SYSERR	—

7.1.4 DMA Trigger Factors

DMA trigger factors can be selected by setting the DTFRm.DTFRmIFCm[5:0] bits.

The following table lists all DMA trigger factors which can be selected by the DTFRm register.

Table 7.6 DMA Trigger Factors (m = 0 to 7) (1/2)

DMA Trigger Number DTFRm.DTFRmIFCm[5:0]	DMA Trigger Interrupt	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
0	INTTAUD0I0	√	√	√	√	√	√
1	INTTAUD0I4	√	√	√	√	√	√
2	INTTAUD0I8	√	√	√	√	√	√
3	INTTAUD0I12	√	√	√	√	√	√
4	INTADCA0I0	√	√	√	√	√	√
5	INTADCA0I1	√	√	√	√	√	√
6	INTADCA0I2	√	√	√	√	√	√
7	ADC_CONV_END0	√	√	√	√	√	√
8	INTCSIG0IC	√	√	√	√	√	√
9	INTCSIG0IR	√	√	√	√	√	√
10	INTRLIN30UR0	√	√	√	√	√	√
11	INTRLIN30UR1	√	√	√	√	√	√
12	INTP0	√	√	√	√	√	√
13	INTP2	√	√	√	√	√	√
14	INTP4	√	√	√	√	√	√
15	INTTAUD0I1	√	√	√	√	√	√
16	INTTAUD0I5	√	√	√	√	√	√
17	INTTAUD0I9	√	√	√	√	√	√
18	INTTAUD0I13	√	√	√	√	√	√
19	INTRIIC0TI	√	√	√	√	√	√
20	INTRIIC0RI	√	√	√	√	√	√
21	INTTAUJ0I0	√	√	√	√	√	√
22	INTTAUJ0I3	√	√	√	√	√	√
23	Setting prohibited	—	—	—	—	—	—
24	Setting prohibited	—	—	—	—	—	—
25	Setting prohibited	—	—	—	—	—	—
26	Setting prohibited	—	—	—	—	—	—
27	Setting prohibited	—	—	—	—	—	—
28	INTCSIH1IC	—	—	√	√	√	√
29	INTCSIH1IR	—	—	√	√	√	√
30	INTCSIH1IJC	—	—	√	√	√	√
31	INTP6	—	—	√	√	√	√
32	INTP8	—	—	√	√	√	√
33	INTTAUB0I0	—	—	—	√	√	√
34	INTTAUB0I2	—	—	—	√	√	√
35	INTTAUB0I4	—	—	—	√	√	√
36	INTTAUB0I6	—	—	—	√	√	√
37	INTTAUB0I9	—	—	—	√	√	√
38	INTTAUB0I11	—	—	—	√	√	√

Table 7.6 DMA Trigger Factors (m = 0 to 7) (2/2)

DMA Trigger Number DTFRm.DTFRmIFCm[5:0]	DMA Trigger Interrupt	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
39	INTTAUB0I13	—	—	—	√	√	√
40	INTTAUB0I15	—	—	—	√	√	√
41	INTCSIH3IC	—	—	—	√	√	√
42	INTCSIH3IR	—	—	—	√	√	√
43	INTCSIH3IJC	—	—	—	√	√	√
44	INTRLIN32UR0	—	—	√	√	√	√
45	INTRLIN32UR1	—	—	√	√	√	√
46	INTTAUJ1I0	—	—	—	√	√	√
47	INTTAUJ1I2	—	—	—	√	√	√
48	Setting prohibited	—	—	—	—	—	—
49	Setting prohibited	—	—	—	—	—	—
50	INTRLIN34UR0	—	—	—	—	√	√
51	INTRLIN34UR1	—	—	—	—	√	√
52	INTTAUB1I0	—	—	—	—	—	√
53	INTTAUB1I2	—	—	—	—	—	√
54	INTTAUB1I4	—	—	—	—	—	√
55	INTTAUB1I6	—	—	—	—	—	√
56	INTTAUB1I9	—	—	—	—	—	√
57	INTTAUB1I11	—	—	—	—	—	√
58	INTTAUB1I13	—	—	—	—	—	√
59	INTTAUB1I15	—	—	—	—	—	√
60	Setting prohibited	—	—	—	—	—	—
61	Setting prohibited	—	—	—	—	—	—
62	Setting prohibited	—	—	—	—	—	—
63	Setting prohibited	—	—	—	—	—	—

Table 7.7 DMA Trigger Factors (m = 8 to 15) (1/2)

DMA Trigger Number DTFRm.DTFRmIFCm[5:0]	Dma Trigger Interrupt	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
0	INTTAUD0I2	√	√	√	√	√	√
1	INTTAUD0I6	√	√	√	√	√	√
2	INTTAUD0I10	√	√	√	√	√	√
3	INTTAUD0I14	√	√	√	√	√	√
4	Setting prohibited	—	—	—	—	—	—
5	Setting prohibited	—	—	—	—	—	—
6	INTCSIH0IC	√	√	√	√	√	√
7	INTCSIH0IR	√	√	√	√	√	√
8	INTCSIH0IJC	√	√	√	√	√	√
9	INTP1	√	√	√	√	√	√
10	INTP3	√	√	√	√	√	√
11	INTP5	√	√	√	√	√	√
12	INTTAUD0I3	√	√	√	√	√	√
13	INTTAUD0I7	√	√	√	√	√	√
14	INTTAUD0I11	√	√	√	√	√	√
15	INTTAUD0I15	√	√	√	√	√	√
16	INTTAUJ0I1	√	√	√	√	√	√
17	INTTAUJ0I2	√	√	√	√	√	√
18	Setting prohibited	—	—	—	—	—	—
19	Setting prohibited	—	—	—	—	—	—
20	Setting prohibited	—	—	—	—	—	—
21	INTDMAFL	√	√	√	√	√	√
22	INTRLIN31UR0	—	√	√	√	√	√
23	INTRLIN31UR1	—	√	√	√	√	√
24	INTP7	—	—	√	√	√	√
25	INTCSIH2IC	—	—	√	√	√	√
26	INTCSIH2IR	—	—	√	√	√	√
27	INTCSIH2IJC	—	—	√	√	√	√
28	INTTAUB0I1	—	—	—	√	√	√
29	INTTAUB0I3	—	—	—	√	√	√
30	INTTAUB0I5	—	—	—	√	√	√
31	INTTAUB0I7	—	—	—	√	√	√
32	INTTAUB0I8	—	—	—	√	√	√
33	INTTAUB0I10	—	—	—	√	√	√
34	INTTAUB0I12	—	—	—	√	√	√
35	INTTAUB0I14	—	—	—	√	√	√
36	INTTAUJ1I1	—	—	—	√	√	√
37	INTTAUJ1I3	—	—	—	√	√	√
38	INTP9	—	—	—	—	√	√
39	INTADCA1I0	—	—	—	—	√	√
40	INTADCA1I1	—	—	—	—	√	√
41	INTADCA1I2	—	—	—	—	√	√
42	ADC_CONV_END1	—	—	—	—	√	√

Table 7.7 DMA Trigger Factors (m = 8 to 15) (2/2)

DMA Trigger Number DTFRm.DTFRmIFCm[5:0]	Dma Trigger Interrupt	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
43	Setting prohibited	—	—	—	—	—	—
44	Setting prohibited	—	—	—	—	—	—
45	INTCSIG1IC	—	—	—	—	√	√
46	INTCSIG1IR	—	—	—	—	√	√
47	INTRLIN33UR0	—	—	—	√	√	√
48	INTRLIN33UR1	—	—	—	√	√	√
49	INTRLIN35UR0	—	—	—	—	√	√
50	INTRLIN35UR1	—	—	—	—	√	√
51	INTTAUB1I1	—	—	—	—	—	√
52	INTTAUB1I3	—	—	—	—	—	√
53	INTTAUB1I5	—	—	—	—	—	√
54	INTTAUB1I7	—	—	—	—	—	√
55	INTTAUB1I8	—	—	—	—	—	√
56	INTTAUB1I10	—	—	—	—	—	√
57	INTTAUB1I12	—	—	—	—	—	√
58	INTTAUB1I14	—	—	—	—	—	√
59	Setting prohibited	—	—	—	—	—	—
60	Setting prohibited	—	—	—	—	—	—
61	Setting prohibited	—	—	—	—	—	—
62	Setting prohibited	—	—	—	—	—	—
63	Setting prohibited	—	—	—	—	—	—

7.2 Overview

7.2.1 Functional Overview

7.2.1.1 DMA Transfer Function

DMA controller (DMAC) function

Registers to store transfer information (transfer address, transfer size, etc.) and registers to control DMAC are included.

Number of channels

2 units × 8 channels (Total: 16 channels)

Transfer data size

8 bits, 16 bits, 32 bits

Transfer data

- Little endian
- Misaligned data not supported

Maximum transfer count

32768 (2^{15}) times

Channel priority control

Fixed priority (highest priority (CH0) → lowest priority (CH15))

Subject to transfer

The relationship between the source and destination of transfer is shown below.

Table 7.8 The Relationship between the Source and Destination of Transfer

		Destination of Transfer				
		Peripheral I/O	Local RAM (Primary, Secondary), Retention RAM	Code Flash	Data Flash	External Memory
Source of Transfer	Peripheral I/O	√	√	—	—	√
	Local RAM (Primary, Secondary), Retention RAM	√	√	—	—	√
	Code Flash	√	√	—	—	√
	Data Flash	√	√	—	—	√
	External memory	√	√	—	—	√

Transfer type

2-cycle transfer (dual address transfer)

The address of both the transfer source and destination is accessed. Two bus cycles are required to execute one transfer (read cycle + write cycle). Because the bus is not locked between the read cycle and write cycle, a CPU cycle may interrupt between the read and write access.

Transfer mode

- Single transfer mode (when hardware DMA transfer request is generated)

  When a hardware DMA transfer request is generated, the bus mastership is acquired, and the bus is always released after transfer has been executed once. If another hardware DMA transfer request is generated after that, transfer is executed once again. This operation is repeated until transfer has been executed the number of times specified by the transfer count register (DTCm).

- Single-step transfer mode (when software DMA transfer request is generated)

  When a software DMA transfer request is generated, the bus mastership is acquired, and the bus is released each time transfer has been executed once. Once the software DMA transfer has completed, this operation is repeated until transfer has been executed the number of times specified by the transfer count register (DTCm).

Transfer address control

The address of the transfer source and destination can be incremented or fixed per channel.

Transfer error support

If any of the errors below occur during access for DMA transfer, DMA transfer is suspended and a SYSERR exception is output to the CPU.

- An 2-bit ECC error is generated in the primary local RAM, secondary local RAM, or retention RAM.
- An 2-bit ECC error is generated in the code flash.
- A peripheral function register not being supplied with a clock signal is accessed.

DMA transfer request

Any DMA transfer request from hardware or software DMA transfer can be selected for each channel (**Figure 7.5, Configuration Diagram of Transfer Requests of Channels**).

Software DMA transfer requests can be set by using the software (DTSm.DTSmSR bit = 1).

Status Bit (DTSm.DTSmDR bit) which indicates a hardware transfer request is available.

Transfer completion interrupt output function

This function outputs a transfer completion interrupt request signal (INTDMA15 to INTDMA0) when DMA transfer of each channel has been completed the number of times specified by the transfer count register (DTCm).

Stand-by support

DMA transfer is suspended momentarily by shifting to STOP mode.

DMA transfer suspend function

This function supports suspending DMA transfer by software.

Protection function

This function enables protection against writing to the DMA control register by setting an access master.

Loop function

At the completion of DMA transfer, this function controls the acceptance of the request for the next DMA transfer.

7.2.1.2 DTFR Function

The DMA trigger factor register (DTFR) selects DMA trigger factors from among interrupt request signals, and requests DMAC for DMA transfer. DTFRm (m = 15 to 0) registers are included for selecting the signals to be used for DMA transfer requests from among the 128 (64 × 2) input interrupt request signals.

Number of trigger factors

DMA transfer requests (for 16 channels) are selected from among 128 (64 × 2) interrupt request signals.

DMAC interface

The DMA transfer request signal m (m = 15 to 0) is output.
The DMA transfer request signal m is cleared by an acknowledge signal from DMA.

CPU interface

The last transfer signal from DMA is output as a CPU interrupt request signal.

Clearing of transfer request

A function that clears transfer request signals sent to DMA through register access is provided.

Confirmation of transfer request

A function that checks transfer request signals sent to DMA through register access is provided.

7.2.1.3 DMA Access Memory Map

See **Section 3.2.4, RH850/F1L Memory Map** for details.

7.2.1.4 Prioritization of Channels

When multiple DMA triggers occur simultaneously, bus access is determined by the following priority. Among DMAC, the priority is set as CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 > CH8 > CH9 > CH10 > CH11 > CH12 > CH13 > CH14 > CH15, in which CH0 has the highest priority.

7.2.2 Terms

The terms used in this section are defined as follows.

Table 7.9 Definition of Terms

Term	Function
DMA transfer	Period from the start of the first DMA cycle to assertion of INTDMAm
DMA cycle	Period of transferring one unit of data (A read cycle and a write cycle are generated to 2-cycle transfer and DMA Bus.)
Hardware DMA transfer request	DMA transfer request by the DMA trigger factors selected in the DTFRm register (It is possible to check whether there is any request from DTSm.DTSmDR bit.)
Software DMA transfer request	DAM transfer request by internal register (setting 1 to the DTSm.DTSmSR bit)
Read cycle	DMA cycle reading
Write cycle	DMA cycle writing
Single transfer	For DMAC, one DMA cycle is executed per transfer request. A single-transfer can be executed only by a hardware DMA transfer request.
Single step transfer	The number of transfers set in the transfer count setting register (DTCm) is executed per software DMA transfer request. Since the bus is released for each DMA cycle (after a read cycle or write cycle), the CPU is able to generate interrupts. If a higher-priority transfer request occurs during execution of a single step transfer, the single step transfer is suspended while the higher-priority transfer request is executed. (A single-step transfer can be executed only by a software DMA transfer request.)

7.2.3 Block Diagram

The following figure shows the main components of the DMA.

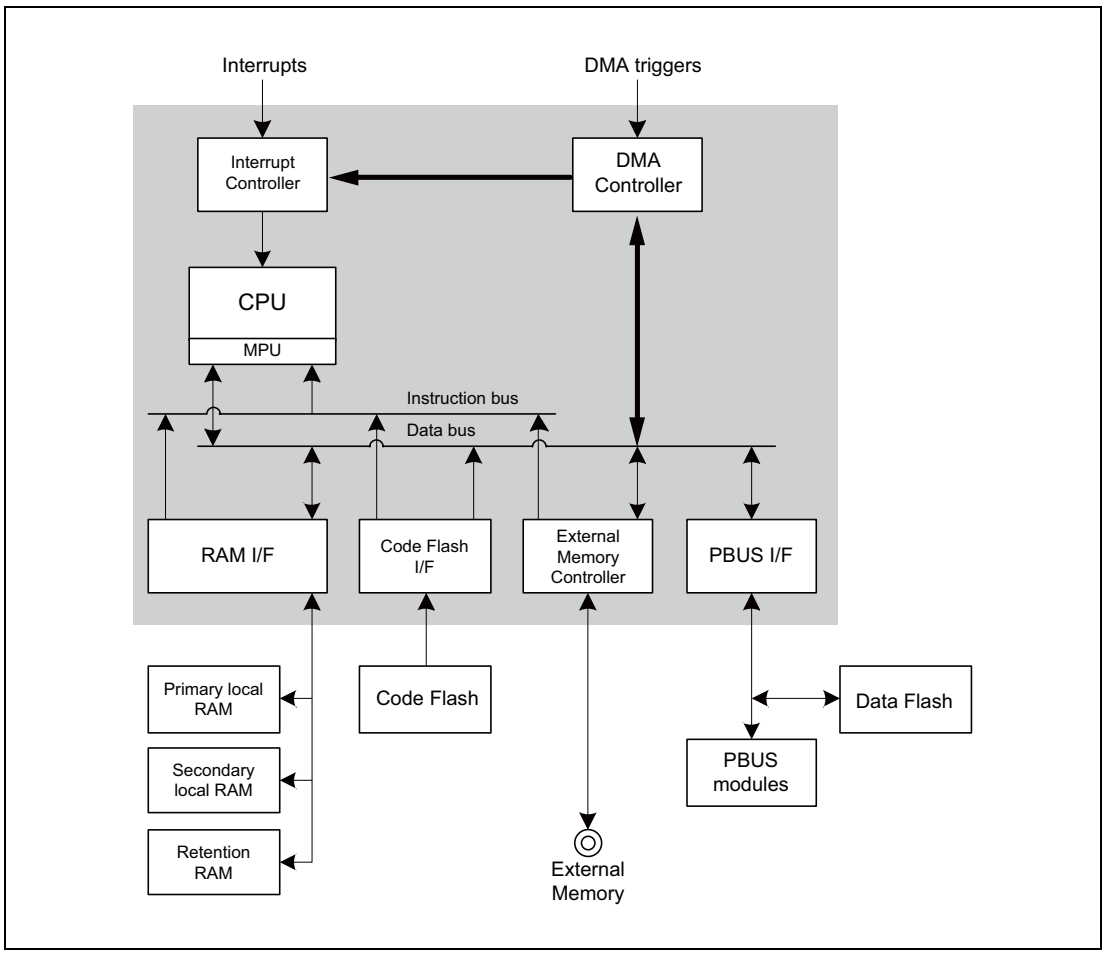


Figure 7.1 Block Diagram of the DMA

7.3 Registers

7.3.1 List of Registers

The following table lists the DMA registers.

For details about <DMA_base>, see **Section 7.1.2, Register Base Addresses**.

Table 7.10 List of Registers (1/2)

Unit Name	Register Name	Abbreviation	Address
DMA	**Channel common registers**		
	DMA transfer request control register 0	DTRC0	<DMA_base> + 00_H
	DMA channel master setting register 0	DMCM0	<DMA_base> + 04_H
	Channel registers (m = 0 to 7)		
	DMA source address register CHm	DSAm	<DMA_base> + (m × 30_H) + 14_H
	DMA source address register L CHm	DSAmL	<DMA_base> + (m × 30_H) + 14_H
	DMA source address register H CHm	DSAmH	<DMA_base> + (m × 30_H) + 16_H
	DMA destination address register CHm	DDAm	<DMA_base> + (m × 30_H) + 24_H
	DMA destination address register L CHm	DDAmL	<DMA_base> + (m × 30_H) + 24_H
	DMA destination address register H CHm	DDAmH	<DMA_base> + (m × 30_H) + 26_H
	DMA transfer count register	DTCm	<DMA_base> + (m × 30_H) + 32_H
	DMA transfer control register CHm	DTCTm	<DMA_base> + (m × 30_H) + 38_H
	DMA transfer status register CHm	DTSm	<DMA_base> + (m × 30_H) + $3A_H$
	Channel registers (m = 8 to 15)		
	DMA source address register CHm	DSAm	<DMA_base> + (m × 30_H) + 94_H
	DMA source address register L CHm	DSAmL	<DMA_base> + (m × 30_H) + 94_H
	DMA source address register H CHm	DSAmH	<DMA_base> + (m × 30_H) + 96_H
	DMA destination address register CHm	DDAm	<DMA_base> + (m × 30_H) + $A4_H$
	DMA destination address register L CHm	DDAmL	<DMA_base> + (m × 30_H) + $A4_H$
	DMA destination address register H CHm	DDAmH	<DMA_base> + (m × 30_H) + $A6_H$
	DMA transfer count register	DTCm	<DMA_base> + (m × 30_H) + $B2_H$
	DMA transfer control register CHm	DTCTm	<DMA_base> + (m × 30_H) + $B8_H$
	DMA transfer status register CHm	DTSm	<DMA_base> + (m × 30_H) + BA_H
	Trigger source registers		
	DMA trigger factor register 0	DTFR0	<DMA_base> + 800_H
	DMA trigger factor register 1	DTFR1	<DMA_base> + 802_H
	DMA trigger factor register 2	DTFR2	<DMA_base> + 804_H
	DMA trigger factor register 3	DTFR3	<DMA_base> + 806_H
	DMA trigger factor register 4	DTFR4	<DMA_base> + 808_H
	DMA trigger factor register 5	DTFR5	<DMA_base> + $80A_H$
	DMA trigger factor register 6	DTFR6	<DMA_base> + $80C_H$
	DMA trigger factor register 7	DTFR7	<DMA_base> + $80E_H$
	DMA trigger factor register 8	DTFR8	<DMA_base> + 810_H
	DMA trigger factor register 9	DTFR9	<DMA_base> + 812_H
	DMA trigger factor register 10	DTFR10	<DMA_base> + 814_H
	DMA trigger factor register 11	DTFR11	<DMA_base> + 816_H
	DMA trigger factor register 12	DTFR12	<DMA_base> + 818_H

Table 7.10 List of Registers (2/2)

Unit Name	Register Name	Abbreviation	Address
DMA	DMA trigger factor register 13	DTFR13	<DMA_base> + $81A_H$
	DMA trigger factor register 14	DTFR14	<DMA_base> + $81C_H$
	DMA trigger factor register 15	DTFR15	<DMA_base> + $81E_H$
	DMA request clear register	DRQCLR	<DMA_base> + 840_H
	DMA request check register	DRQSTR	<DMA_base> + 844_H

7.3.2 Enabling or Disabling Writing Control Registers

The following control registers cannot be written while DMA transfer is enabled (DTSm.DTE = 1). All these registers can always be read, however.

Table 7.11 Enabling/Disabling Writing Control Registers

Always writable	DTRC0, DTSm, DRQCLR, DTCTm.LE
Writing prohibited while DMA transfer is enabled (DTSm.DTSmDTE = 1) (Operation is not guaranteed if these registers are written.)	DSAm, DSAmL, DSAmH, DDAm, DDAmL, DDAmH, DTCm, DTCTm (excluding the LE bit), DTFRm

m = 0 to 15

7.3.3 DMAC Control Registers

7.3.3.1 DTRC0 — DMA Transfer Request Control Register 0

Access: This register can be read or written in 8- or 1-bit units.

Address: <DMA_base> + 00_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	DTRC0ERR	—	—	—	—	—	—	DTRC0ADS
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R/W

Table 7.12 DTRC0 Registers Contents

Bit Position	Bit Name	Function
7	DTRC0ERR	DMA transfer error status This bit indicates that an error response has been received from the transfer target during DMA transfer. If an error response is received, the ERR and ADS bits are set and a SYSERR exception is generated. To clear this bit, write “0” to it. 0: No DMA transfer error 1: DMA transfer error
6 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	DTRC0ADS	DMA transfer suspended This bit indicates that DMA transfer has been suspended by the generation of DMA transfer error. In addition, the current DMA transfer can be suspended if the user writes “1” to this bit. 0: DMA transfer not suspended 1: DMA transfer suspended/DMA transfer suspended request

7.3.3.2 DMCM0 — DMA Channel Master Setting Register 0

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

Address: <DMA_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
	DMCM0 UM15	DMCM0 UM14	DMCM0 UM13	DMCM0 UM12	DMCM0 UM11	DMCM0 UM10	DMCM0 UM9	DMCM0 UM8	DMCM0 UM7	DMCM0 UM6	DMCM0 UM5	DMCM0 UM4	DMCM0 UM3	DMCM0 UM2	DMCM0 UM1	DMCM0 UM0
Value after 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 7.13 DMCM0 Registers Contents

Bit Position	Bit Name	Function
15 to 0	DMCM0UM15 to DMCM0UM0	These bits set masters to be assigned to each channel. 0: Supervisor mode is assigned as the master 1: User mode is assigned as the master **CAUTION** In supervisor mode, the setting register of channels set in user mode can be accessed. However, when the setting register of channels set in supervisor mode is accessed in user mode, an access error occurs.

7.3.3.3 DSAm — DMA Source Address Register

Access: DSAm can be read or written in 32-bit units.
DSAmL and DSAmH can be read or written in 16-bit units.

Address:
- When m is 0 to 7:
DSAm: <DMA_base> + (m × 30_H) + 14_H
DSAmL: <DMA_base> + (m × 30_H) + 14_H
DSAmH: <DMA_base> + (m × 30_H) + 16_H
- When m is 8 to 15:
DSAm: <DMA_base> + (m × 30_H) + 94_H
DSAmL: <DMA_base> + (m × 30_H) + 94_H
DSAmH: <DMA_base> + (m × 30_H) + 96_H

Value after reset: 0XXX XXXX$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DSAmHSA26	DSAmHSA25	DSAmHSA24	DSAmHSA23	DSAmHSA22	DSAmHSA21	DSAmHSA20	DSAmHSA19	DSAmHSA18	DSAmHSA17	DSAmHSA16
Value after reset	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
	DSAmLSA15	DSAmLSA14	DSAmLSA13	DSAmLSA12	DSAmLSA11	DSAmLSA10	DSAmLSA9	DSAmLSA8	DSAmLSA7	DSAmLSA6	DSAmLSA5	DSAmLSA4	DSAmLSA3	DSAmLSA2	DSAmLSA1	DSAmLSA0
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 7.14 DSAm Registers Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
26 to 16	DSAmHSA26 to DSAmHSA16	DMA source address These bits set the higher-order 10 bits of the transfer source address of channel m. If this register is referenced during DMA transfer, the address for which a DMA cycle is to be executed next can be read. When referencing the register, it is recommended to access DSAmL or DSAmH in 32-bit units. When DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.
15 to 0	DSAmLSA15 to DSAmLSA0	DMA source address These bits set the lower-order 16 bits of the transfer source address of channel m. If this register is referenced during DMA transfer, the address for which a DMA cycle is to be executed next can be read. When referencing the register, it is recommended to access DSAmL or DSAmH in 32-bit units. When DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

CAUTIONS

1. Writing this register is prohibited while DMA transfer is enabled (DTSm.DTSmDTE = 1). If it is written, correct DMA operation is not guaranteed.
2. Set an address by accessing in 32-bit units while the DTSmDTE bit is “0” in order to avoid data being transferred from an address that has not been completely set.
3. DMA transfer of misaligned data is not supported. The lower 2 bits of an address corresponding to the transfer data size are as follows (x indicates any bit).
The operation is not guaranteed if a setting other than the following is made.

Data Size	DSAm LSA1	DSAm LSA0
8 bits	—	—
16 bits	—	0
32 bits	0	0

4. When the DSAm register is read in 32-bit units while DMA transfer is enabled (DTSm.DTSmDTE = 1), the read value may not be correct. Whether the read value is correct or not can be judged using the following method.
   (Method of judging)
   - Read DSAm register in 32-bit units twice in succession (read for the first time, and then read for the second time)
     If the upper 16 bits of the first and the second read values are the same, the second read value is correct.
     If the upper 16 bits of the first and the second read values are different, the first read value is correct.
5. The DMA access destination by the DSAm setting is an address where bit 26 is sign-extended for the higher-order 5 bits of DSAmH. However, when reading the DSAmH register, “0” is read from the higher-order 5 bits.

7.3.3.4 DDAm — DMA Destination Address Register

Access: DDAm can be read or written in 32-bit units.
DDAmL and DDAmH can be read or written in 16-bit units.

Address: • When m is 0 to 7:
DDAm: <DMA_base> + (m × 30_H) + 24_H
DDAmL: <DMA_base> + (m × 30_H) + 24_H
DDAmH: <DMA_base> + (m × 30_H) + 26_H
• When m is 8 to 15:
DDAm: <DMA_base> + (m × 30_H) + $A4_H$
DDAmL: <DMA_base> + (m × 30_H) + $A4_H$
DDAmH: <DMA_base> + (m × 30_H) + $A6_H$

Value after reset: 0XXX $XXXX_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DDAmHDA26	DDAmHDA25	DDAmHDA24	DDAmHDA23	DDAmHDA22	DDAmHDA21	DDAmHDA20	DDAmHDA19	DDAmHDA18	DDAmHDA17	DDAmHDA16
Value after reset	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
	DDAmLDA15	DDAmLDA14	DDAmLDA13	DDAmLDA12	DDAmLDA11	DDAmLDA10	DDAmLDA9	DDAmLDA8	DDAmLDA7	DDAmLDA6	DDAmLDA5	DDAmLDA4	DDAmLDA3	DDAmLDA2	DDAmLDA1	DDAmLDA0
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 7.15 DDAm Registers Contents

Bit Position	Bit Name	Function
31 to 27	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
26 to 16	DDAmHDA26 to DDAmHDA16	DMA destination address These bits set the higher-order 10 bits of the transfer destination address of channel m. If this register is referenced during DMA transfer, the address for which a DMA cycle is to be executed next can be read. When referencing the register, it is recommended to access DDAmL or DDAmH in 32-bit units. When DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.
15 to 0	DDAmLDA15 to DDAmLDA0	DMA destination address These bits set the lower-order 16 bits of the transfer destination address of channel m. If this register is referenced during DMA transfer, the address for which a DMA cycle is to be executed next can be read. When referencing the register, it is recommended to access DDAmL or DDAmH in 32-bit units. When DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

CAUTIONS

1. Writing these bits is prohibited while DMA transfer is enabled (DTSm.DTSmDTE = 1). If they are written, the operation is not guaranteed.
2. Set an address by accessing in 32-bit units while the DTSmDTE bit is “0” in order to avoid data being transferred from an address that has not been completely set.
3. If an error occurs in the transfer target in the read cycle of DMA transfer, the write cycle is not executed but the destination address is updated.
4. DMA transfer of misaligned data is not supported. The lower 2 bits of an address corresponding to the transfer data size are as follows (x indicates any bit).
The operation is not guaranteed if a setting other than the following is made.

Data Size	DDAm LDA1	DDAm LDA0
8 bits	—	—
16 bits	—	0
32 bits	0	0

5. When the DDAm register is read in 32-bit units while DMA transfer is enabled (DTSm.DTSmDTE = 1), the read value may not be correct. Whether the read value is correct or not can be judged using the following method.
(Method of judging)
   - Read DDAm register in 32-bit units twice in succession (read for the first time, and read for the second time)
     If the upper 16 bits of the first and the second read values are the same, the second read value is correct.
     If the upper 16 bits of the first and the second read values are different, the first read value is correct.
6. The DMA access destination by the DDAm setting is an address where bit 26 is sign-extended for the higher-order 5 bits of DDAmH. However, when reading the DDAmH register, “0” is read from the higher-order 5 bits.

7.3.3.5 DTCm — DMA Transfer Count Register (m = 0 to 15)

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

Address:
- When m is 0 to 7:
DTCm: <DMA_base> + (m × 30_H) + 32_H
- When m is 8 to 15:
DTCm: <DMA_base> + (m × 30_H) + $B2_H$

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DTCm DTC14	DTCm DTC13	DTCm DTC12	DTCm DTC11	DTCm DTC10	DTCm DTC9	DTCm DTC8	DTCm DTC7	DTCm DTC6	DTCm DTC5	DTCm DTC4	DTCm DTC3	DTCm DTC2	DTCm DTC1	DTCm DTC0
Value after reset	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 7.16 DTCm Registers Contents

Bit Position	Bit Name	Function
15	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
14 to 0	DTCmDTC14 to DTCmDTC0	DMA transfer count These bits specify the number of times of DMA transfers (DMA transfer count) for channel m. When this register is referenced during DMA transfer, the remaining number of times DMA transfer to be executed can be read. When DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

DTCmDTC[14:0]	Operation
0000_H	Transfer executed 32,768 times
0001_H	Transfer executed once or transfer to be executed once
:	:
$7FFF_H$	Transfer executed 32,767 times or 32,767 times of transfer to be executed

CAUTIONS

1. Writing these bits is prohibited while DMA transfer is enabled (DTSm.DTSmDTE bit = 1). If they are written, the operation is not guaranteed.
2. If an error occurs in the transfer target in the read cycle of DMA transfer, the write cycle is not executed but the transfer count is updated.

7.3.3.6 DTCTm — DMA Transfer Control Register (m = 0 to 15)

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

Address: • When m is 0 to 7:
DTCTm: <DMA_base> + (m × 30_H) + 38_H
• When m is 8 to 15:
DTCTm: <DMA_base> + (m × 30_H) + $B8_H$

Value after reset: $x000_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DTCTm DS1	DTCTm DS0	DTCTm LE	—	—	—	—	DTCTm SACM1	—	DTCTm DACM1	—	—	—	—	—
Value after reset	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	R	R	R/W	R	R/W	R	R	R	R	R

Table 7.17 DTCTm Registers Contents

Bit Position	Bit Name	Function
15	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
14, 13	DTCTmDS1 DTCTmDS0	DMA transfer data size These bits specify the DMA transfer data size of channel m.
12	DTCTmLE	Loop enable This bit specifies whether to acknowledge the next DMA transfer request, even if the DTSm.TC bit is not cleared (to “0”) after DMA transfer has been completed in single transfer mode. If this bit is set (to “1”), the DTSm.DTSmDTE bit is not cleared upon completion of DMA transfer. Even if the TC bit is not cleared, DMA transfer is executed if a DMA transfer request is issued. This bit can be written while the DTSmDTE bit is being written (to “1”). 0: Clears the DTSmDTE bit upon completion of DMA transfer. 1: Does not clear the DTSmDTE bit upon completion of DMA transfer.
11 to 8	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
7	DTCTmSACM1	DMA transfer source address counting direction These bits specify the direction in which the transfer source address of channel m is to be counted. 0: Increment 1: Fixed
6	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
5	DTCTmDACM1	DMA transfer destination address counting direction These bits specify the direction in which the transfer destination address of channel m is to be counted. 0: Increment 1: Fixed
4 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

DTCTmDS1	DTCTmDS0	DMA Transfer Data Size
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	Setting prohibited

CAUTION

Writing these bits is prohibited while DMA transfer is enabled (DTSm.DTSmDTE bit = 1). If they are written, the bits other than the LE bit are not affected by the write operation (only the LE bit can be updated).

7.3.3.7 DTSm — DMA Transfer Status Register (m = 0 to 15)

Access: This register can be read or written in 8- or 1-bit units.

Address: • When m is 0 to 7:
DTSm: <DMA_base> + (m × 30_H) + $3A_H$
• When m is 8 to 15:
DTSm: <DMA_base> + (m × 30_H) + BA_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	DTSmTC	DTSmDT	—	—	DTSmER	DTSmDR	DTSmSR	DTSmDTE
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 7.18 DTSm Registers Contents (1/2)

Bit Position	Bit Name	Function
7	DTSmTC	DMA transfer end status This bit indicates that DMA transfer has been completed. Write “0” to this bit to clear it after reading “1” from it. It is recommended to write this bit using bit manipulation such as CLR1. Note that, when bit manipulation (SET1, CLR1, NOT1, or TST1) is performed to other bits of the DTS register while the TC bit is set, it is considered that “1” is read from this bit. 0: DMA transfer not completed 1: DMA transfer completed
6	DTSmDT	DMA transfer status This bit indicates that a DMA transfer request has been acknowledged and that DMA transfer is in progress. It is not set (to “1”) when only a DMA transfer request is issued. This bit is cleared (to “0”) when DMA transfer has been completed. If the DTSmDTE bit is “0”, this bit can be cleared by the user. (It can also be written at the same time as the DTSmDTE bit.) After the clearance operation, when the ongoing DMA cycle has been completed, this bit is updated. 0: DMA transfer is not executed 1: DMA transfer in progress
5, 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	DTSmER	DMA transfer error flag This bit indicates that a DMA transfer error has occurred in channel m. It is cleared (to “0”) when the DTRC0.ERR bit is cleared. Note that this bit is read only. 0: No DMA transfer error 1: DMA transfer error
2	DTSmDR	Hardware DMA transfer request flag This bit indicates that channel m has a hardware DMA transfer request. It is cleared (to “0”) when the hardware DMA transfer request is deasserted. This bit operates regardless of the status of the DTSmDTE bit. It is not set (to “1”) by a software DMA transfer request. Note that this bit is read only. 0: No hardware DMA transfer request 1: Hardware DMA transfer request

Table 7.18 DTSm Registers Contents (2/2)

Bit Position	Bit Name	Function
1	DTSmSR	Software DMA transfer request This bit sets a software DMA transfer request. If the SR bit is set to “1” while the DTSmDTE bit is set to “1”, DMA transfer is executed. This bit is automatically cleared (to “0”) when DMA transfer has been completed. Writing “0” to the SR bit during single-step transfer issued by this bit suspends DMA transfer. 0: No software DMA transfer request 1: Software DMA transfer request
0	DTSmDTE	DMA transfer enable This bit enables or disables DMA transfer. DMA transfer is executed if “1” is written to this bit and a DMA transfer request is issued. This bit is automatically cleared (to “0”) if the LE bit is “0” when DMA transfer has been completed. DMA transfer is suspended if “0” is written to this bit during DMA transfer. 0: Disables DMA transfer 1: Enables DMA transfer

7.3.4 DTFR Control Registers

7.3.4.1 DTFRm — DTFRm Register (m = 0 to 15)

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

Address: DTFRm: <DMA_base> + 800_H + (m × 2)

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTFRmREQEN	—	—	—	—	—	—	—	—	—	DTFRmIFCm5	DTFRmIFCm4	DTFRmIFCm3	DTFRmIFCm2	DTFRmIFCm1	DTFRmIFCm0
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	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.19 DTFRm Registers Contents

Bit Position	Bit Name	Function
15	DTFRmREQEN	This bit enables or disables the generation of the DMA trigger factor. 1: Enables the generation of the DMA trigger factor. 0: Disables the generation of the DMA trigger factor.
14 to 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5 to 0	DTFRmIFCm5 to DTFRmIFCm0	These bits select the transfer source. The set values are shown in **Table 7.6** and **Table 7.7**.

NOTE

The selected transfer source is always sampled. When a transfer source is generated, a bit in the DRQSTR is always set.

7.3.4.2 DRQCLR — DMA Request Clear Register

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

Address: <DMA_base> + 840_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRQCLRRQCR15	DRQCLRRQCR14	DRQCLRRQCR13	DRQCLRRQCR12	DRQCLRRQCR11	DRQCLRRQCR10	DRQCLRRQCR9	DRQCLRRQCR8	DRQCLRRQCR7	DRQCLRRQCR6	DRQCLRRQCR5	DRQCLRRQCR4	DRQCLRRQCR3	DRQCLRRQCR2	DRQCLRRQCR1	DRQCLRRQCR0
Value after 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 7.20 DRQCLR Register Contents

Bit Position	Bit Name	Function
15 to 0	DRQCLRRQCR15 to DRQCLRRQCR0	Set these bits to 1 to clear the corresponding transfer request held in channel m.

NOTE

Writing “0” to bits 15 to 0 is ignored. When read, these bits are always read as 0.

7.3.4.3 DRQSTR — DMA Request Check Register

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

Address: <DMA_base> + 844_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRQSTRRQST15	DRQSTRRQST14	DRQSTRRQST13	DRQSTRRQST12	DRQSTRRQST11	DRQSTRRQST10	DRQSTRRQST9	DRQSTRRQST8	DRQSTRRQST7	DRQSTRRQST6	DRQSTRRQST5	DRQSTRRQST4	DRQSTRRQST3	DRQSTRRQST2	DRQSTRRQST1	DRQSTRRQST0
Value after reset	0	0	0	0	0	0	0	0	0	0	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 7.21 DRQSTR Register Contents

Bit Position	Bit Name	Function
15 to 0	DRQSTRRQST15 to DRQSTRRQST0	DMA transfer request status flag 1: Request issued (DMA transfer request signal m is “1”) 0: No request (DMA transfer request signal m is “0”)

NOTE

The selected transfer source is always sampled. When a transfer source is generated, a bit in the DRQSTR is always set.

7.4 Functions

7.4.1 DMAC Transfer Modes

A single-transfer mode and a single-step transfer mode are supported as transfer modes.

In either mode, transfer is executed in two cycles (dual address transfer) and therefore, a read cycle and a write cycle are generated each time a transfer is executed.

Because the bus is not locked, a CPU cycle may occur between the read and write cycles.

7.4.1.1 Single Transfer Mode (when hardware DMA transfer request is generated)

When a hardware DMA transfer request is acknowledged, data of the selected transfer size (8, 16, or 32 bits) is transferred. Each time transfer has been executed, the bus is released and the DMA controller waits for a DMA transfer request.

Each time a hardware DMA transfer request has been acknowledged, transfer is executed once and the DMA transfer request is cleared. This operation is repeated the number of times specified by the DMA transfer count register m (DTCm (m = 15 to 0)). INTDMAm is generated when the transfer is completed.

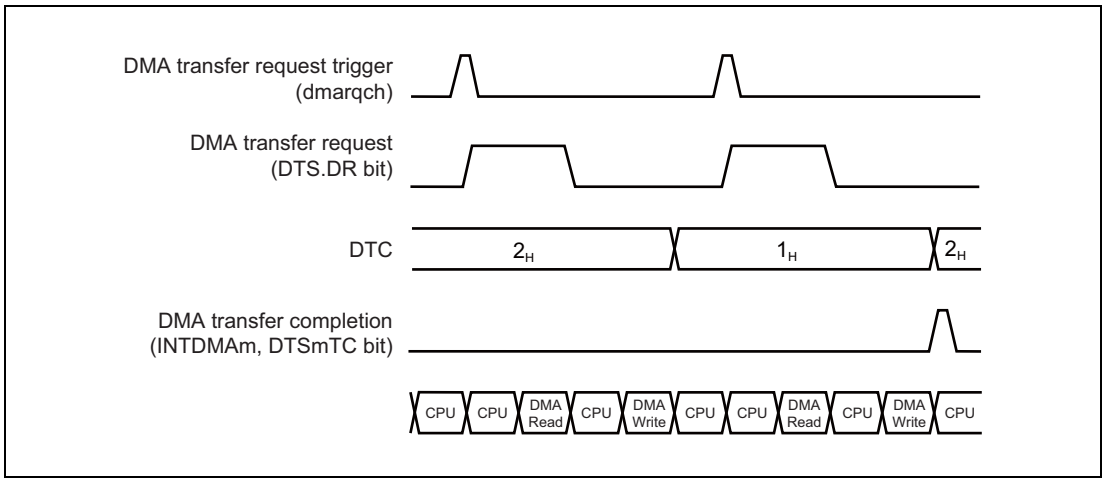


Figure 7.2 Example of Single Transfer (Transfer count is two times)

7.4.1.2 Single-Step Transfer Mode (when software DMA transfer request is generated)

When a software DMA transfer request is acknowledged, data of the selected transfer size (8, 16, or 32 bits) is transferred. Each time transfer has been executed, the bus is released.

Once a software DMA transfer request has been acknowledged, this operation is repeated the number of times specified by the DMA transfer count register m (DTCm (m = 15 to 0)). When the transfer is completed, INTDMAm is generated and software DMA transfer request is cleared. Because the priority is identified each time transfer is executed, the DMA cycle of a channel having the higher priority may interrupt.

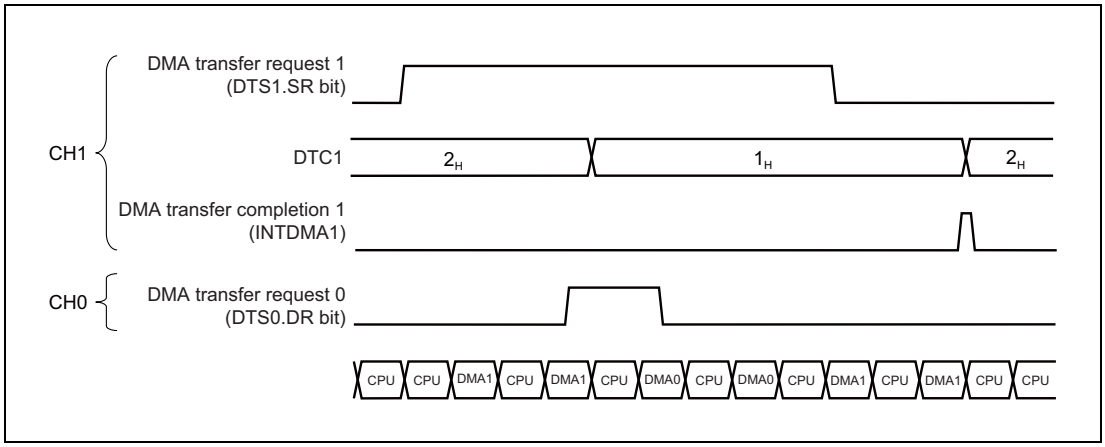


Figure 7.3 Example of Single-Step Transfer

7.4.2 DMAC Channel Priority Control

The priority of each channel is fixed and is as follows.

DMAC0 (CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 > CH8 > CH9 > CH10 > CH11 > CH12 > CH13 > CH14 > CH15)

In single-step transfer, the priority is decided per round of transfer, so interruption of the DMA cycle of a channel with higher priority is possible. However, DMA transfer with higher priority does not interrupt between the read and write cycles.

An example where DMA transfer request with a high priority is generated when DMA transfer is executed is shown below.

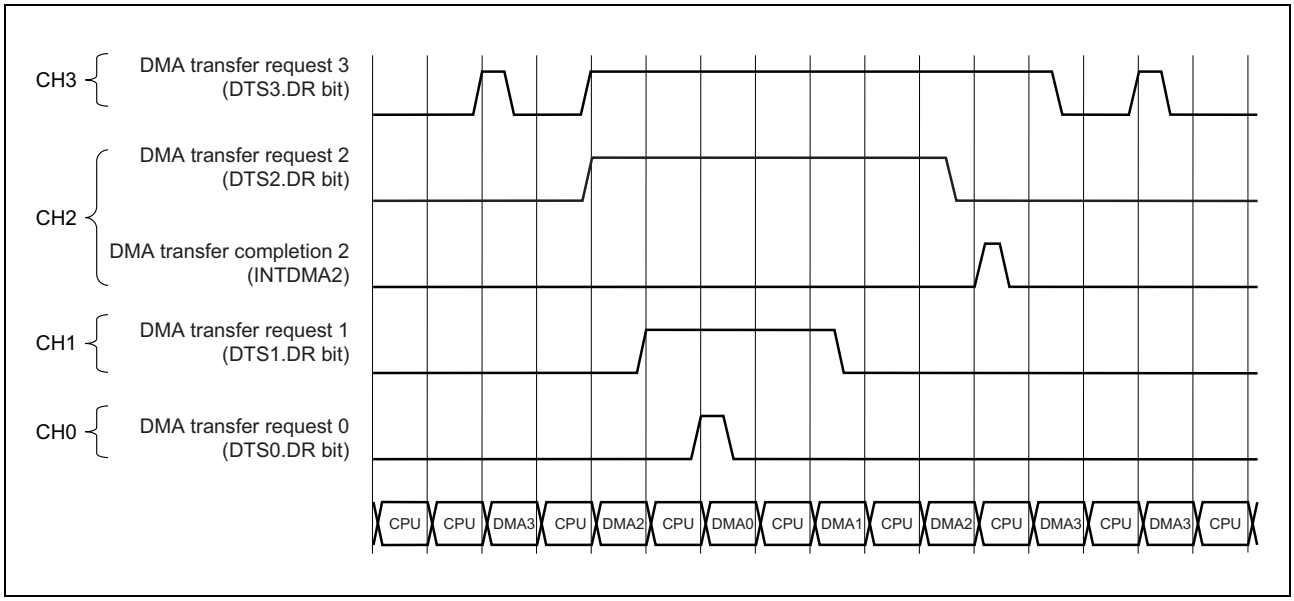


Figure 7.4 Example of Priority Control of DMA Transfer Channels

7.4.3 Valid DMAC Transfer Request Conditions

Whether a DMA transfer request of channel m is acknowledged depends on the setting of the ERR and ADS bits of the DMA transfer request control register 0 (DTRC0), the LE bit of the DMA transfer control register (DTCTm), and the TC and DTSmDTE bits of the DMA transfer status register (DTSm). **Table 7.22** shows the relationship between the setting of each of the above bits and whether a DMA transfer request is acknowledged.

Table 7.22 Valid DMA Transfer Request Conditions of Channel m

DTFRmREQEN	DTCTmLE	DTSmTC	DTSmDTE	DTRC0ERR	DTRC0ADS	Hardware DMA Transfer Request	Software DMA Transfer Request	Status
—	—	—	—	—	1	Disabled	Disabled	DMA transfer is suspended
—	—	—	—	1	—	Disabled	Disabled	DMA transfer error occurs
—	—	—	0	—	—	Disabled	Disabled	DMA transfer is disabled
0	—	0	1	0	0	Disabled	Enable	Hardware transfer request mask by DTFR
1	—	0	1	0	0	Enable*1	Enable*1	DMA transfer request can be acknowledged
—	0	1	0	0	0	Disabled	Disabled	DMA transfer is completed (LE disable)
0	1	1	1	0	0	Disabled	Enable	DMA transfer is completed (LE enable)
1	1	1	1	0	0	Enable*1	Enable*1	DMA transfer is completed (LE enable)

Note 1. The configuration diagram of DMA transfer request is shown below. A hardware DMA transfer request and software DMA transfer request are logically added (ORed). Therefore, a hardware DMA transfer request and software DMA transfer request cannot be set at the same time. When DMA transfer is executed by a software DMA transfer request, it is not affected by a hardware DMA transfer request by setting the DTFR.REQEN bit to “0”.
Note that, when a hardware DMA trigger factor (dmarqch (m)) is entered while the REQEN bit is set to “0”, the entered trigger factor is held in the DRQSTR register. In this case, even if a software DMA transfer request is acknowledged, the held hardware DMA trigger factor is not cleared.

Note: m = 15 to 0 (16-channel DMA)

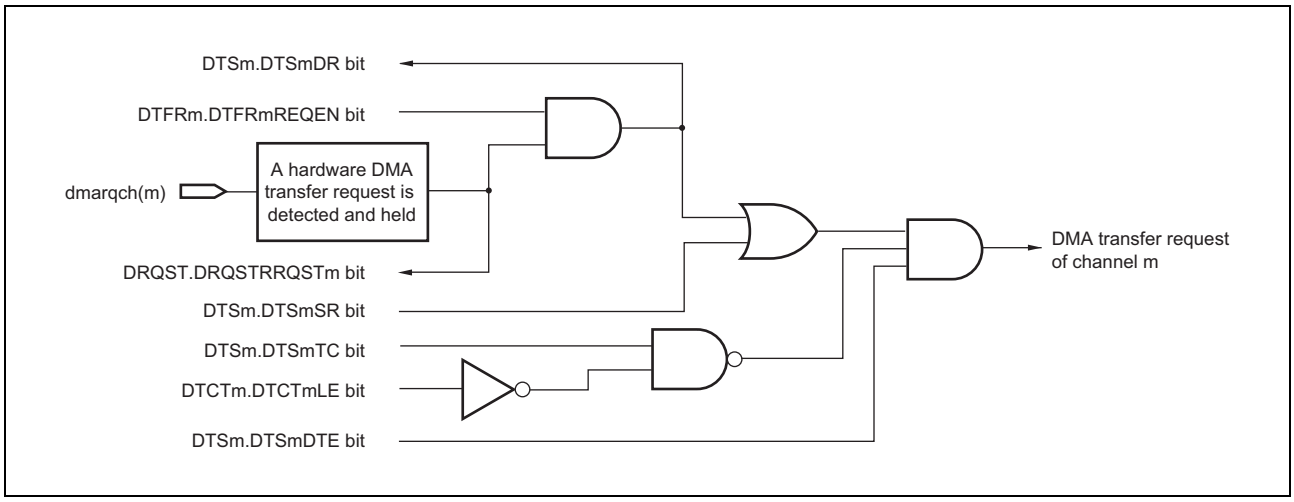


Figure 7.5 Configuration Diagram of Transfer Requests of Channels

7.4.4 Suspending/Resuming/Forcibly Terminating DMAC Transfer

7.4.4.1 Suspending or Resuming DMA Transfer for All Channels through Software

By setting the DMA transfer suspend bit (ADS) of the DMA transfer request control register 0 (DTRC0), the DMA transfer can be suspended. During a DMA cycle, the next DMA transfer is suspended after the ongoing DMA cycle has been completed. Note that the DMA transfer enable bit (DTE) and the software DMA transfer request bit (SR) of the DMA transfer status register (DTSm) are not cleared.

To resume the suspended DMA transfer, clear the ADS bit. If a DMA transfer is requested at that point, the transfer is executed starting from the channel having the highest priority at that time. To end DMA transfer, clear the DMA transfer request with the DTSmDTE bit cleared.

7.4.4.2 Suspending or Resuming DMA Transfer for Each Channel using DMA Transfer Enable Bit (DTE)

By clearing the DMA transfer enable bit (DTE) of the DMA transfer status register (DTSm), the DMA transfer can be suspended. During a DMA cycle, the next DMA transfer is suspended after the ongoing DMA cycle is completed. Note that the software DMA transfer request bit (SR) of DTSm is not cleared.

To resume the suspended DMA transfer, set the DTSmDTE bit. If another channel is not executing DMA transfer at that point, the priority is identified as usual. If another channel is executing DMA transfer, the priority is identified after that transfer has been completed. To end DMA transfer, clear the DMA transfer request with the DTSmDTE bit cleared.

7.4.4.3 Suspending or Resuming DMA Transfer by using Software DMA Transfer Request Bit (SR)

By clearing the software DMA transfer request bit (SR) of the DMA transfer status register (DTSm), the next DMA transfer and those that follow can be suspended. During a DMA cycle, the next DMA transfer is suspended after the ongoing DMA cycle has been completed. Note that DMA transfer enable bit (DTE) of DMA transfer status register (DTSm) is not cleared.

To resume the suspended DMA transfer, set the SR bit. If another channel is not executing DMA transfer at that point, the priority is identified as usual. If another channel is executing DMA transfer, the priority is identified after that transfer has been completed.

7.4.4.4 Forcibly Terminating Suspended DMA Transfer

To end the suspended DMA transfer, perform the following procedure for each channel.

- Clear the DTSmDTE bit (to “0”) while the DMA transfer is suspended.
- Clear the DMA transfer status bit (DT) of the DMA transfer status register (DTSm).
- Clear the DMA transfer request (the RQSTm bit of the DRQSTR register for hardware DMA transfer requests, or the SR bit of the DTSm register for software DMA transfer requests) (to “0”).

Note that execution of the above procedure may not end DMA transfer immediately. When re-setting the DMA setting register after executing the procedure to end transfer, do so after confirming that the read value of the DT bit is 0.

Note also that DMA transfer cannot be resumed once the above procedure is executed.

7.4.5 Error Response

7.4.5.1 Suspending DMA Transfer by Error Response

When an error occurs at the DMA transfer source or transfer destination, the DMA transfer suspend bit (ADS) of the DMA transfer request control register 0 (DTRC0) is set to suspend all of the subsequent DMA transfers. At the same time, the DMA transfer error status bit (ERR) is set and a SYSERR exception is generated towards the CPU. The user can evaluate channel in which the error has occurred, by using the DMA transfer error flag (ER) of the DMA transfer status register (DTSm), when the user has confirmed that ERR has been set.

In this case, note that, if an error response is acknowledged in the read cycle, the write cycle is not executed, but the transfer address and the transfer count are updated.

7.4.5.2 Canceling Transfer Suspend by Error Response

DMA transfer suspend can be canceled by clearing the DMA transfer suspend bit (ADS) and DMA transfer error status bit (ERR) of the DMA transfer request control register 0 (DTRC0).

Clear the DMA transfer enable bit (DTE) and the DMA transfer status bit (DT) in the DMA transfer status register (DTSm) in advance, so that DMA transfer is not resumed after its suspend has been canceled. In the case of a hardware DMA transfer request, only the RQSTm bit of the DRQSTR register requires clearing. For a software DMA transfer request, also clear the SR bit of the DTSm register.

7.4.6 Stand-By Support

When a transfer request to STOP mode is generated, DMA transfer stops after completion of the DMA cycle currently being executed. Unlike DMA transfer suspend caused by software, this does not require any operation of the DMA control register. DMA transfer resumes upon cancellation of the STOP mode, and if a DMA request is retained, that DMA transfer starts in order of priority of each channel.

7.5 Protection Function

An access master can be set by using a DMA control register (the DMCM0 register) in supervisor mode or user mode. Even if the user mode is set as the master by the DMCM0 register, each register can be accessed in supervisor mode. Note that, when the following registers are accessed in user mode, an access error occurs.

- DMA setting register set in supervisor mode
- DMA channel master setting register 0 (DMCM0)

When an access error occurs, writing the DMA setting register is ignored, but reading it can be performed as usual.

The DMCM0 register cannot be accessed in user mode. It can be accessed only in supervisor mode.

7.6 DMAC Transfer Setting Flow

Figure 7.6 shows the flow for setting DMAC transfer.

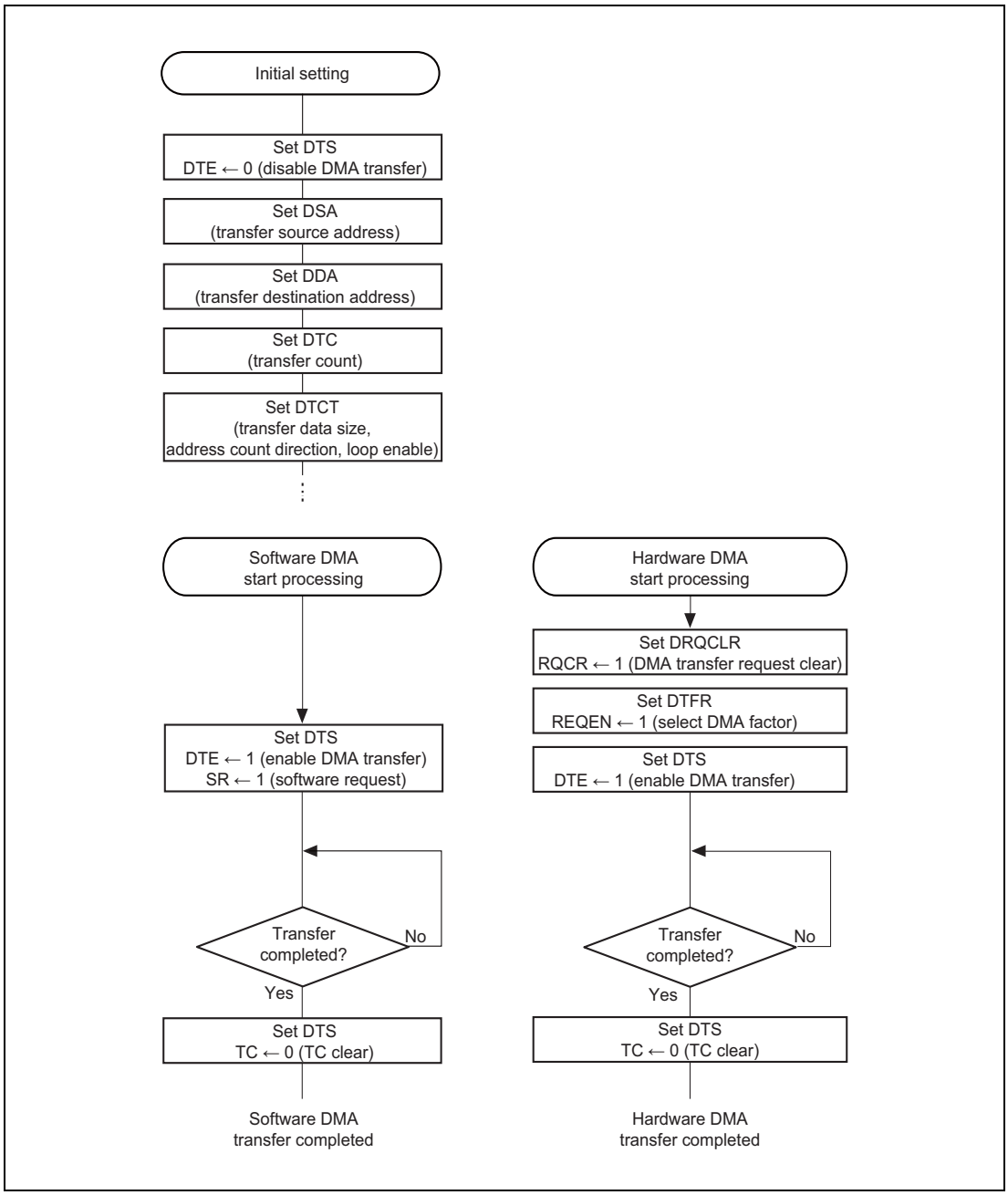


Figure 7.6 DMAC Transfer Setting Flow

CAUTION

Because DMA always samples hardware DMA trigger factors, a trigger factor may be held in the DRQSTR register sometimes during the initial settings of DMA transfer. The held trigger factor can be cleared by the function of the DRQCLR register. If necessary, clear unnecessary trigger factors before setting the DTE bit to 1.

Section 8 Resets

8.1 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 monitors 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 ($\overline{\text{CVMRES}}$)
- At the transition to DEEPSTOP mode

8.1.1 Reset Sources

Reset levels and reset sources are shown below.

Various reset sources are assigned to the different levels of the reset.

Table 8.1 Reset Sources and Reset Targets

Reset Level	Reset Source	Clock Generator (except PLL)/ Real-Time Clock/CVM	AWO Modules*1	ISO Modules *3
1	Power-on-clear (POCRES) Debugger reset ($\overline{\text{DBRES}}$)	Reset*2	Reset	Reset
2	External reset ($\overline{\text{RESET}}$) Watchdog timer reset (WDTA0RES, WDTA1RES) Clock monitor reset ($\overline{\text{CLMA0RES}}$, $\overline{\text{CLMA1RES}}$, $\overline{\text{CLMA2RES}}$) Core voltage monitor ($\overline{\text{CVMRES}}$) Low voltage indicator reset ($\overline{\text{LVIRES}}$) Software reset (SWRES)	Not for reset target*4	Reset	Reset
3	Reset by DEEPSTOP mode	Not for reset target	Not for reset target	Reset

Note 1. Clock generator, real-time clock, and CVM 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, ocsillator-related registers for clock monitoring are initialized.

Reset level 1: Initializes the entire microcontroller.

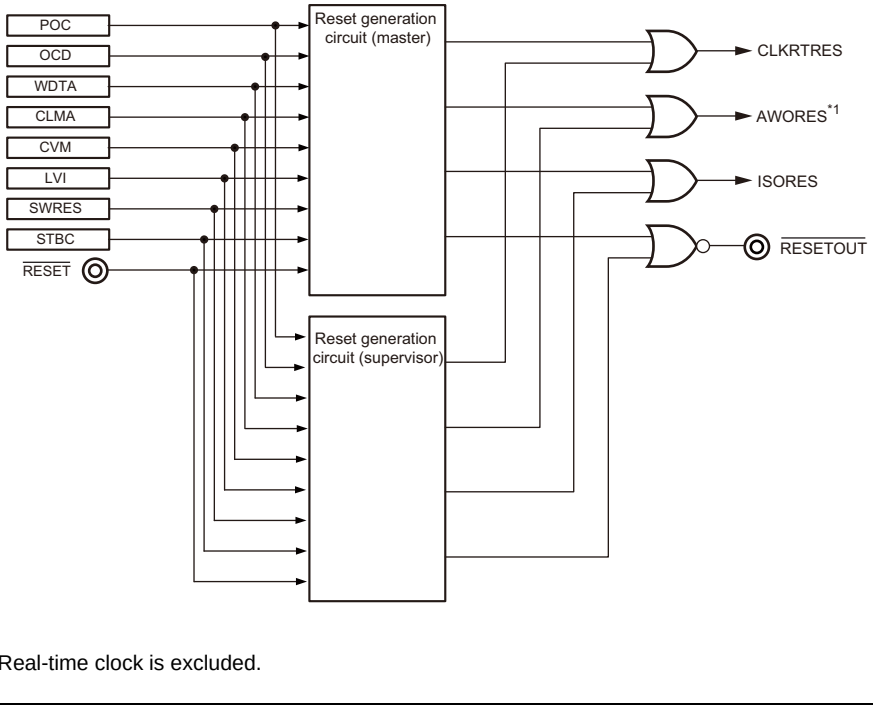
Reset level 2: For a quick return to normal operating mode by eliminating the oscillator stabilization time, initializes the entire microcontroller except for the clock generator and the real-time clock.

Reset level 3: At the transition to DEEPSTOP mode, initializes all the isolated areas (ISO).

8.1.2 Reset Controller Redundancy

This microcontroller includes double reset controllers. This enables initialization of the reset targeted area without failure even if one of the two reset controllers breaks down.

The configuration of the reset controller is shown in the figure below.



Note 1. Real-time clock is excluded.

Figure 8.1 Reset Controller Redundancy

At the generation of a reset, the same reset source signal is input to two reset controllers.

According to a reset source, two reset controllers 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 two reset controllers. Thus, a reset signal is generated normally even if one of the two reset controllers breaks down.

Whether a reset controller operates normally can be checked by reading and comparing the reset source registers of the respective reset controllers.

8.1.3 Reset Output ($\overline{\text{RESETOUT}}$)

When a reset source 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.

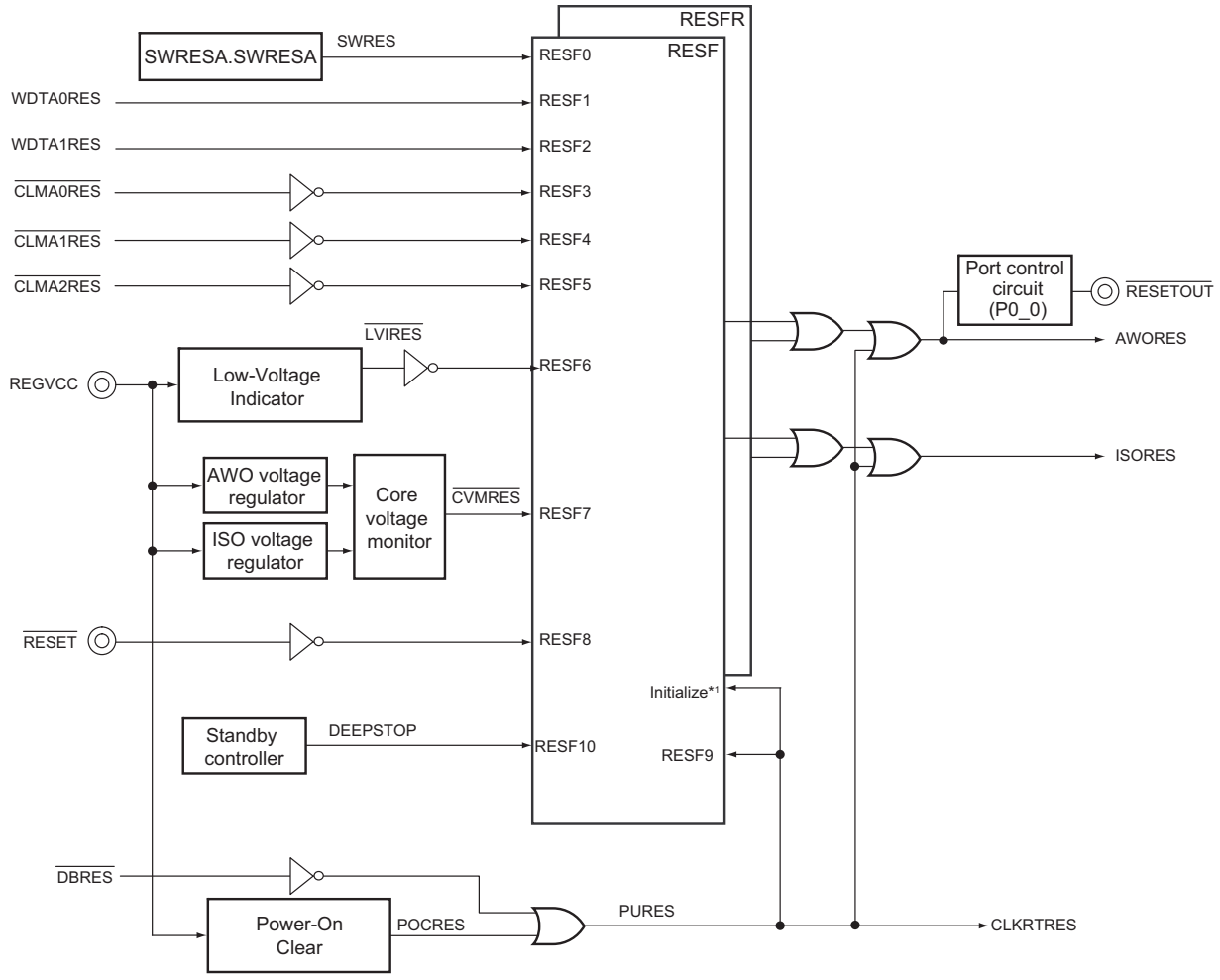
8.1.4 Reset Flag

To identify a reset source, two registers with a flag for each reset source are provided. The main configuration elements of the reset controller are shown in **Figure 8.2, Block Diagram of the Reset Controller**.

8.2 Configuration

8.2.1 Block Diagram

Block diagram of reset circuits are shown below.



Note 1. Registers RESF and RESFR are initialized by PURES.
After an initialization, bits RESF9 and RESFR9 are set to 1.

Figure 8.2 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, and core voltage monitor.
- 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 caused by the power-on clear and debugger reset sources.

At the generation of AWORES, all the clock generation circuits (excluding PLL) operating up to that time continue to operate (when MOSCSTPMSK is 1). However, at the generation of $\overline{\text{CLMA0RES}}$, HS IntOSC targeted for CLMA0 monitoring is reset. Also, at the generation of $\overline{\text{CLMA1RES}}$, MainOSC targeted for CLMA1 monitoring is reset.

At the generation of PURES, since the clock generation circuit is reset, the operation should be re-started after it is stopped once.

(2) Reset Flags

The reset source register (RESF) and the redundant reset source 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 are cleared by software.

For details, see **Section 8.1.4, Reset Flag**.

(3) On-Chip Modules Resets

(a) Watchdog Timers Reset

The watchdog timers 0, 1 can generate two types of resets: WDTA0RES and WDTA1RES.
For details, see **Section 8.4.6, 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 8.4.8, 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 8.4.9, 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 8.4.7, Software Reset**.

(5) Reset Output Signal

During reset and after reset release, port P0_0 outputs low level as RESETOUT function. After reset release, any change of the port P0_0 configuration (i.e., output level setting change by port register, operating mode specification change by port mode control register, etc.) invalidates the RESETOUT function.

(6) Power Supply Supervision

The following power supply detection circuits observe the level of the external power supply REGVCC.

(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 8.4.3, Low-Voltage Detection Circuit (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 goes below the defined limit.
For details, see **Section 8.4.2, 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 8.4.4, Core Voltage Monitor (CVM) Reset**.

(7) Masking of Reset Source at Debugging Mode

Masking of the following reset sources is enabled during debugging.

Table 8.2 Reset Sources to be Masked at Debugging

Reset Source	Masked/Not Masked
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 the transition to DEEPSTOP mode	—

8.3 Registers

This section contains a description of all registers of the reset controller.

8.3.1 Reset Controller Registers Overview

The reset controller is controlled and operated by the following registers:

Table 8.3 Reset Controller Registers Overview

Register Name	Symbol	Address
General reset flags registers		
Reset factor register	RESF	FFF8 0760$_H$
Reset factor clear register	RESFC	FFF8 0768$_H$
Redundant reset source register	RESFR	FFF8 0860$_H$
Redundant reset source clear register	RESFCR	FFF8 0868$_H$
Software reset control registers		
Software reset register	SWRESA	FFF8 0A04$_H$

NOTES

1. For the LVI related, RAM store related, and CVM related registers, see **Section 9, Supply Voltage Monitor**.
2. As for the protection registers, see **Section 4, Write-Protected Registers**.

8.3.2 Details of General Reset Flag Registers

8.3.2.1 RESF — Reset Factor Register

This register contains information about which type of resets occurred since 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 can only be read in 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	0	0	0	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 8.4 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 8.4 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

8.3.2.2 RESFC — Reset Factor Clear Register

This register clears the reset flags of the RESF register.

Access: This register can only 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 8.5 RESFC Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When writing to these bits, write “0”.
10	RESFC10	Reset flag clear by DEEPSTOP mode 0: Do not clear flag 1: Clear reset 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 reset 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

8.3.2.3 RESFR — Redundant Reset Source Register

This register is a duplication of the reset factor flag register. This register is initialized by a power-up reset PURES.

In accordance with the setting conditions for each bit in the reset factor flag register, the same bits are set in this register.

Access: This register can only 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	0	0	0	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 8.6 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 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: Do not clear flag 1: Clear flag
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 8.6 RESFR Register Contents (2/2)

Bit Position	Bit Name	Function
0	RESFR0	Software reset flag 0: No reset occurred 1: Reset has occurred

8.3.2.4 RESFCR — Redundant Reset Source Clear Register

This register clears the reset flags of the RESFR.

Access: This register can only 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 8.7 RESFCR Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When writing to these bits, write “0”.
10	RESFCR10	Reset flag clear by DEEPSTOP 0: Do not clear reset flag 1: Clear reset 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 reset flag 1: Clear reset 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

8.3.3 Details of Software Reset Control Registers

8.3.3.1 SWRESA — Software Reset Register

This register is used to generate a software reset SWRES. Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.

For details, see **Section Section 4, Write-Protected Registers**.

Access: This register can only 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 8.8 SWRESA Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing to these bits, write “0”.
0	SWRESA	Software reset trigger 0: No Software reset trigger is generated. 1: Software reset trigger is generated.

8.4 Functional Description

8.4.1 Reset Flags

The reset source register (RESF) and the redundant reset source register (RESFR) provide reset flags for each reset source.

If a reset has occurred, the assigned 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 the initialization at the generation of PURES). In addition, flags in RESF and RESFR are cleared by the reset source clear register (RESFC) and the redundant reset source clear register (RESFRC).

Each reset source can set the corresponding flag independently from other reset sources.

RESF9 and RESFR9 are set when PURES is generated.

8.4.2 Power-On-Clear (POC) Reset

The power-on-clear circuit (POC) permanently 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 the defined limit.

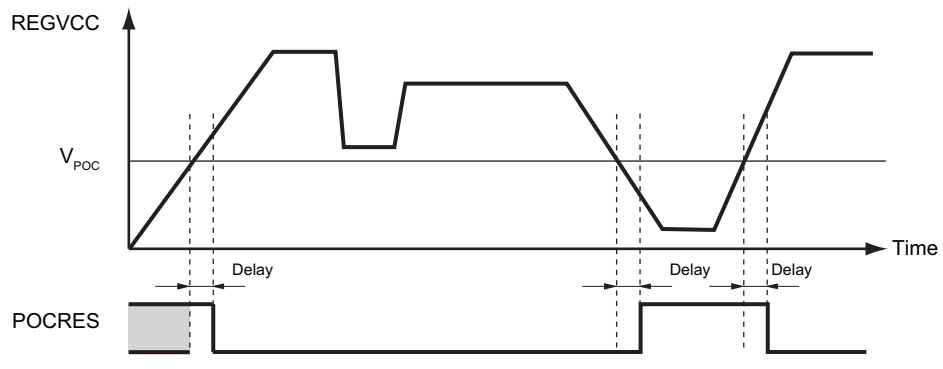
If REGVCC falls below the internal reference voltage (REGVCC < V_{POC}), the internal reset signal POCRES and a power-up reset PURES are generated.

For details on the specification of the internal voltage reference level V_{POC}, see the Data Sheet.

The reset source register (RESF) and the redundant reset source register (RESFR) are cleared by the power-on clear reset. RESF9 and RESFR9 are set to 1 after the 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:
A delay time is induced between REGVCC crossing the V_{POC} level and assertion of POCRES. See the Data Sheet for the delay time.

Figure 8.3 POC Reset Timing

(1) Overview of CPU System Startup after Power-On-Clear

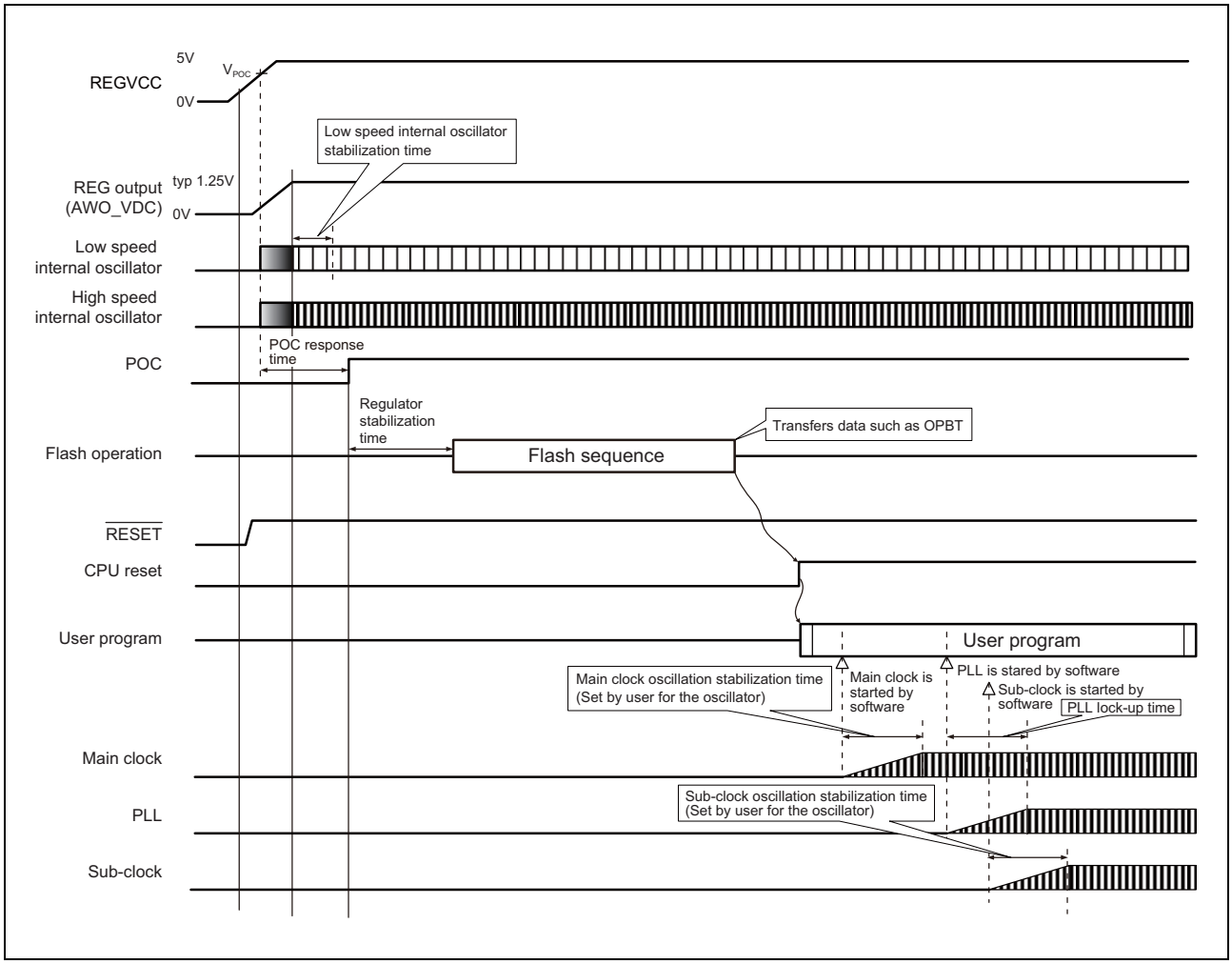


Figure 8.4 When $\overline{\text{RESET}}$ is Released before Execution of Flash Sequence

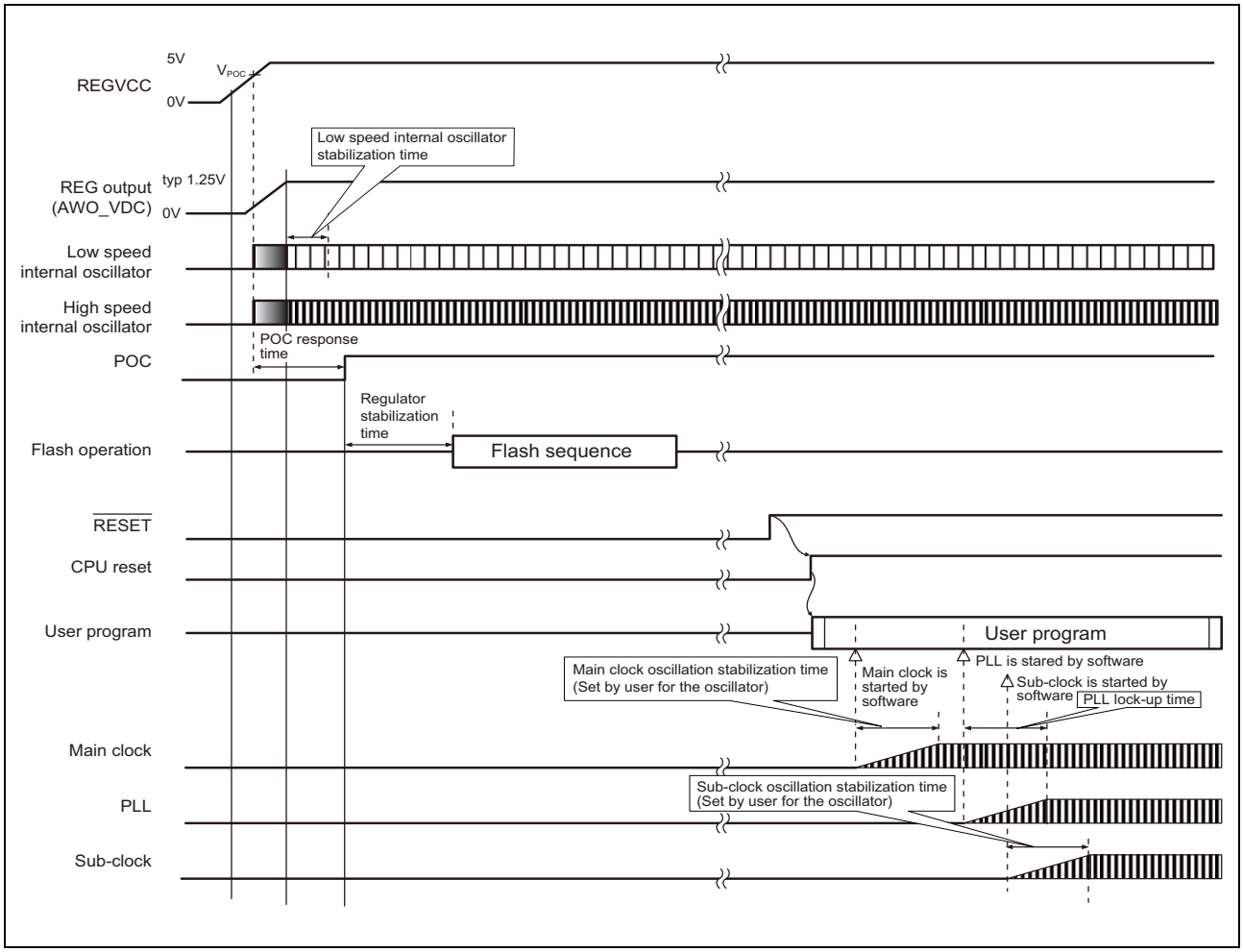


Figure 8.5 When $\overline{\text{RESET}}$ is Released after Execution of Flash Sequence

8.4.3 Low-Voltage Detection Circuit (LVI) Reset

The low-voltage indicator circuit (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.

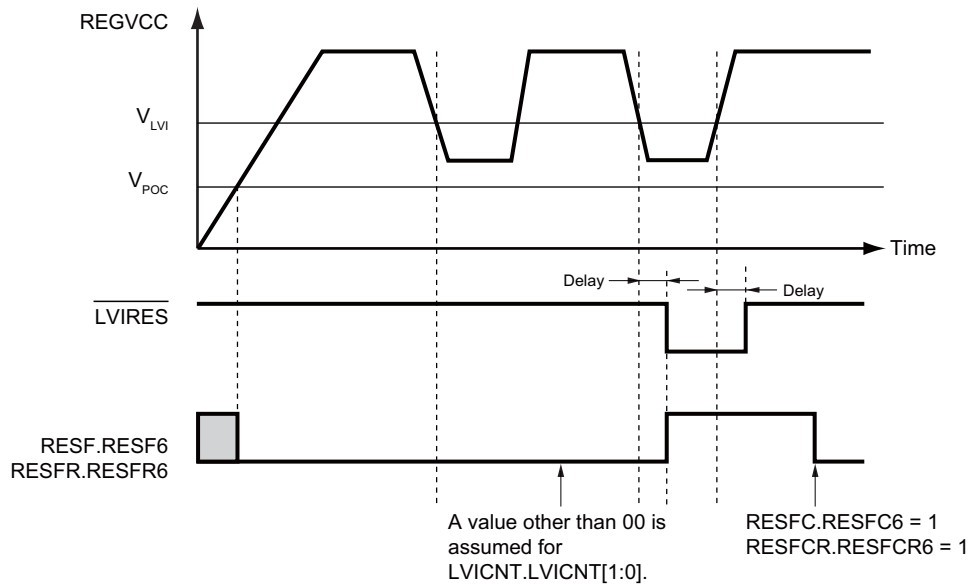
Additionally, the $\overline{\text{LVIRES}}$ flags (bits RESF.RESF6 and RESFR.RESFR6) are set.

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 9.2, Registers**.

The following figure illustrates the timing of a $\overline{\text{LVIRES}}$ and bits RESF.RESF6 and RESFR.RESFR6.



Note: Delay:
A delay time is induced between REGVCC crossing the V_{LVI} level and assertion of $\overline{\text{LVIRES}}$ which sets bits RESF.RESF6 and RESFR.RESFR6.
See the Data Sheet for the delay time.

Figure 8.6 LVI Reset Timing

8.4.4 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.

- Set 1 to the RESFC.RESFC7 bit to clear RESF.RESF7 bit.
  Set 1 to the RESFCR.RESFCR7 to clear RESFR.RESFR7 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

In addition, if the core voltage exceeds the specified voltage, the power supply to the ISO area is stopped. Once $\overline{\text{CVMRES}}$ (high detection) occurs, it is necessary to cancel the internal reset state by inserting an external reset ($\overline{\text{RESET}}$) because the microcontroller holds the reset state.

For details on the CVM function, see **Section 9, Supply Voltage Monitor**.

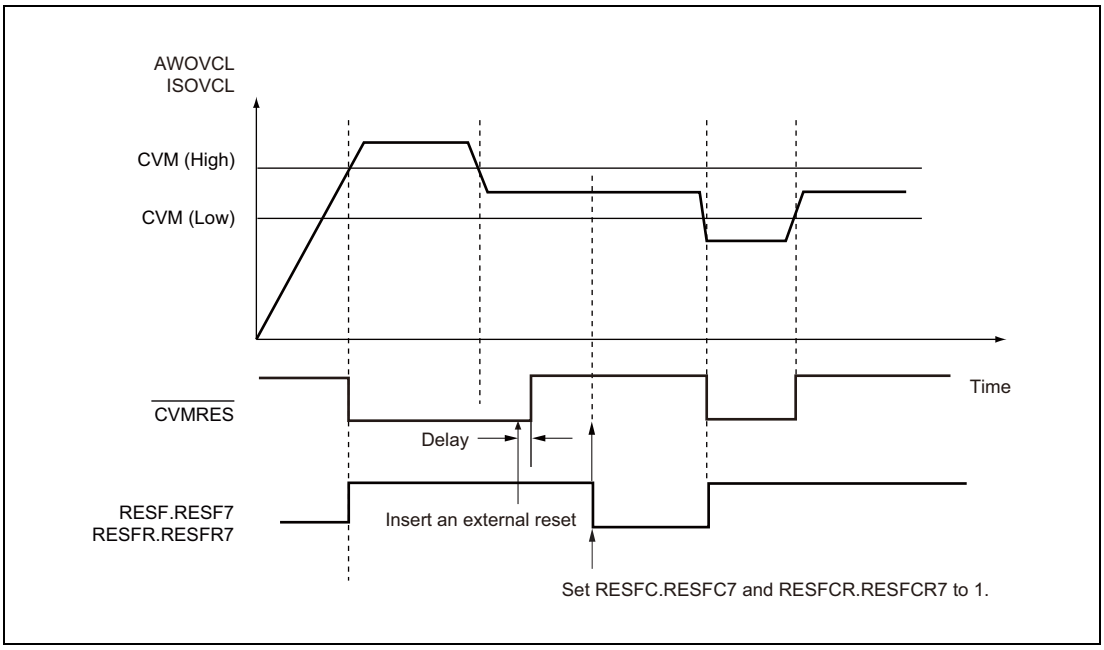


Figure 8.7 CVM Reset Timing

8.4.5 External Reset ($\overline{\text{RESET}}$)

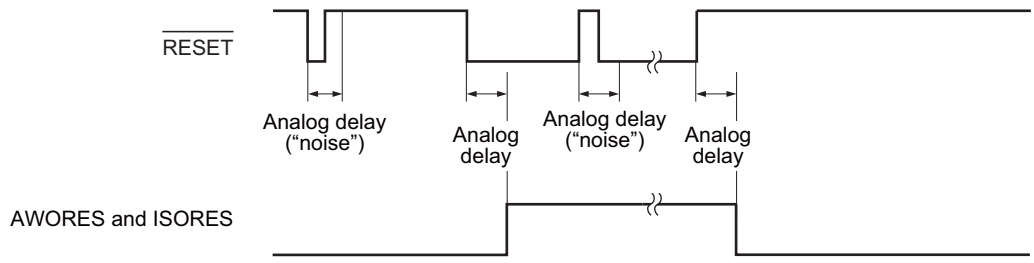
Reset is performed when a low level signal is applied to the $\overline{\text{RESET}}$ pin setting bits RESF.RESF8 and RESFR.RESFR8.

After that, bits RESF.RESF8 and RESFR.RESFR8 are not cleared automatically, even if $\overline{\text{RESET}}$ becomes inactive. Bits RESF.RESF8 and RESFR.RESFR8 are cleared as described below.

- Setting the RESFC.RESFC8 bit to 1 clears the RESF.RESF8 bit and 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.



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 8.8 External Reset ($\overline{\text{RESET}}$)

8.4.6 Watchdog Timer (WDTA) Reset

The watchdog timers can be configured to generate a reset if the overflow time expires. After watchdog timer reset, 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, respectively) 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) enters the inactive state.

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 $\overline{\text{DBRES}}$)

8.4.7 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 $\overline{\text{DBRES}}$)

8.4.8 Clock Monitor (CLMA) Reset

The clock monitors can generate the following resets:

- $\overline{\text{CLMA0RES}}$, when frequency abnormality of HS IntOsc is detected
- $\overline{\text{CLMA1RES}}$, when frequency abnormality of MainOsc is detected
- $\overline{\text{CLMA2RES}}$, if frequency abnormality of 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 CLM0RES, CLM1RES, and 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.

- As for $\overline{\text{CLMA0RES}}$:
  Setting the RESFC.RESFC3 bit to 1 clears the RESF.RESF3 bit.
  Setting the RESFCR.RESFCR3 bit to 1 clears the RESFR.RESFR3 bit.
- As for $\overline{\text{CLMA1RES}}$:
  Setting the RESFC.RESFC4 bit to 1 clears the RESF.RESF4 bit.
  Setting the RESFCR.RESFCR4 bit to 1 clears the RESFR.RESFR4 bit.
- As for $\overline{\text{CLMA2RES}}$:
  RESFC.RESFC5 bit to 1 clears the RESF.RESF5 bit.
  RESFCR.RESFCR5 bit to 1 clears the RESFR.RESFR5 bit.
- Power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

8.4.9 Debugger Reset

Debugger reset ($\overline{\text{DBRES}}$) is generated via a debugger command. $\overline{\text{DBRES}}$ activates PURES, thus operates in the same way like the power-on-clear reset POCRES:

- It resets clock generators. Hence these clock generators stop operation and must be restarted.
- The reset source register RESF and the redundant reset source register RESFR are cleared (Bits RESF9 and RESFR9 are set to 1 after initialization.).

Section 9 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.

9.1 Overview

9.1.1 Functional Overview

This product monitors continuously 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 9.1 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.

9.1.2 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 8.4.2, Power-On-Clear (POC) Reset**.

9.1.3 Low Voltage Indicator Circuit (LVI)

The LVI continuously compares the external power supply voltage REGVCC with the reference voltage V_{LVI}.

If REGVCC falls below the reference voltage (REGVCC < V_{LVI}), an internal reset signal or interrupt request signal is generated.

9.1.3.1 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 specification of the reference voltage level V_{LVI}, see **Section 9.2.2.1, LVICNT — LVI Control Register**.

9.1.3.2 LVI Reset (LVIRES)

When setting the LVI detection voltage and releasing the LVIRESMK, if REGVCC falls below the reference voltage (REGVCC < V_{LVI}), the internal reset signal $\overline{\text{LVIRES}}$ is generated.

For the specification on generation of $\overline{\text{LVIRES}}$, see **Section 8.4.3, Low-Voltage Detection Circuit (LVI) Reset**.

9.1.3.3 LVI Interrupt (INTLVIL / INTLVIH)

At the time of LVI detection voltage setting and LVIRESMK setting, if REGVCC falls below the reference voltage (REGVCC (MIN) < V_{LVI}), the LVI interrupt INTLVIL is generated.

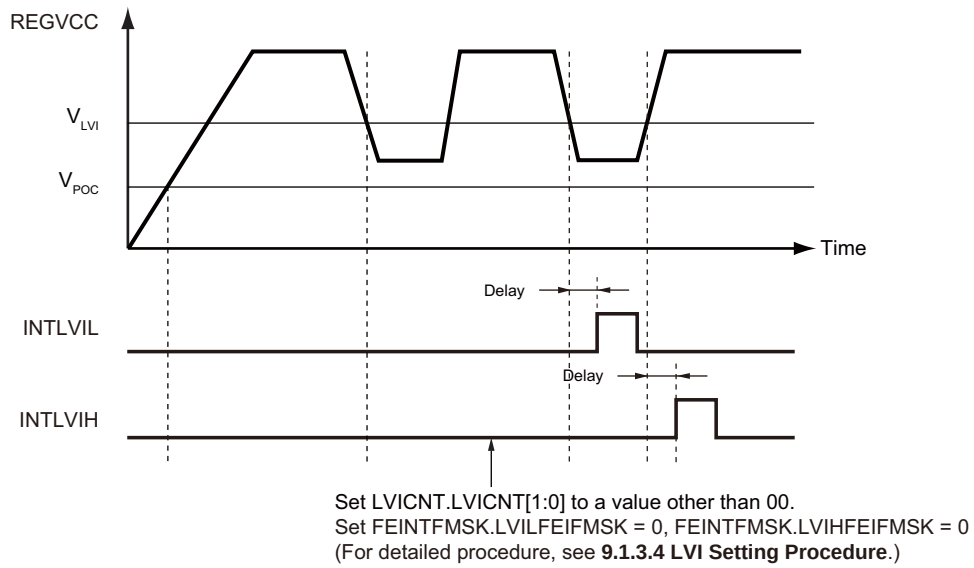
To use the LVI as an interrupt source, the INTLVIL interrupt must be unmasked.

INTLVIL interrupt can be used as wake-up source from all of standby modes. For details, see **Section Section 11, Stand-By Controller (STBC)**.

At the time of LVI detection voltage setting and LVIRESMK setting, and 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:
INTLVIL / INTLVIH is generated after REGVCC crosses the V_{LVI} level.
See the Data Sheet for delay time.

Figure 9.1 INTLVIL / INTLVIH Generation Timing

9.1.3.4 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 4, Write-Protected Registers**.

CAUTION

If REGVCC is not stable around the LVI detection level (VLVI), correct judgment of whether INTLVIH or INTLVIL interrupt processing should proceed may not be possible. Please take care on this point.

Accordingly, the software must be configured so that LVI detection interrupt processing is completed before a next LVI detection. Also, consider control of REGVCC etc.

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 VLVI, the software erroneously judges that REGVCC (min.) < VLVI.

9.1.4 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 microcomputer.

If the regulator output voltage comes out of 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.

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.

9.1.4.1 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 specification on generation of $\overline{\text{CVMRES}}$, see **Section 8.4.4, Core Voltage Monitor (CVM) Reset**.

9.1.4.2 CVM Setting

Use the option byte to enable the high-voltage monitor and the low-voltage monitor. For details, see **Section 35.11, Option Bytes**.

9.1.4.3 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.

(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 4, Write-Protected Registers**.

9.1.5 RAM Retention Voltage Indicator (Very-Low-Voltage Detection Circuit)

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 specification of the RAM retention voltage level V_{VLVI}.

9.1.5.1 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.

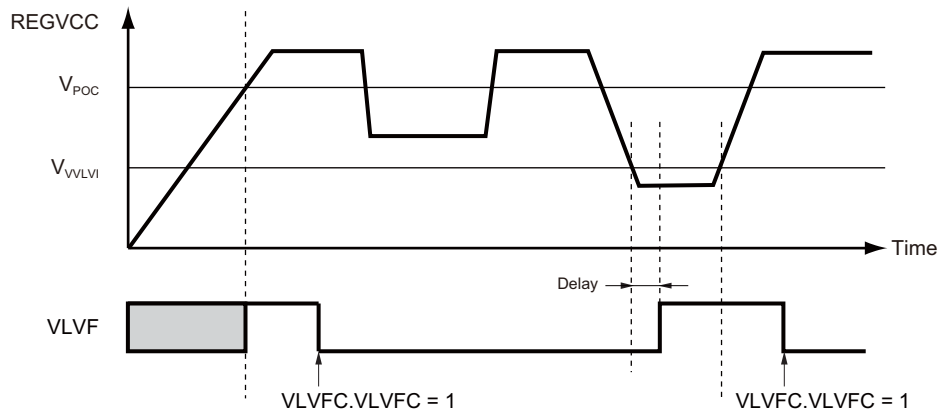
If REGVCC falls below V_{VLVI}, the RRAM content cannot be guaranteed. Thus the entire RRAM must be restored before continuing operation.

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:
A delay time is induced between REGVCC crossing the V_{VLVI} level and setting the VLVF bit to 1. See the Data Sheet for delay time.

Figure 9.2 VLVF Operation Timing

9.1.6 Block Diagram

The block diagram of the supply voltage monitor is shown below.

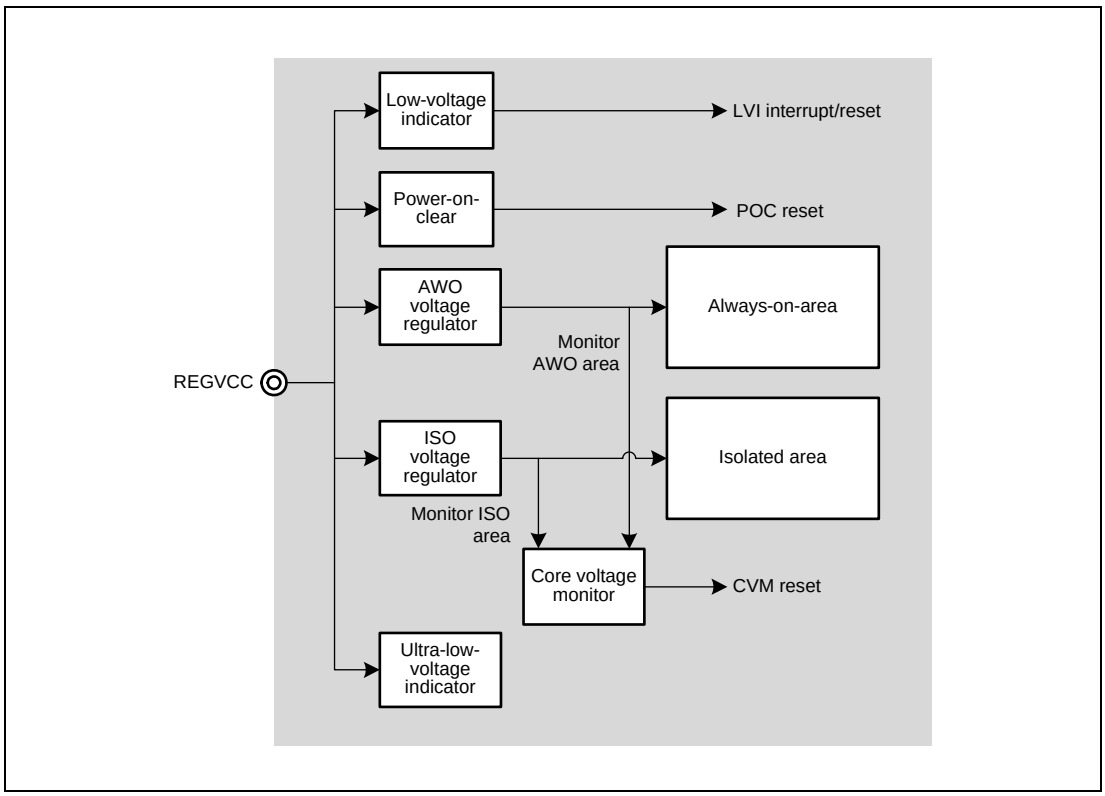


Figure 9.3 Supply Voltage Monitor

9.2 Registers

9.2.1 List of Registers

The following table lists the supply voltage monitor registers.

Table 9.2 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	FFF5 0000$_H$
CVM detection enable register	CVMDE	FFF5 0004$_H$
CVM diagnostic mode setting register	CVMDIAG	FFF5 0014$_H$
Very-low-voltage detection control register		
Very-low-voltage detection register	VLVF	FFF8 0980$_H$
Very-low-voltage detection clear register	VLVFC	FFF8 0988$_H$

9.2.2 Low-Voltage Indicator Reset Control Registers

9.2.2.1 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.

Writing to this register is protected by a special sequence of instructions. For details, see **Section Section 4, Write-Protected Registers**.

Access: This register can be read/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	LVICNTn[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 9.3 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	LVICNTn[1:0]	Detection Level 0 0: LVI is ignored 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.

9.2.3 Core Voltage Monitor Control Registers

9.2.3.1 CVMF — CVM Factor Register

This register records the core voltage failure state generated after the last POC reset.

When 1 is set to the L_V_F or H_V_F bit of this register, the bit is not updated until it is initialized by writing 0 to the Power-On-Clear or each bit (CVMF.L_V_F, CVMF.H_V_F). However, it continuously monitors an error signal from the core voltage monitoring circuit in diagnostic mode.

Writing to this register is protected by a special sequence of instructions. For details, see **Section 4.1.5, Write-Protection Target Registers**.

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

Address: FFF5 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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 9.4 CVMF Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	The write value should always be 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

9.2.3.2 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 can be read/written in 32-bit units.

Address: FFF5 0004_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 35.11, Option Bytes**.

Table 9.5 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: Disable high-voltage detection 1: Enable high-voltage detection
0	L_D_E	Low-Voltage Monitor Enable 0: Disable low-voltage detection 1: Enable low-voltage detection

9.2.3.3 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 9.1.4.3, Diagnostic (DIAG) Mode**.

Writing to this register is protected by a special sequence of instructions. For details, see **Section 4.1.5, Write-Protection Target Registers**.

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

Address: FFF5 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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 9.6 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

9.2.4 Very-Low-Voltage Detection Control Registers

9.2.4.1 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 can only be read in 32-bit units.

Address: FFF8 0980_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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	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 9.7 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 **9.1.5.1, Retention RAM Content Retention**

9.2.4.2 VLVFC — Very-Low-Voltage Detection Clear Register

This register clears the VLVF.VLVF bit.

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

Address: FFF8 0988$_H$

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	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLVFC
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 9.8 VLVFC Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	VLVFC	Clear VLVF.VLVF bit. 0: Do not clear 1: Clear

Section 10 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/F1L microcontrollers. The ensuing sections describe the clock oscillation circuit, clock output function, and clock monitor function that make up the clock controller.

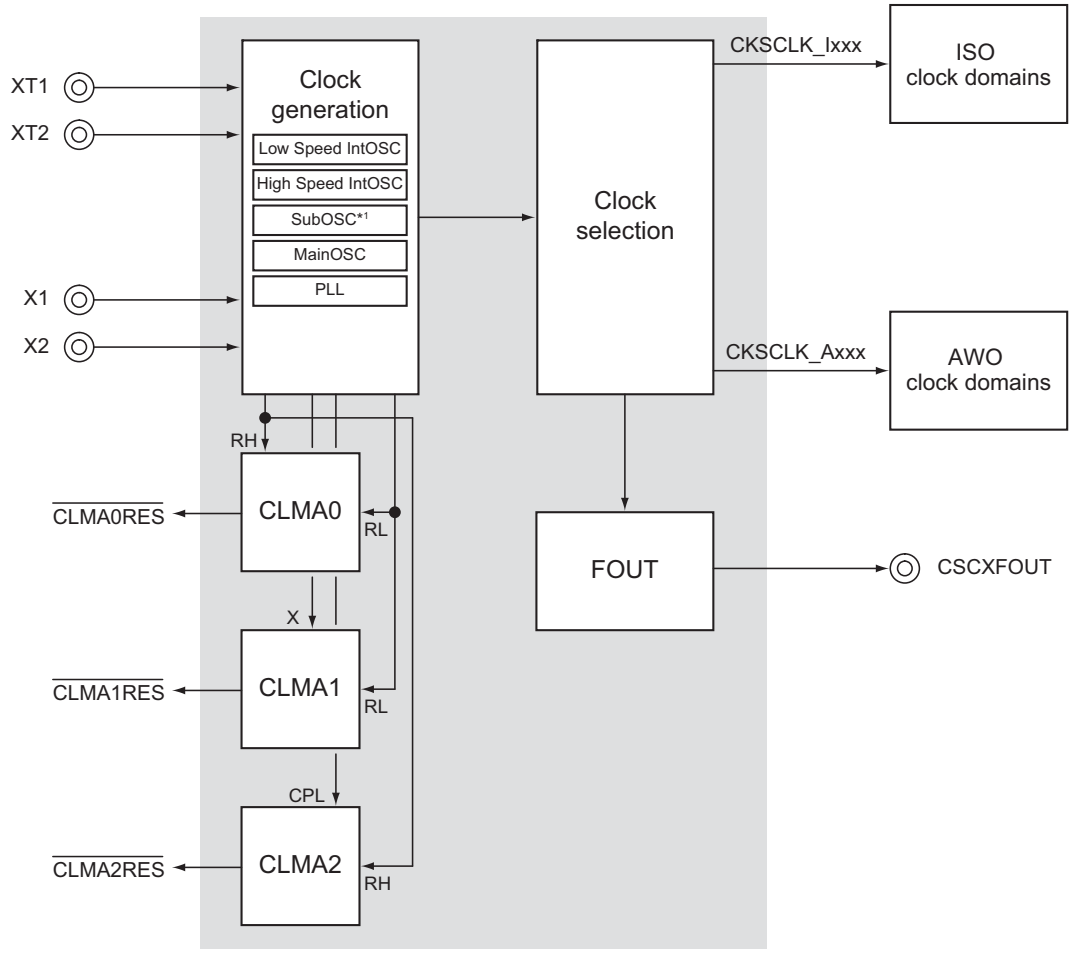
10.1 Features of Clock Controller of RH850/F1L

The clock controller of the RH850/F1L microcontrollers has the following features.

- Five 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*1
  - 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
- Fine management of clock supply to peripheral modules through clock domains
- On-chip clock monitor that detects of CPU clock anomalies of the main oscillation and high-speed internal oscillation during PLL use
- CPU internal clock output (FOUT)

Note 1. The sub-oscillation circuit is supported only for the 144-pin and 176-pin products.

Figure 10.1 shows the schematic diagram of the clock controller.



Note 1. Supported only for 144-pin and 176-pin products

Figure 10.1 Clock Controller Overview

10.2 Configuration of Clock Controller

This section describes the configuration of the clock controller.

The clock controller is configured of a clock oscillator 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 10.2 shows the configuration of the clock controller.

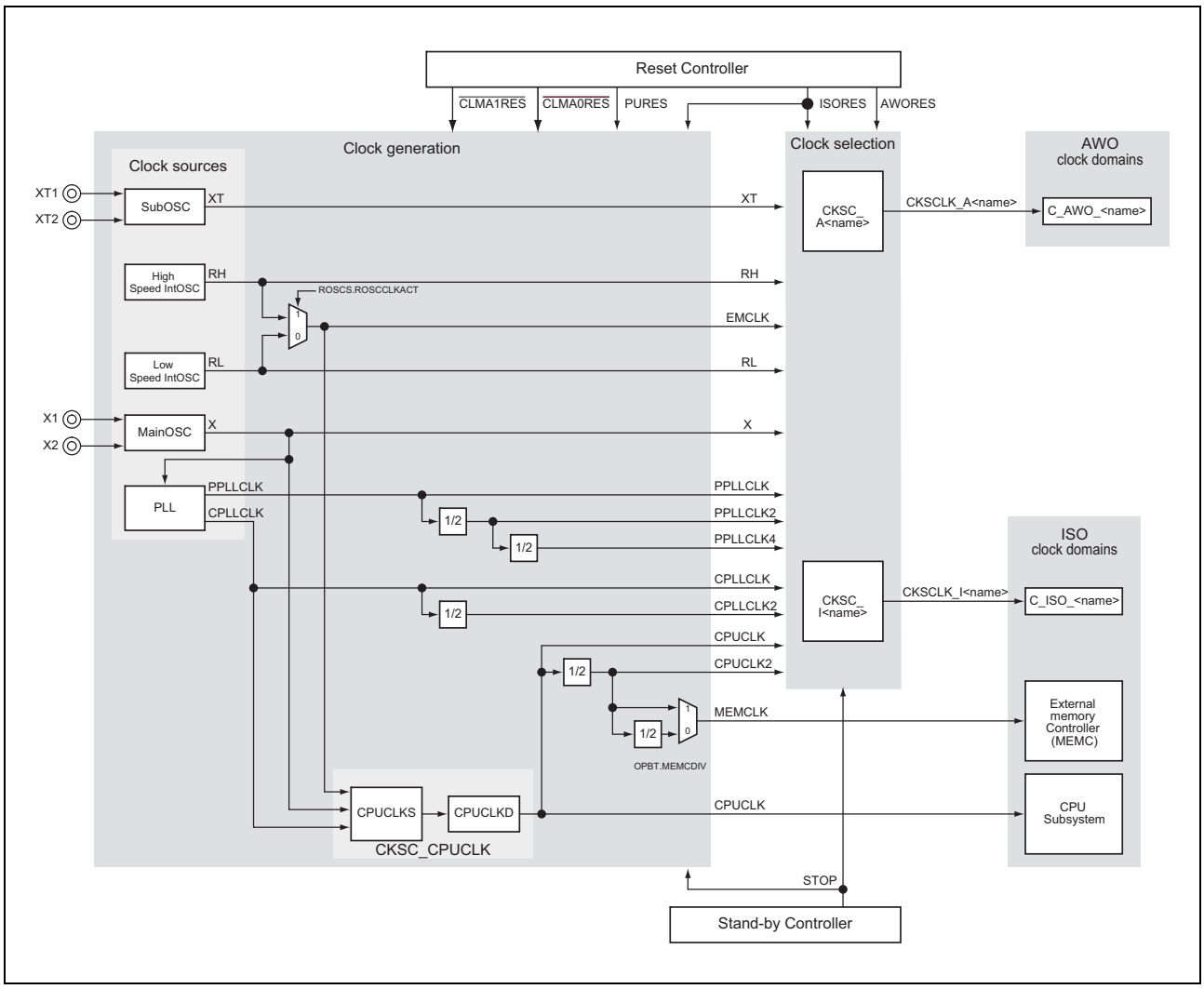


Figure 10.2 Clock Controller Structure

NOTE

Clock domain and clock control register naming conventions

The clock signals, their control registers, etc., follow the following naming conventions, that reflect their membership to a certain power domain and clock domain. 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>: clock selector for Always-On-Area clock selectors
  - CKSC_I<name>: clock selector for Isolated-Area clock selectors
- Clock selector registers:
  - CKSC_A<name>S_CTL: Always-On-Area source clock selector registers
  - CKSC_A<name>D_CTL: Always-On-Area source clock divider registers
  - CKSC_I<name>S_CTL: Isolated-Area source clock selector registers
  - CKSC_I<name>D_CTL: Isolated-Area source clock divider registers

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, Isolated_Area
Indicates the power supply domain. Always-On-Area (AWO) is an always-on power supply, and the Isolated_Area (ISO) is an a power supply that is switched on or off by the operation mode. For details, see **Section 38, Power Supply and Power Domains**.

10.2.1 Clock Generation Circuits

Five clock oscillators are provided:

Four clock oscillators are located on the Always-On-Area (AWO) and PLL is located on the Isolated-Area (ISO).

Main Oscillator (MainOSC)

The MainOSC generates the clock X.
Generation of the clock X requires the connection of an external resonator to X1 and X2.
The clock X is used as the reference clock for the PLL.

Sub Oscillator (SubOSC)

The SubOSC generates the sub-clock XT, which runs at a frequency of typical 32.768 kHz. 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)

This oscillator 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.

PLL

The PLL circuits generate all high speed operation clocks CPLLCLK and PPLLCLK for normal operation of the microcontroller.

The clock generation circuits generate the clocks generated by clock oscillators (X, XT, RH, RL, CPLLCLK, PPLLCLK) and their frequencies (CPLLCLK2, PPLLCLK2, PPLLCLK4).

10.2.2 Clock Selection

The clocks, generated by the clock oscillators, are input to the clock selectors CKSC_A<name>/ CKSC_I<name>.

Separate clock selector registers and partly additional clock dividers are provided for each domain clock CKSCLK_A<name>/CKSCLK_I<name>:

- CKSC_I<name>S_CTL/CKSC_I<name>D_CTL registers: determine the clock for the Isolated-Area clock domains.
- CKSC_A<name>S_CTL/CKSC_A<name>D_CTL registers: determine the clock for the Always-On-Area clock domains.

Note that not all available clocks from the clock oscillators are input to each clock selector.

The following clocks are supplied to the CPU and related modules from the clock generation circuit.

Emergency Clock (EMCLK)

The emergency clock EMCLK is supplied by the

- HS IntOSC, if it is active
- LS IntOSC, if the High Speed IntOSC is inactive

The selection is done automatically, so in case if the HS IntOSC is stopped for any reason, vital modules of the microcontroller are still in operation, since the LS IntOSC can not be stopped.

CPU Subsystem Clock (CPUCLK)

The CPU Subsystem clock CPUCLK is derived from PLL 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 derived from CPUCLK.

External Memory Controller Clock (MEMCLK)

The CPUCLK clock divider provides the operating clock for the external memory controller (MEMCLK).

The division ratio is selected by a bit of the option bytes (OPBT.MEMCDIV).

- MEMCDIV = 0: $f_{MEMCLK} = f_{CPUCLK} / 2$
- MEMCDIV = 1: $f_{MEMCLK} = f_{CPUCLK} / 4$

10.2.3 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 10.5.3, Clock Domain Settings**.

10.2.4 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 10.3, Clock Oscillators**.

CAUTION

For the specifications of the frequencies, acceptable variation, and other parameters of the clock generators, see the Data Sheet.

10.3 Clock Oscillators

10.3.1 Main Oscillator (MainOSC)

The Main Oscillator generates the clock X. X is also used as the PLL input clock PLLCLKIN.

Figure 10.3 shows the basic structure and signals of the MainOSC.

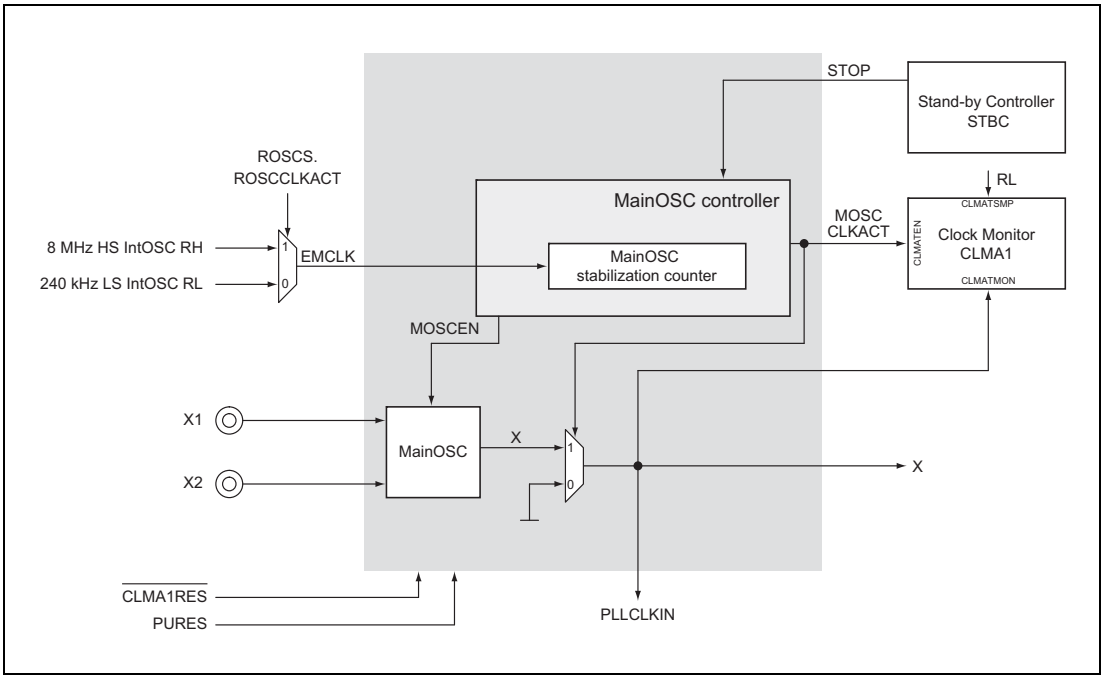


Figure 10.3 Main Oscillator (MainOSC)

MainOSC

After reset release, the MainOSC has been stopped. When the MainOSC is used, 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 range of settings is up to (2^{17} –1), and is in cycles of EMCLK.
As long as the MainOSC is not stable, the MOSCCLKACT signal disables the X output.
If the MainOSC stabilization counter has reached the value, defined by MOSCST.MOSCCLKST[16:0], X is judged as stable and the change of MOSCCLKACT from 0 to 1 makes X available.

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 from 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 in stand-by.
  If the MainOSC was in operation before stand-by, it is automatically re-started after wake-up from stand-by.
- MOSCSTPM.MOSCSTPMSK = 1:
  The STOP request signal is not masked, so the MainOSC continues to operate in 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 deactivated.

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, i.e. if MOSCCLKACT = 1.

10.3.2 Sub Oscillator (SubOSC)

The Sub Oscillator generates the sub clock XT. XT has usually a frequency of 32.768 kHz and is used for the Real-time Clock.

Figure 10.4 shows the basic structure and signals of the SubOSC.

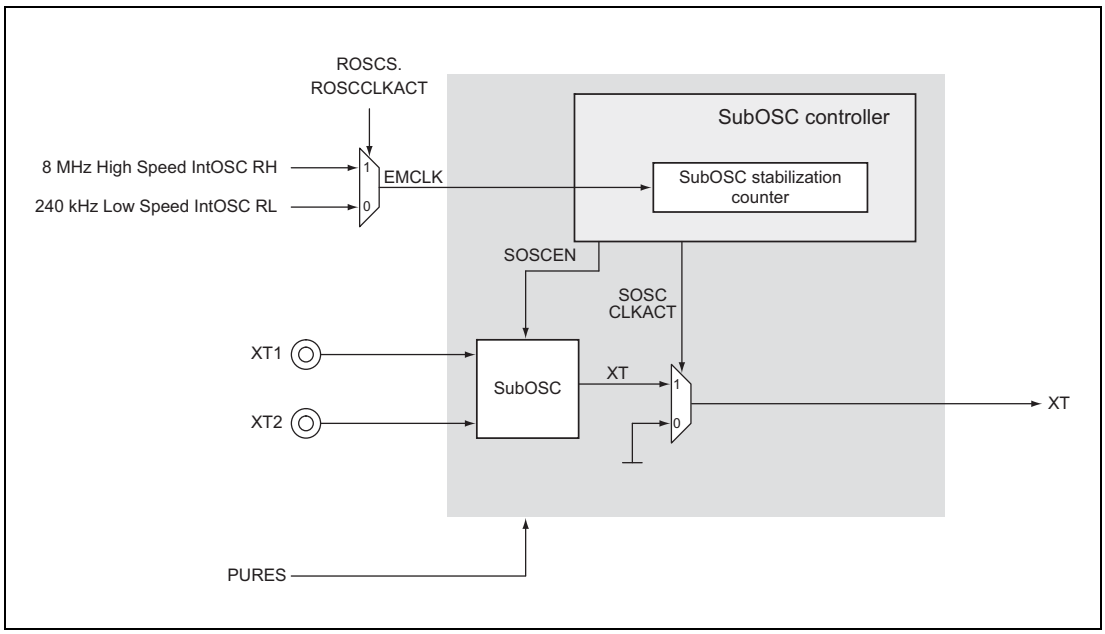


Figure 10.4 Sub Oscillator (SubOSC)

SubOSC enable

After reset release, SubOSC has been stopped. When SubOSC is used, 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.
If the SubOSC stabilization counter has reached the value specified by the SOSCST.SOSCCLKST[29:0] bits, XT is judged as stable and the change of SOSCCLKACT from 0 to 1 makes XT available.
Secure the stabilization time longer than 2 seconds.
Stable and active XT clock is indicated by SOSCS.SOSCCLKACT = 1.

SubOSC input frequencies

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.

- SubOSC is stopped when disenable trigger (SOSCE.SOSCDISTRG) is set to 1. Disenable trigger setting is enable only when SubOSC is in the active state (SOSCS.SOSCCLKACT = 1).

10.3.3 High Speed Internal Oscillator (HS IntOSC)

The High Speed Internal Oscillator generates the clock RH. RH has a nominal frequency of 8 MHz.

Figure 10.5 shows the basic structure and signals of the HS IntOSC.

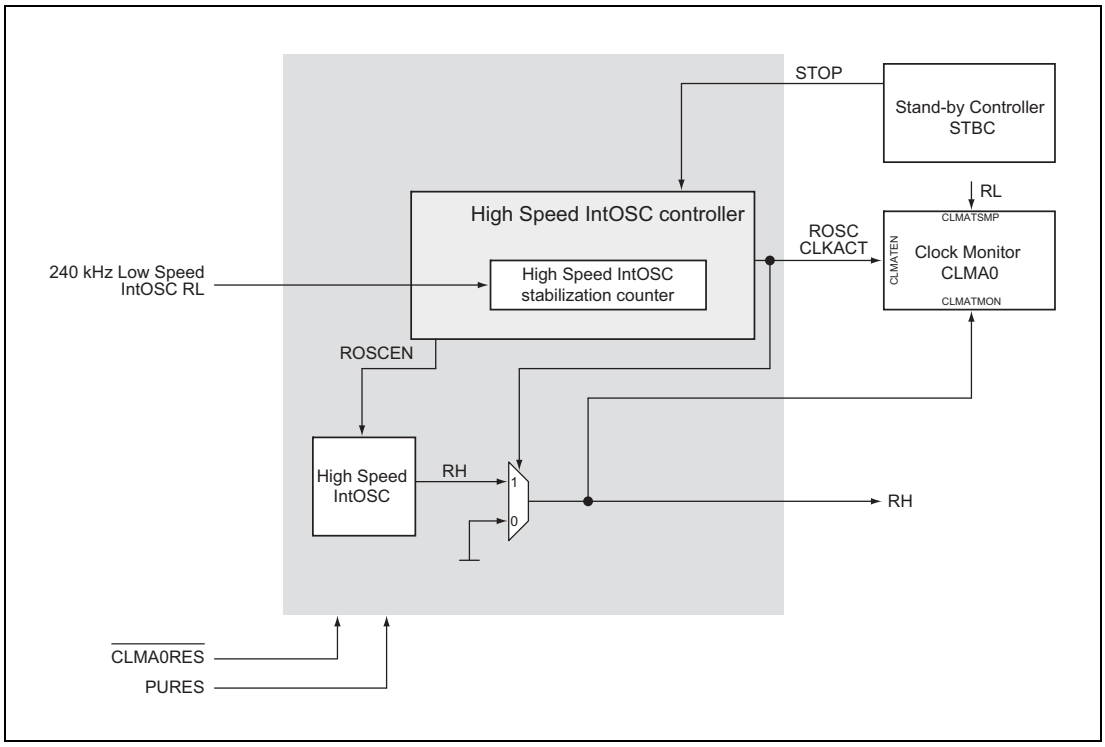


Figure 10.5 High Speed Internal Oscillator (HS IntOSC)

After reset release the HS IntOSC starts operation.

NOTE

The HS IntOSC can neither be stopped nor started by the application software. It can only be stopped in stand-by modes. On the other hand, when CLMA0 is reset, the IntOsc can be enabled to stop by the application software.

HS IntOSC stabilization

HS IntOSC outputs RH when it is stablized.
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 in stand-by and automatically restarted after wake-up from stand-by.

- ROSCSTPM.ROSCSTPMSK = 1:
  The STOP request signal is masked, so the HS IntOSC continues to operate in 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 f_{RH} by CLMA0 is also deactivated.

The HS IntOSC clock RH is used as the sampling clock for Clock Monitor CLMA2.

10.3.4 Low Speed Internal Oscillator (LS IntOSC)

The Low Speed Internal Oscillator generates the clock RL. RL has a nominal frequency of 240 kHz.

Figure 10.6 shows the basic structure and signals of the LS IntOSC.

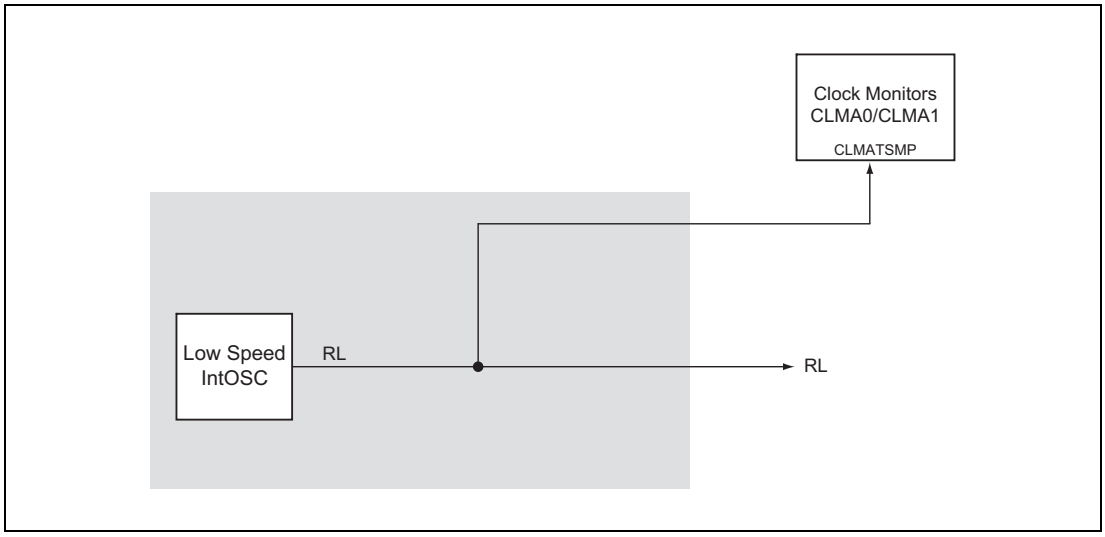


Figure 10.6 Low Speed Internal Oscillator (LS IntOSC)

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.

10.3.5 PLL

The Main Oscillator clock X is input to the Phase-Locked Loops (PLL) clock oscillator. The PLL output clocks CPLLCLK and PPLLCLK serve as the main operation clocks for the microcontroller.

Figure 10.7 shows the basic structure and signals of the PLL.

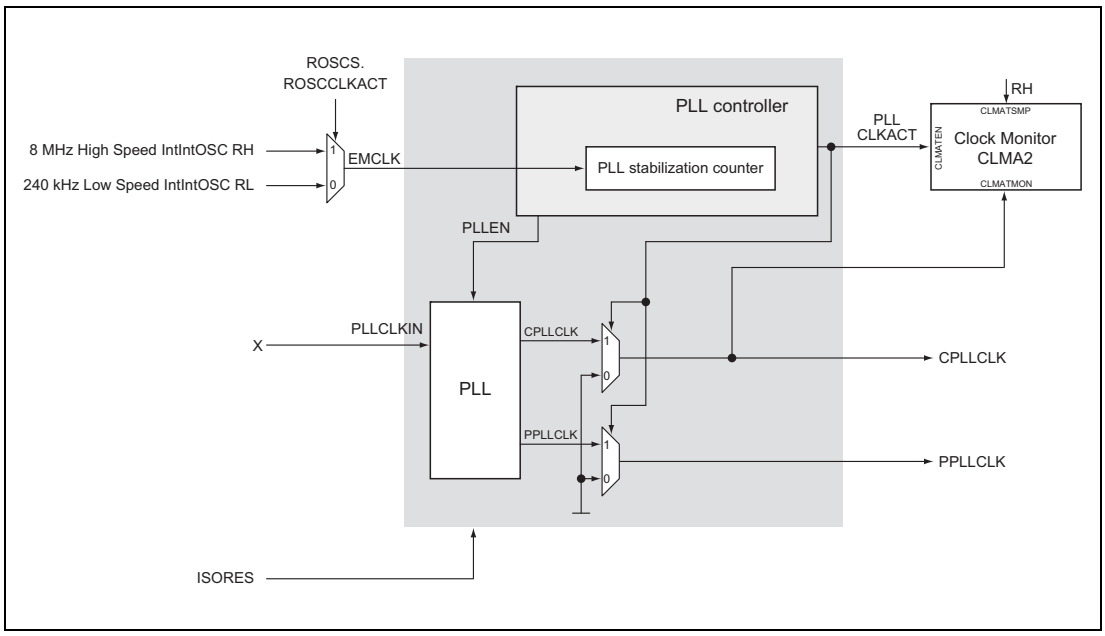


Figure 10.7 PLL

PLL enable

After reset release, the PLL has been stopped. When the PLL is used, set the PLL enable trigger bit (PLLE.PLLENTRG) to 1 to start the PLL.

PLL stabilization

The PLL stabilization counter starts counting the stabilization time.
As long as the PLL is not stable, the PLLCLKACT signal disables the PPLLCLK and CPLLCLK outputs.
If the PLL stabilization counter has reached a predefined value, PPLLCLK and CPLLCLK are judged as stable and the change of PLLCLKACT from 0 to 1 makes PPLLCLK and CPLLCLK available.

The stable and active state of the PPLLCLK and CPLLCLK clocks is indicated by PLLS.PLLCLKACT = 1.

PLL in stand-by modes

In STOP mode, the PLL is automatically stopped and resumes operation after wake-up from STOP mode, if it was operating before a stop entry.

The PLL is also automatically disabled when the mode is shifted to DEEPSTOP mode. After restoring from DEEPSTOP mode, the PLL should be reconfigured.

Clock Monitor control

The PLL activity signal PLLCLKACT enables or disables supervision by the Clock Monitor CLMA2. In case the PLL is inactive (PLLCLKACT = 0), supervision of the output clocks PPLLCLK and CPLLCLK by CLMA2 is also deactivated.

PLL enable/disable trigger

The PLL can be enabled and disabled by the enable and disable trigger control bits:

- Enable trigger PLLE.PLLENTRG = 1 starts the PLL
  Note that setting the enable trigger is only effective if the PLL is inactive, i.e. if PLLS.PLLCLKACT = 0.
- Disable trigger PLLE.PLLDISTRG = 1 stops the PLL
  Note that setting the disable trigger is only effective if the PLL is active, i.e. if PLLS.PLLCLKACT = 1.

10.3.5.1 PLL Parameters

The PLL is configured by a set of parameters, derived from the control register PLLC.

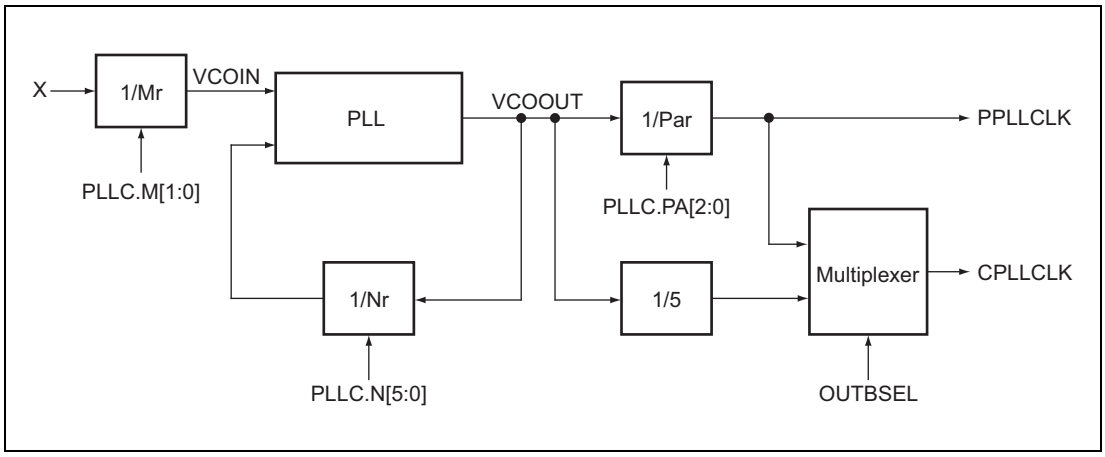


Figure 10.8 PLL Circuit

CPLLCLK and PPLLCLK

The frequency PPLLCLK is the source of the clock supply for several functional modules and must not exceed 80 MHz.

The frequency CPLLCLK is also the source of the CPU Subsystem clock CPUCLK.

$f_{PPLLCLK}$ and $f_{CPLLCLK}$ are the specified fraction of the integer of the VCO output frequency f_{VCOOUT}.

$$f_{VCOOUT} = f_X \times (Nr / Mr)$$

When $f_{CPLLCLK}$ is to be set at 96 MHz, the $f_{CPLLCLK}$ divider must be set with a dedicated divider (1/5) by setting PLLC.OUTBSEL to 1. The frequency $f_{PPLLCLK}$ and $f_{CPLLCLK}$ are calculated as follows:

When PLLC.OUTBSEL is 0:

$$f_{PPLLCLK}, f_{CPLLCLK} = f_X \times (Nr / Mr) \times 1/Par = f_{VCOOUT} \times 1/Par$$

When PLLC.OUTBSEL is 1:

$$f_{PPLLCLK} = f_X \times (Nr / Mr) \times 1/Par = f_{VCOOUT} \times 1/Par$$
$$f_{CPLLCLK} = f_X \times (Nr / Mr) \times 1/5 = f_{VCOOUT} \times 1/5$$

The values Nr, Mr, and Par are derived from PLLC register bits:

- Nr = PLLC.N[5:0] + 1
- Mr = PLLC.M[1:0] + 1
  - The setting range of Mr: $1 \leq Mr \leq 3$
- Par is determined by PLLC.PA[2:0] according to the following table:

PLLC.PA[2:0]	Par
010_B	4
011_B	6
100_B	8
101_B	16

10.4 Registers

10.4.1 List of Registers

The registers of the clock controller are listed below.

Table 10.1 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$_H$
PLL enable register	PLLE	FFF8 9000$_H$
PLL status register	PLLS	FFF8 9004$_H$
PLL control register	PLLC	FFF8 9008$_H$
Clock selectors 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$_H$
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$_H$

Table 10.1 List of Clock Controller Registers (2/2)

Register Name	Symbol	Address
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$_H$

10.4.2 Clock Oscillator Registers

10.4.2.1 MOSCE — MainOSC Enable Register

This register is used to start and stop the MainOSC.

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

This register is initialized by the power-up reset signals PURES and $\overline{\text{CLMA1RES}}$.

Access: This register can be read/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 10.2 MOSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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. Before stopping MainOSC by MOSCDISTRG, check that there is no clock domain that has selected MainOSC.

10.4.2.2 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 can only 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 10.3 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	These bits are read as an undefined value.

10.4.2.3 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/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 10.4 MOSCC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	MOSC AMPSEL[1:0]	MainOSC Frequency Selection

MOSCAMPSEL[1:0]	Amplification Gain	Typical Frequency Range for External Resonator (Crystal Type Dependence)
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 has been stopped.

10.4.2.4 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/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 10.5 MOSCST Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing to these bits, 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 HS IntOSC active (ROSCS.ROSCCLKACT = 1): Stabilization time = MOSCCLKST[16:0] / f_{RH} • If HS IntOSC 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 has been stopped.

10.4.2.5 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/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 10.6 MOSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	MOSCSTPMSK	MainOSC Stop Request Mask 0: MainOSC stops operation in stand-by mode. 1: MainOSC continues operation in stand-by mode.

10.4.2.6 SOSCE — SubOSC Enable Register

This register is used to start and stop the SubOSC.

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0. For details, see **Section 4, Write-Protected Registers**.

This register is initialized by the power-up reset signal PURES.

Access: This register can be read/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 10.7 SOSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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. To stop SubOSC by SOSCDISTRG, confirm that SubOSC is not selected by any clock domain.

10.4.2.7 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 can only 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 10.8 SOSCS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	SOSCCLKACT	SubOSC Activation Status 0: SubOSC is inactive 1: SubOSC is active
1, 0	Reserved	These bits are read as an undefined value.

10.4.2.8 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/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 10.9 SOSCST Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing to these bits, 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_{RH}

NOTE

See the Data Sheet for information about the SubOSC stabilization time.

CAUTION

Set this register when SubOSC has been stopped.

10.4.2.9 ROSCE — HS IntOSC Enable Register

This register is used to stop the HS IntOSC operation.

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0. For details, see **Section 4, 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/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 10.10 ROSCE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, write the value after reset.

10.4.2.10 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 can only 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 10.11 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	These bits are read as an undefined value.

10.4.2.11 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/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 10.12 ROSCSTPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	ROSCSTPMSK	HS IntOSC Stop Request Mask 0: HS IntOSC stops operation in stand-by mode 1: HS IntOSC continues operation in stand-by mode

10.4.2.12 PLLE — PLL Enable Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

This register is initialized by the ISORES signal.

Access: This register can be read/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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLDIS TRG	PLLEN TRG
Value after reset	0	0	0	0	0	0	0	0	0	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 10.13 PLLE Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	PLLDISTRG	PLL Disable Trigger*1 0: No function 1: Stops PLL
0	PLLENTRG	PLL Enable Trigger*2 0: No function 1: Starts PLL

Note 1. Before stopping PLL by PLLDISTRG, check that there is no clock domain that has selected PLL.

Note 2. Before starting PLL by PLLENTRG, confirm that MainOSC is operating.

10.4.2.13 PLLS — PLL Status Register

This register provides active status information about the PLL.

This register is initialized by the ISORES signal.

Access: This register can only 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
	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLCLKACT	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	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 10.14 PLLS Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned.
2	PLLCLKACT	PLL Active Status 0: PLL is inactive 1: PLL is active
1, 0	Reserved	These bits are read as an undefined value.

10.4.2.14 PLLC — PLLC Control Register

This register is used to set the PLL output clock frequencies $f_{PPLLCLK}$ and $f_{CPLLCLK}$, shown is **Section 10.3.5.1, PLL Parameters**.

This register can only be written, if the PLL is disabled.

This register is initialized by the ISORES signal.

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

Address: FFF8 9008_H

Value after reset: 0001 $133B_H$

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	1
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 10.15 PLLC Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
16	OUTBSEL	• Other than ADVANCE, PREMIUM, and Gateway-1MB: Be sure to write 0 to this bit (the value after reset is 1). • ADVANCE, PREMIUM, and Gateway-1MB: $f_{CPLLCLK}$ selection bit 0: Same clock as $f_{PPLLCLK}$ 1: Clock dividing f_{VCOOUT} by 5
15 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12, 11	M[1:0]	

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	

Bit Position	Bit Name	Function
10 to 8	PA[2:0]	P Divider Selection

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 10.15 PLLC Register Contents (2/2)

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5 to 0	N[5:0]	Division ratio Nr is set. For N[5:0] example settings, see **Table 10.16, PLL Output Table**.

CAUTION

Set this register when PLL has been stopped.

Table 10.16 PLL Output Table

MainOSC = 8 MHz (1/2)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ 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	←

MainOSC = 8 MHz (2/2)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ Frequency (MHz)
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	←
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	6	1	1	1	0	1	1	60	80.00	96.00

MainOSC = 12 MHz (1/2)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ 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	←

MainOSC = 12 MHz (2/2)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ Frequency (MHz)
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	6	1	0	0	1	1	1	40	80.00	96.00

MainOSC = 16 MHz (1/3)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ 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	←

MainOSC = 16 MHz (2/3)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ Frequency (MHz)
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	←
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	←

MainOSC = 16 MHz (3/3)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ Frequency (MHz)
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	2	6	1	1	1	0	1	1	60	80.00	96.00

MainOSC = 20 MHz (1/2)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ 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	←
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	←

MainOSC = 20 MHz (2/2)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ Frequency (MHz)
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	2	6	1	0	1	1	1	1	48	80.00	96.00

MainOSC = 24 MHz (1/3)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ 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	←

MainOSC = 24 MHz (2/3)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ Frequency (MHz)
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	←
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	←

MainOSC = 24 MHz (3/3)

OUTBSEL	Mr	Par	N5	N4	N3	N2	N1	N0	Nr	$f_{PPLLCLK}$ Frequency (MHz)	$f_{CPLLCLK}$ Frequency (MHz)
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	←
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	6	1	0	0	1	1	1	40	80.00	96.00
1	3	6	1	1	1	0	1	1	60	80.00	96.00

10.4.3 Clock Selector Control Register

10.4.3.1 WDTA0 Clock Domain C_AWO_WDTA

(1) CKSC_AWDTAD_CTL — C_AWO_WDTA Divided Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.17 CKSC_AWDTAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 f_{RL} (240 kHz) / 128 (default) 10_B: LS IntOSC f_{RL} (240 kHz) / 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 Divided Clock Active Register

Access: This register can only 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. When the value becomes 01$_B$ after entering the active state.

Table 10.18 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]	Current active clock divider for C_AWO_WDTA

(3) CKSC_AWDTAD_STPM — C_AWO_WDTA Stop Mask Register

Access: This register can be read/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 “1” value after reset of bit 1.

Table 10.19 CKSC_AWDTAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, 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.

10.4.3.2 TAUJ Clock Domain C_AWO_TAUJ

(1) CKSC_ATAUJS_CTL — C_AWO_TAUJ Source Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.20 CKSC_ATAUJS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, 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 f_{RH} (8 MHz) (default) 010_B: MainOSC f_X 011_B: LS IntOSC f_{RL} (240 kHz) 100_B: PPLLLCLK2 Other than above: Setting prohibited

Note 1. Before shifting to DEEPSTOP mode, select one other than PPLLCLK2.

(2) CKSC_ATAUJS_ACT — C_AWO_TAUJ Source Clock Active Register

Access: This register can only 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 10.21 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]	Current active C_AWO_TAUJ source clock

(3) CKSC_ATAUJD_CTL — C_AWO_TAUJ Divided Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.22 CKSC_ATAUJD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, 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 Divided Clock Active Register

Access: This register can only 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 10.23 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]	Current active clock divider for C_AWO_TAUJ

(5) CKSC_ATAUJD_STPM — C_AWO_TAUJ Stop Mask Register

Access: This register can be read/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 “1” value after reset of bit 1.

Table 10.24 CKSC_ATAUJD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	ATAUJD STPMSK	0: Clock domain C_AWO_TAUJ is stopped in stand-by mode. 1: Clock domain C_AWO_TAUJ is not stopped in stand-by mode.

10.4.3.3 RTCA Clock Domain C_AWO_RTCA

(1) CKSC_ARTCAS_CTL — C_AWO_RTCA Source Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.25 CKSC_ARTCAS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing to these bits, write the value after reset.
1, 0	ARTCASCSID [1:0]	Source Clock Setting for C_AWO_RTCA 00_B: Disable (default) 01_B: SubOSC XT 10_B: MainOSC X 11_B: LS IntOSC RL (240 kHz)

(2) CKSC_ARTCAS_ACT — C_AWO_RTCA Source Clock Active Register

Access: This register 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 10.26 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 Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.27 CKSC_ARTCAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When writing to these bits, 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

Access: This register 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 10.28 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

Access: This register can be read/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 “1” value after reset of bit 1.

Table 10.29 CKSC_ARTCAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When writing to these bits, write the value after reset.
0	ARTCAD STPMSK	0: Clock domain C_AWO_RTCA is stopped in stand-by mode. 1: Clock domain C_AWO_RTCA is not stopped in stand-by mode.

10.4.3.4 ADCA0 Clock Domain C_AWO_ADCA

(1) CKSC_AADCAS_CTL — C_AWO_ADCA Source Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.30 CKSC_AADCAS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 RH (8 MHz) (default) 10_B: MainOSC X 11_B: PPLLCLK2

Note 1. Before shifting to DEEPSTOP mode, select one other than PPLLCLK2.

(2) CKSC_AADCAS_ACT — C_AWO_ADCA Source Clock Active Register

Access: This register can only 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 10.31 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 Divided Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.32 CKSC_AADCAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 8MHz after division by 2.

(4) CKSC_AADCAD_ACT — C_AWO_ADCA Divided Clock Active Register

Access: This register can only 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 10.33 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]	Current active clock divider for C_AWO_ADCA

(5) CKSC_AADCAD_STPM — C_AWO_ADCA Stop Mask Register

Access: This register can be read/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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AADCADSTPMSK
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 “1” value after reset of bit 1.

Table 10.34 CKSC_AADCAD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	AADCADSTPMSK	0: Clock domain C_AWO_ADCA is stopped in stand-by mode. 1: Clock domain C_AWO_ADCA is not stopped in stand-by mode.

10.4.3.5 FOUT Clock Domain C_AWO_FOUT

(1) CKSC_AFOUTS_CTL — C_AWO_FOUT Source Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.35 CKSC_AFOUTS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, 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 X 010_B: HS IntOSC RH (8 MHz) 011_B: LS IntOSC RL (240 kHz) 100_B: SubOSC XT 101_B: CPLLCLK2 110_B: PPLLCLK4 111_B: Setting prohibited

Note 1. Before shifting to DEEPSTOP mode, select one other than CPLLCLK2 and PPLLCLK4.

(2) CKSC_AFOUTS_ACT — C_AWO_FOUT Source Clock Active Register

Access: This register can only 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 10.36 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

Access: This register can be read/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 “1” value after reset of bit 1.

Table 10.37 CKSC_AFOUTS_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, 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.

10.4.3.6 CPU Clock Domain C_ISO_CPUCLK

(1) CKSC_CPUCLKS_CTL — C_ISO_CPUCLK Source Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.38 CKSC_CPUCLKS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	CPUCLKSCSID [1:0]	Source Clock Setting for C_ISO_CPUCLK 00_B: Setting prohibited 01_B: Emergency clock EMCLK (default) 10_B: MainOSC X 11_B: CPPLLCLK

(2) CKSC_CPUCLKS_ACT — C_ISO_CPUCLK Source Clock Active Register

Access: This register can only 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 10.39 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 Divided Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.40 CKSC_CPUCLKD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, 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 Divided Clock Active Register

Access: This register can only 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 10.41 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]	Current active clock divider for C_ISO_CPUCLK

10.4.3.7 Peripheral Clock Domains C_ISO_PERI1 and C_ISO_PERI2

(1) CKSC_IPERI1S_CTL — C_ISO_PERI1 Source Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.42 CKSC_IPERI1S_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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

Access: This register can only 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 10.43 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

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.44 CKSC_IPERI2S_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing to these bits, 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

Access: This register 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 10.45 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

10.4.3.8 RLIN Clock Domains C_ISO_LIN

(1) CKSC_ILINS_CTL — C_ISO_LIN Source Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.46 CKSC_ILINS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing to these bits, write the value after reset. When writing to these bits, write the value after reset.
1, 0	ILINSCSID[1:0]	Source Clock Setting for C_ISO_LIN 00_B: Disabled 01_B: CPUCLK2 (default) 10_B: MainOSC X 11_B: PPLLCLK2

(2) CKSC_ILINS_ACT — C_ISO_LIN Source Clock Active Register

Access: This register can only 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 10.47 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 Divided Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.48 CKSC_ILIND_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 applied to RLIN30.

(4) CKSC_ILIND_ACT — C_ISO_LIN Divided Clock Active Register

Access: This register can only 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 10.49 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

Access: This register can be read/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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ILINDSTPMSK
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 “1” value after reset of bit 1.

Table 10.50 CKSC_ILIND_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	ILINDSTPMSK	0: Clock domain C_ISO_LIN is stopped in stand-by mode. 1: Clock domain C_ISO_LIN is not stopped in stand-by mode.

10.4.3.9 ADCA1 Clock Domain C_ISO_ADCA

(1) CKSC_IADCAS_CTL — C_ISO_ADCA Source Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.51 CKSC_IADCAS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	IADCASCSID [1:0]	Source Clock Setting for C_ISO_ADCA 00_B: Disabled 01_B: HS IntOSC RH (8 MHz) (default) 10_B: MainOSC X 11_B: PPLLCLK2

CAUTION

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 divided clock 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 divided clock selection register (CKSC_IADCAD_CTL)

(2) CKSC_IADCAS_ACT — C_ISO_ADCA Source Clock Active Register

Access: This register can only 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 10.52 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 Divided Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.53 CKSC_IADCAD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 8MHz after division by 2.

CAUTION

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 divided clock 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 divided clock selection register (CKSC_IADCAD_CTL)

(4) CKSC_IADCAD_ACT — C_ISO_ADCA Divided Clock Active Register

Access: This register can only 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 10.54 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]	Current active clock divider for C_ISO_ADCA

10.4.3.10 RS-CAN Clock Domains C_ISO_CAN and C_ISO_CANOSC

(1) CKSC_ICANS_CTL — C_ISO_CAN Source Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.55 CKSC_ICANS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	ICANS CSID[1:0]	Source Clock Setting for C_ISO_CAN 00_B: Disabled 01_B: MainOSC X 10_B: CPLLCLK 11_B: CPUCLK (default)

(2) CKSC_ICANS_ACT — C_ISO_CAN Source Clock Active Register

Access: This register can only 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 10.56 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]	Current active C_ISO_CAN source clock

(3) CKSC_ICANS_STPM — C_ISO_CAN Stop Mask Register

Access: This register can be read/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 “1” value after reset of bit 1.

Table 10.57 CKSC_ICANS_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, 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 Divided Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.58 CKSC_ICANOSCD_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	ICANOSCD CSID[1:0]]	Clock Divider Setting for C_ISO_CANOSC 00_B: Disabled (default) 01_B: MainOSC X /1 10_B: MainOSC X /2 11_B: Setting prohibited

(5) CKSC_ICANOSCD_ACT — C_ISO_CANOSC Divided Clock Active Register

Access: This register can only 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 10.59 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]	Current active clock divider for C_ISO_CANOSC

(6) CKSC_ICANOSCD_STPM — C_ISO_CANOSC Stop Mask Register

Access: This register can be read/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 “1” value after reset of bit 1.

Table 10.60 CKSC_ICANOSCD_STPM Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, 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.

10.4.3.11 CSI Clock Domain C_ISO_CSI

(1) CKSC_ICSIS_CTL — C_ISO_CSI Source Clock Selection Register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 10.61 CKSC_ICSIS_CTL Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing to these bits, write the value after reset.
1, 0	ICSISCSID[1:0]	Source Clock Setting for C_ISO_CSI 00_B: Disabled 01_B: CPUCLK (default) 10_B: PPLLCLK 11_B: Setting prohibited

(2) CKSC_ICSIS_ACT — C_ISO_CSI Source Clock Active Register

Access: This register 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 10.62 CKSC_ICSIS_ACT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1, 0	ICSISCSID[1:0]	Source clock for currently active C_ISO_CSI

10.5 Clock Domain Setting Method

10.5.1 Clock Domain Setting

10.5.1.1 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 one out of a selection of source clock, to be used for the domain clock.
  - AWO source clock selection: CKSC_A<name>S_CTL
  - ISO source clock selection: CKSC_I<name>S_CTL
- Divided clock selection registers
  These registers set up a divisor of clock divider for the selected source clock.
  - AWO clock divider: CKSC_A<name>D_CTL
  - ISO clock divider: CKSC_I<name>D_CTL
- Source clock active registers and divided clock active registers
  These registers return the currently active source clock selection and divisor value, respectively.
  - AWO source clock active register/divided clock active register: CKSC_A<name>S_ACT/ CKSC_A<name>D_ACT
  - ISO source clock active register/divided clock active register: CKSC_I<name>S_ACT/ CKSC_I<name>D_ACT

NOTE

- Not all clock selectors provide all above described controls.
- The symbol “I”, which indicates the power domain, is not added to the names of registers within clock domain C_ISO_CPUCLK.

10.5.1.2 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. Setting a divider clock
   - Select a divider clock. (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. After CKSC_A<name>S_ACT and CKSC_I<name>S_ACT are updated with the new values which have been written to CKSC_A<name>S_ACT and CKSC_I<name>S_ACT, continue processing.

Note 2. After CKSC_A<name>D_ACT and CKSC_I<name>D_ACT are updated with the new values which have been written to CKSC_A<name>D_ACT and CKSC_I<name>D_ACT, continue processing.

CAUTION

The source clock to be selected must be operated before the above settings.

The behavior and performance are not guaranteed when setup is made while the source clock is stopped.

10.5.2 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 in stand-by.
  If the domain clock was in operation before stand-by, it is automatically re-started after wake-up from stand-by.
- 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 in stand-by.

10.5.3 Clock Domain Settings

The following table shows a selectable source clock, a frequency division ratio, and a register to be used for each clock domain.

Table 10.63 List of Selectable Clocks (1/2)

Clock Domain	Domain Clock	Selectable Register		Frequency Divided Register		Applicable 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			
			CPLLCLK2			
			PPLLCLK4			
			Disable			
C_ISO_CPUCLK	CKSCLK_ICPUCLK	CKSC_CPUCLKS_CTL	MainOSC	CKSC_CPUCLKD_CTL	**1/1**	CPU subsystem (MEMC)*1
			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*2	**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 10.63 List of Selectable Clocks (2/2)

Clock Domain	Domain Clock	Selectable Register		Frequency Divided Register		Applicable Unit
C_ISO_CAN	CKSCLK_ICAN	CKSC_ICANS_CTL	MainOSC	—	**1/1**	RS-CANn
			CPLLCLK			
			CPUCLK			
			Disable			
C_ISO_CANOSC	CKSCLK_ICANOSC	—	MainOSC	CKSC_ICANOSCD_CTL	**1/1**	RS-CANn
					1/2	
					Disable	
C_ISO_CSI	CKSCLK_ICSI	CKSC_ICSIS_CTL	PPLLCLK	—	**1/1**	CSIGn CSIHn
			CPUCLK			
			Disable			

Note: The items written in **bold** are the initial setting clocks for each register.

Note 1. The MEMC clock is set based on C_ISO_CPUCLK using the option bytes.

Note 2. This setting is applied to only RLIN30.

CAUTION

To stop the clock source selected for the clock domain before shifting to STOP/ DEEPSTOP mode, select “Disable” for that clock domain in advance. In the case of the source clock of a clock domain for which “Disable” cannot be selected, do not stop the clock source during function operation in that clock domain. “Disable” need not be selected in order to stop a clock source by shifting to STOP/DEEPSTOP mode.

10.6 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.

10.6.1 Functional Overview

Figure 10.9 shows the configuration of the frequency output function.

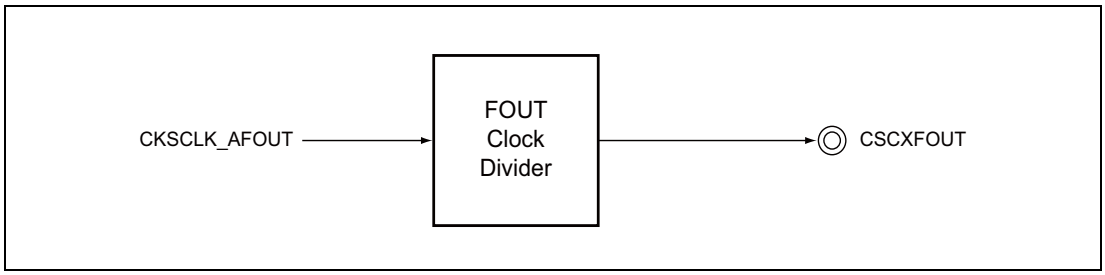


Figure 10.9 Frequency Output Function

The clock output function outputs the CKSCLK_AFOUT clock divided by 1 to 63 through the clock division circuit 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.

$f_{CSCXFOUT}$ = (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.

10.6.2 Register

10.6.2.1 List of Registers

The FOUT registers are listed in the following table.

Table 10.64 Register of the CSCXFOUT Clock Divider

Register Name	Symbol	Address
Clock divider register	FOUTDIV	FFF8 2800_H
Clock divider status register	FOUTSTAT	FFF8 2804_H

10.6.2.2 FOUTDIV — Clock Division Ratio Register

This register defines the clock divisor.

Access: This register can be read/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 10.65 FOUTDIV Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5 to 0	FOUTDIV[5:0]	Clock Divisor N 00_H: Clock output is stopped*1 01_H: N = 1 02_H: N = 2 ... $3E_H$: N = 62 $3F_H$: N = 63

Note 1. It is possible to set up only when output of CKSCLK_AFOUT is suspended (CKSC_AFOUTS_CTL = 000_B).

10.6.2.3 FOUTSTAT — Clock Divider Status Register

This register indicates the clock output status.

Access: This register can only 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 10.66 FOUTSTAT Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	FOUTCLKACT	Divider Clock Active 0: Frequency output is stopped. 1: Frequency output is ongoing.
0	FOUTSYNC	Divider Clock Synchronized 0: The divider clock is in the process of changing the settings. 1: The divider clock is stable (the “stopped” state is included).

10.7 Clock Monitor A (CLMA)

This section describes clock monitor A (CLMA).

The first section describes the attributes specific to the RH850/F1L microcontrollers, including the number of channels, register bases, addresses, and input/output signal names. The ensuing sections describe the functions relevant to all the operations.

10.7.1 Features of RH850/F1L CLMA

10.7.1.1 Number of Channels

The RH850/F1L microcontrollers respectively incorporate a CLMA with the following number of channels.

Table 10.67 Number of Channels

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of channels	3					
Name	CLMA0, CLMA1, CLMA2					

10.7.1.2 Register Base Addresses

The CLMA base addresses are listed in the following table.

The CLMA register addresses are expressed as an offset of the base address.

Table 10.68 Register Base Addresses

Base Address Name	Base Address
<CLMA0_base>	FFF8 $C000_H$
<CLMA1_base>	FFF8 $D000_H$
<CLMA2_base>	FFF8 $E000_H$

10.7.1.3 Clock Supply to CLMA

The clocks monitored by CLMA and the CLMA sampling clocks are indicated below.

Table 10.69 CLMA Monitored Clocks

Channel Name	Clock for the Unit	Clock Name
CLMA0	CLMATMON (monitored clock)	HS IntOSC
	CLMATSMP (sampling clock)	LS IntOSC
CLMA1	CLMATMON (monitored clock)	MainOSC
	CLMATSMP (sampling clock)	LS IntOSC
CLMA2	CLMATMON (monitored clock)	CPLLCLK
	CLMATSMP (sampling clock)	HS IntOSC

10.7.1.4 Internal Input/Output Signals

The internal input/output signals of CLMA are listed in the following table.

Table 10.70 Internal Output Signals

Unit Signal Name	Description	Connection
$\overline{\text{CLMATRES}}$	CLMA0 error reset	Reset controller ($\overline{\text{CLMA0RES}}$)
$\overline{\text{CLMATRES}}$	CLMA1 error reset	Reset controller ($\overline{\text{CLMA1RES}}$)
$\overline{\text{CLMATRES}}$	CLMA2 error reset	Reset controller ($\overline{\text{CLMA2RES}}$)

10.7.2 Overview

10.7.2.1 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 10.10**.

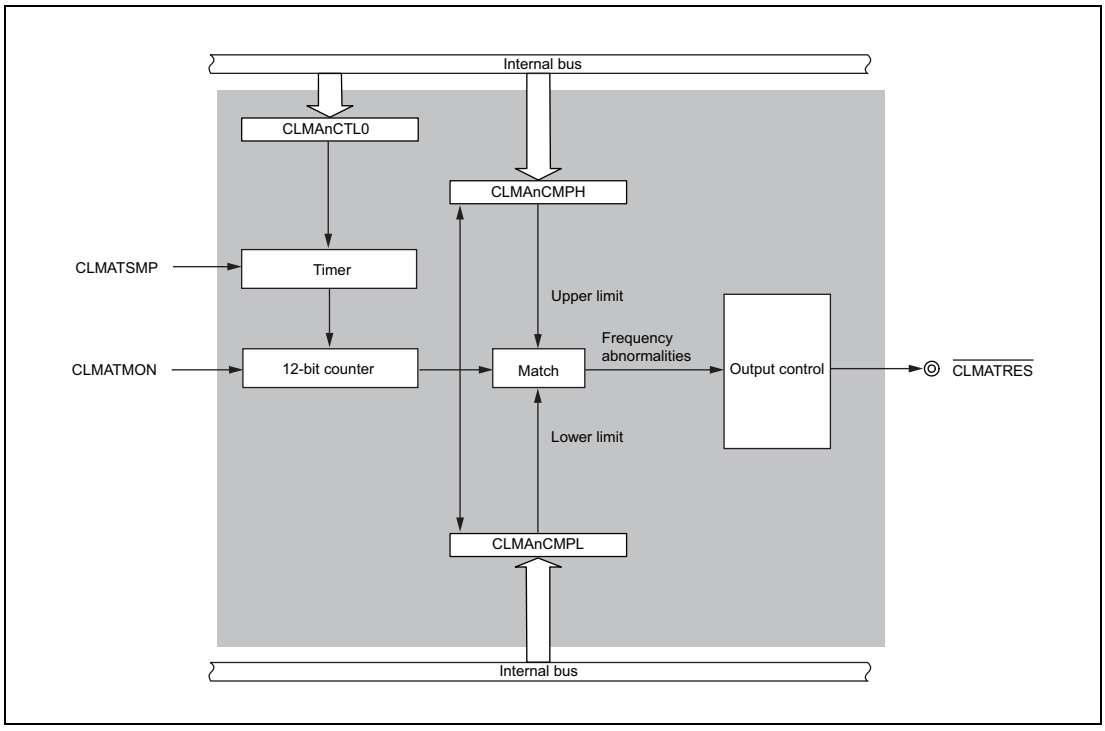


Figure 10.10 Block Diagram of the Clock Monitor A

NOTE

Once enabled, only a reset can disable the CLMAn.

10.7.3 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.

10.7.4 Functions

10.7.4.1 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 stops or its frequency 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].

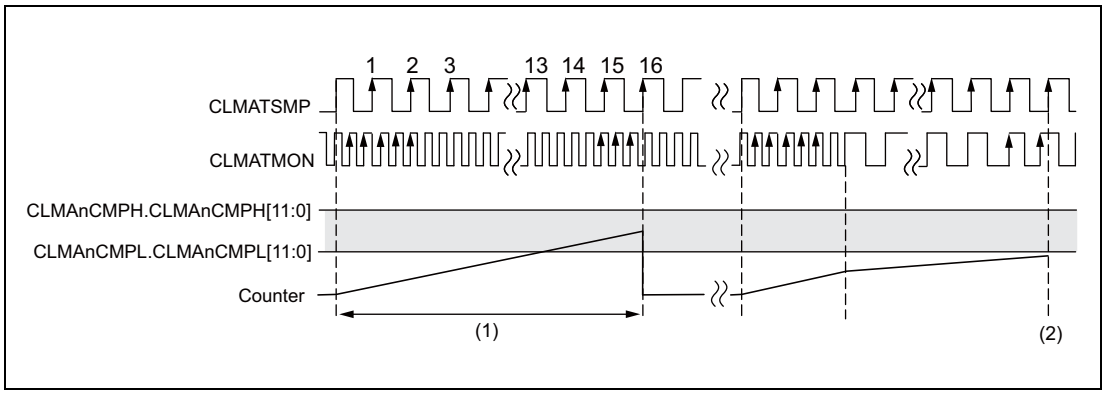


Figure 10.11 Example: $f_{CLMATMON}$ is Lower than the Specified Limit

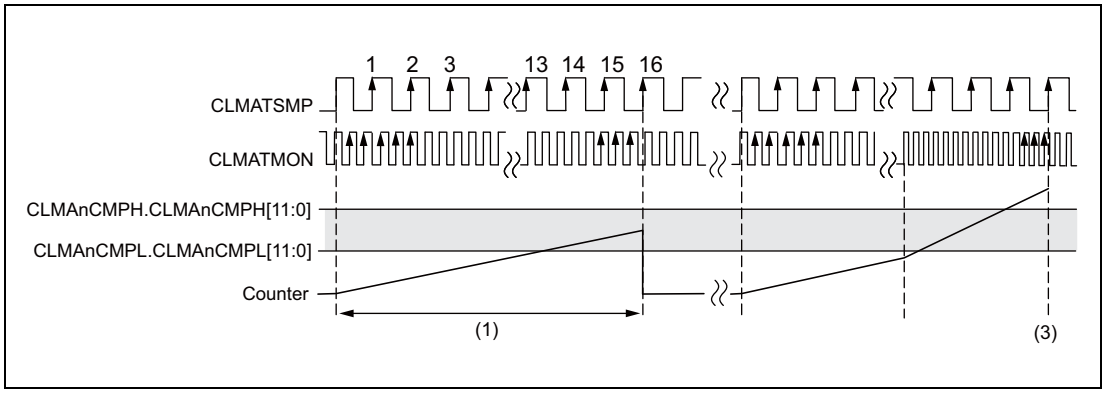


Figure 10.12 Example: $f_{CLMATMON}$ is Higher than the Specified Limit

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:

$$\text{Lower threshold} = N_{min} = \frac{f_{CLMATMON(dmin)}}{f_{CLMATSMP(max)}} \times 16 - 1$$

$$\text{Upper threshold} = N_{max} = \frac{f_{CLMATMON(max)}}{f_{CLMATSMP(min)}} \times 16 + 1$$

Example:

When $f_{CLMATSMP}$ = 240 kHz (±8%) and $f_{CLMATMON}$ = 16 MHz (±5%), the recommended threshold values are the followings:

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 reset values 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] = $3FF_H$

10.7.4.2 Notification of Abnormal Clock Frequency

If $f_{CLMATMON}$ exceeds the upper threshold or fall 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.

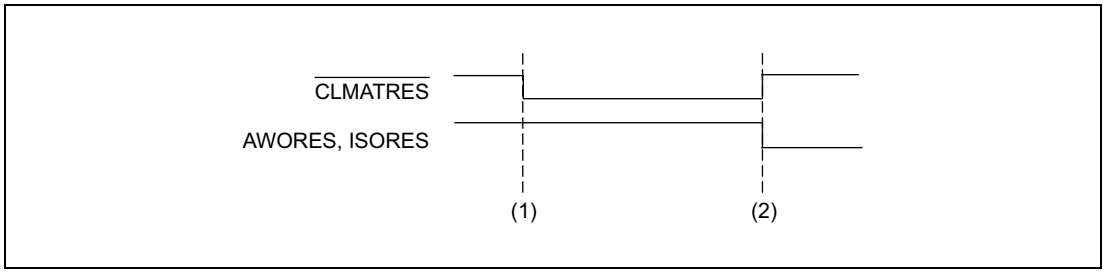


Figure 10.13 Error Request Signal Output if $f_{CLMATMON}$ Exceeds Upper Threshold

CAUTION

For usage notes for CLMAn abnormality detection, see Section 10.7.6, Usage Notes for CLMAn.

10.7.4.3 CLMAn Enable (Write to CLMAnCTL0)

The control register CLMAnCTL0 is a write protection register to enable CLMAn.

NOTE

CLMAn can be only disabled by reset. Writing to CLMAnCTL0 cannot disable CLMAn.

(1) Value after reset of CLMAnCTL0

The value after reset of CLMAnCTL0 is 00_H.

(2) Procedures to enable CLMAn

Set 01_H in CLMAnCTL0 under the following instruction sequence.

1. Write $A5_H$ in CLMAnPCMD.
2. Writing CLMAnCTL0 should be proceeded under the following sequence to enable CLMAn:
   - Write 01_H first.
   - Write the reversed value FE_H.
   - Write the target value 01_H again.
3. Read out the value of CLMAnCTL0.
   If the value of CLMAnCTL0 is 01_H, CLMAn has been enabled.
   In other cases as shown below, the value of the CLMAnCTL0 write operation status register, CLMAnPS should be checked.
   - When CLMAnPS = 01_H, the instruction sequence does not proceed correctly. Execute the sequence from step 1 again to enable CLMAn.

10.7.5 Registers

10.7.5.1 List of Registers

The following table lists the CLMA registers.

<CLMAn_base> is defined in **Section 10.7.1.2, Register Base Addresses**.

Table 10.71 List of Clock Monitor Registers

Module Name	Register Name	Abbreviation	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

10.7.5.2 CLMAnCTL0 — CLMAn Control Register 0

This register enables the clock monitor CLMAn. Writing to this register is protected by a special sequence of instructions. For details, see **Section 10.7.4.3, CLMAn Enable (Write to CLMAnCTL0)**.

Access: This register can be read/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 10.72 CLMAnCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset. When writing to these bits, write the value after reset.
0	CLMAnCLME	Enables or disables the clock monitor. 0: CLMAn is disabled. 1: CLMAn is enabled.

CAUTION

The only condition for clearing a CLMAnCTL0.CLMAnCLME bit is a reset (AWORES, ISORES).

10.7.5.3 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 **(1) Calculation method of the thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]**.

Access: This register can be read/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 10.73 CLMAnCMPH Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When writing to these bits, write the value after reset. When writing to these bits, 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.

10.7.5.4 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 **(1) Calculation method of the thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]**.

Access: This register can be read/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 10.74 CLMAnCMPL Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When writing to these bits, write the value after reset. When writing to these bits, 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.

10.7.5.5 CLMATEST — CLMA Test Register

This register is used to test CLMA0/CLMA1/CLMA2.

Writing to this register is protected by the special sequence of instructions.

For details, see **Section 4, Write-Protected Registers**.

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

Address: FFF8 C100$_H$

Value after reset: 0000 0000$_H$ This register is initialized by a power-up reset PURES (power-on-clear or debugger reset).

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 10.75 CLMATEST Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When writing to these bits, write the value after reset. When writing to these bits, 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 1: Reset released 0: Reset executed

Note 1. This bit is enabled when the CLMAnTESEN bit is 1.

10.7.5.6 CLMATESTS — CLMA Test Status Register

This register is used to test CLMA0/CLMA1/CLMA2.

Access: This register can only be read 32-bit units.

Address: FFF8 C104$_H$

Value after reset: 0000 0000$_H$ This register is initialized by a power-up reset PURES (power-on-clear or debugger reset).

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 10.76 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

10.7.5.7 CLMAnEMU0 — CLMAn Emulation Register 0

This register is used to set up pseudo flags at the time of emulation.

This register can be accessed only during break by debugger and be reset by break release.

Access: This register can be read/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 10.77 CLMAnEMU0 Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing to these bits, write the value after reset.
1	CLMAnSLFST	Specifies the higher value of $f_{CLMATMON}$. 0: CLMATMON is assumed to be within the normal frequency range 1: CLMATMON is assumed to exceed the upper threshold.
0	CLMAnSLSLW	Specifies the lower value of $f_{CLMATMON}$. 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 too low and too high CLMATMON at the same time. Thus CLMAnEMU0 must not be set to 03_H.

10.7.6 Usage Notes for CLMAn

Do not use a clock for which the CLMAn bit indicates an abnormality. The behavior and performance are not guaranteed if such a clock is used. When CLAM0 detects clock abnormality, modifying the clock domain settings is prohibited.

Table 10.78 provides usage notes for each CLAMn.

Table 10.78 Usage Notes for CLMAn

Monitor Clock	CPU Clock after CLMA Reset Release	Note
HS IntOSC (CLMA0)	EMCLK*1	Set ROSCE.ROSCDISTRG to 1.*1 After occurrence of the CLMA0RES, modifying the clock domain settings using HS intOSC is prohibited.
MainOSC (CLMA1)	EMCLK (= HS IntOSC)	Do not set control registers of MainOSC and PLL. After occurrence of the CLMA1RES, modifying the clock domain settings using MainOSC and PLL is prohibited.
PLL (CLMA2)	EMCLK (= HS IntOSC)	Do not set control registers of PLL. After occurrence of the CLMA2RES, modifying the clock domain settings using PLL is prohibited.

Note 1. The state of the CPU clock varies with the state of HS IntOSC operation.
When HS IntOSC is stopped, EMCLK becomes LS IntOSC.
When HS IntOSC oscillates at a frequency above the range for CLMA0, EMCLK becomes HS IntOSC.
When ROSCE.ROSCDISTRG is set to “1”, EMCLK changes from HS IntOSC to LS IntOSC.

Section 11 Stand-By Controller (STBC)

This section describes the functions of the stand-by controller (STBC), the registers, and various stand-by modes.

11.1 Functions

11.1.1 Types of Stand-By Mode

This RH850/F1L supports the following five stand-by modes:

- HALT mode
  HALT mode can be entered from normal RUN mode by performing the CPU instruction “HALT”. This stops the CPU operation, while all clocks other than the CPU clock continue to operate and all areas remain under power.
  For details, see *RH850 Family Users Manual: Software*.
- STOP mode
  In STOP mode, certain clock supplies to a 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 10, 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 CPU, peripherals on Always-ON area, and RLIN3 can operate.
  The CPU 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 CPU 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 CPU is running is called RUN mode.

11.1.2 Wake-Up Control

11.1.2.1 Wake-Up Factors from Stand-By Mode

The stand-by controller can initiate return from stand-by mode by the following wake-up factors.

Table 11.1 Wake-Up Factors 1

Wake-up Factors	Unit	Stop →Run	DEEPSTOP →Run	Cyclic RUN →Run*1	Cyclic STOP →Run*1
TNMI	Port	√	√	√	√
WDTA0NMI	WDTA0	√	√	√	√
INTLVIL	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	√	—	—	—
INTRCANGRECC*2	RSCAN0	√	—	—	—
INTRCAN0REC*2	RSCAN0	√	—	—	—
INTRCAN1REC*2	RSCAN0	√	—	—	—
INTRCAN2REC*2	RSCAN0	√	—	—	—
INTRCAN3REC*2	RSCAN0	√	—	—	—
INTRCAN4REC*2	RSCAN0	√	—	—	—
INTRCAN5REC*2	RSCAN0	√	—	—	—

Note 1. When returning to RUN from Cyclic RUN and Cyclic STOP, it is via DEEPSTOP.

Note 2. By using the INTP external interrupt assigned to the alternate-function pin shared with the CAN reception pin, wake-up from DEEPSTOP is possible.

Table 11.2 Wake-Up Factors 2

Wake-Up Factors	Unit	DEEPSTOP → Cyclic RUN	Cyclic STOP → Cyclic RUN
INTADCA0I0	ADCA0	—	√
INTADCA0I1	ADCA0	—	√
INTADCA0I2	ADCA0	—	√
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	—	√

CAUTION

For the pins of the function used for the wake-up factors from DEEPSTOP, use the multiplexed functions of the ports assigned to the AWO area.

11.1.2.2 Setting of Wake-Up Factors

Wake-up factors for returning from stand-by mode are controlled by the following stand-by controller registers:

- Wake-up factor registers: WUF0, WUF20, WUF_ISO0

  Upon occurrence of an unmasked wake-up event, the associated wake-up factor flag is set to 1. By use of this register the application program can identify the wake-up factor.

- Wake-up mask registers: WUFMSK0, WUFMSK20, WUFMSK_ISO0

  Each bit of these registers is assigned to a certain wake-up event. Wake-up by this event is enabled if its mask bit is set to 0. Wake-up factor assigned to Wake-up factor 1 and 2 should not to be enabled at the same time.

- Wake-up factor clear registers: WUFC0, WUFC20, WUFC_ISO0

  In order to clear a Wake-up factor flag of a Wake-up factor register (WUF0, WUF20, WUF_ISO0), its assigned bit has to be set to 1.

NOTE

The Wake-up factor flags in the Wake-up factors registers (WUF0, WUF20, WUF_ISO0) indicate only the occurrence of a Wake-up factor. Thus an asserted Wake-up factor flag does not mean that the transition from stand-by to normal operation mode is already accomplished.

The assignment of the wake-up events to the wake-up control and status register bits is given in the table below.

For details about the wake-up control and status registers, see **Section 11.2.2.3, WUF0/WUF20/WUF_ISO0 — Wake-Up Factor Registers**, **Section 11.2.2.4, WUFMSK0/WUFMSK20/WUFMSK_ISO0 — Wake-Up Factor Mask Registers**, and **Section 11.2.2.5, WUFC0/WUFC20/WUFC_ISO0 — Wake-Up Factor Clear Registers**.

Table 11.3 Wake-Up Factors 1 Registers Assignments

Name	Assignment of WUFm/WUFMSKm/ WUFCm Register Bits			Function	48 pins	64 pins	80 pins	100 pins	144 pins	176 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	√	√	√	√	√	√
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	√	√	√	√	√	√
INTRCANGRECC	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	—	—	—	*1	√	√
INTRCAN4REC	WUF_ISO0 [7]	WUFMSK_ISO0[7]	WUFC_ISO0[7]	RSCAN0	—	—	—	*1	*2	√
INTRCAN5REC	WUF_ISO0 [8]	WUFMSK_ISO0[8]	WUFC_ISO0[8]	RSCAN0	—	—	—	*1	*2	√

Note 1. Supported only for F1L for Gateway

Note 2. Supported only for Code Flash 1.5 M and 2 M products

Table 11.4 Wake-Up Factors 2 Registers Assignments

Name	Assignment of WUFm/ WUFMSKm/WUFCm Register Bits			Function	48 pins	64 pins	80 pins	100 pins	144 pins	176 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	—	—	—	—	√	√

11.1.3 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 any stand-by mode is 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 the microcontroller from stand-by mode by a manual stop via the debugger.

Thus it is recommended to enable the OCD wake-up for terminating all stand-by modes by setting WUFMSK0[31] = 0.

When the hot plug-in function is used, make sure to enable these factors. The INTDCUTDI interrupt is required to return from stand-by mode as the wake-up handling.

11.1.4 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.
For details on the port groups in the isolated area, see **Section 38, Power Supply and Power Domains**.

11.1.4.1 I/O Buffer Hold State

If an I/O buffer is set into I/O buffer hold state, the state of the buffer is frozen. Thus its input and/or output remains in the state before entering I/O buffer hold state. No external or internal signal can change its state until the I/O buffer hold state is terminated.

11.1.4.2 I/O Buffers during STOP Mode

The I/O buffers of areas in STOP mode (clock has been stopped) remain in the state before entering STOP mode (I/O buffer hold state is not entered).

11.1.4.3 I/O Buffers during DEEPSTOP Mode

The I/O buffers in DEEPSTOP mode transition to I/O buffer hold state by default.

After wake-up from DEEPSTOP, cancel I/O buffer hold state in the steps below.

1. Re-configure the peripheral or port function.
2. IOHOLD.IOHOLD = 0

The statuses of the I/O buffer during standby mode and after wakeup are shown below.

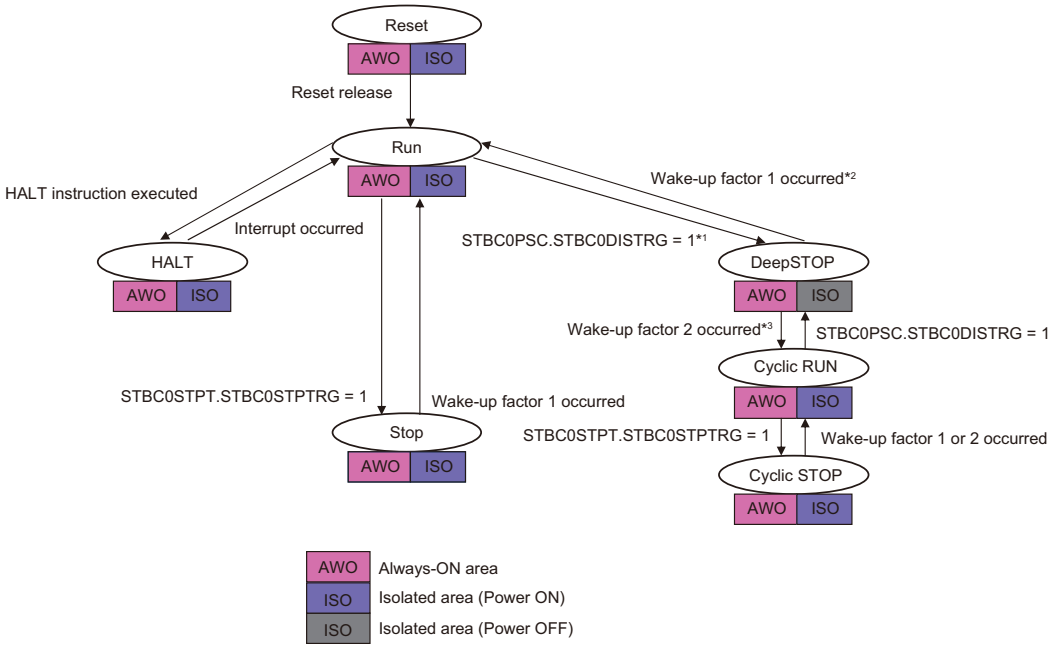
Table 11.5 Buffer Operation during Standby Mode and after WakeUp

	Before Standby	During Standby	After Wakeup
STOP mode	I/O Buffer during normal operation		
DEEPSTOP mode	I/O buffer in 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.

11.1.5 Transition to Stand-By Mode

The figure below shows transition of RUN mode and stand-by mode.



Note 1. Set one other than PLL clock as the CPU clock source.

Note 2. To return from DEEPSTOP to RUN, ISO peripheral function registers must be set anew. The CPU starts the operation from the reset address on the code flash.

Note 3. The CPU starts the operation from the reset address on the retention RAM.

Figure 11.1 Transition of Stand-By Mode

11.2 Registers

11.2.1 List of Registers

The following table lists the stand-by controller registers.

Table 11.6 List of Registers

Module Name	Register Name	Symbol	Address
—	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 registers	IOHOLD	FFF8 $0B00_H$

11.2.2 Details of Stand-By Controller Control Registers

11.2.2.1 STBC0PSC — Power Save Control Register

The requirement for a specific instruction sequence in which the protection command register PROTCMD0 is used protects this register against writing.

Access: This register can be read/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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STBC0 DISTR G	—
Value after reset	0	0	0	0	0	0	0	0	0	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 11.7 STBC0PSC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When writing to these bits, write the value after reset.
1	STBC0DISTRG	0: Writing “0” has no effect. 1: DeepStop mode is entered
0	Reserved	When writing to this bit, write the value after reset.

11.2.2.2 STBC0STPT — Power Stop Trigger Register

The requirement for a specific instruction sequence in which the protection command register PROTCMD0 is used protects this register against writing.

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

Address: FFF8 0110H

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 11.8 STBC0STPT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	STBC0STPTRG	0: Writing “0” has no effect. 1: Stop mode is entered – If in RUN mode: Transition to STOP mode – If in Cyclic RUN mode: Transition to Cyclic STOP mode

11.2.2.3 WUF0/WUF20/WUF_ISO0 — Wake-Up Factor Registers

This register informs of a wake-up event.

Access: This register can only 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 11.9 WUF0/WUF20/WUF_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFy	Indicates the generation of a wake-up event. 0: Wake-up event is not generated 1: Wake-up event 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 event.

Wake-up factor

As for the assignment of a wake-up event to the wake-up factor register bit, see **Table 11.3, Wake-Up Factors 1 Registers Assignments** and **Table 11.4, Wake-Up Factors 2 Registers Assignments**.

The bit to which a wake-up event is not assigned is read as the value “0”.

11.2.2.4 WUFMSK0/WUFMSK20/WUFMSK_ISO0 — Wake-Up Factor Mask Registers

This register enables a wake-up event.

Access: This register can be read/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 11.10 WUFMSK0/WUFMSK20/WUFMSK_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFMSKy	Enables/disables a wake-up event. 0: Wake-up event is enabled 1: Wake-up event 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 event.

Wake-up factor

As for the assignment of a wake-up event to the wake-up factor register bit, see **Table 11.3, Wake-Up Factors 1 Registers Assignments** and **Table 11.4, Wake-Up Factors 2 Registers Assignments**.

Before writing to this register, write the value “1” to the bit to which a wake-up event is not assigned.

11.2.2.5 WUFC0/WUFC20/WUFC_ISO0 — Wake-Up Factor Clear Registers

This register clears the WUFy bit in the wake-up factor register.

Access: This register can only 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
	WUFC31	WUFC30	WUFC29	WUFC28	WUFC27	WUFC26	WUFC25	WUFC24	WUFC23	WUFC22	WUFC21	WUFC20	WUFC19	WUFC18	WUFC17	WUFC16
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
	WUFC15	WUFC14	WUFC13	WUFC12	WUFC11	WUFC10	WUFC09	WUFC08	WUFC07	WUFC06	WUFC05	WUFC04	WUFC03	WUFC02	WUFC01	WUFC00
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 11.11 WUFC0/WUFC20/WUFC_ISO0 Register Contents

Bit Position	Bit Name	Function
31 to 0	WUFCy	Clear the wake-up factor WUFy in the wake-up factor register. 0: WUFy is not modified 1: WUFy is cleared

Wake-up factor

As for the assignment of a wake-up event to the wake-up factor register bit, see **Table 11.3, Wake-Up Factors 1 Registers Assignments** and **Table 11.4, Wake-Up Factors 2 Registers Assignments**.

Before writing to this register, write the value “0” to the bit to which a wake-up event is not assigned.

11.2.2.6 IOHOLD — I/O Buffer Hold Control Register

This register specifies the hold state of the I/O buffer in DEEPSTOP mode. The requirement for a specific instruction sequence in which the protection command register PROTCMD0 is used protects this register against writing.

Access: This register can be read/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 11.12 IOHOLD Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, 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.

11.3 Mode

This section explains the transition procedure of stand-by mode for respective modes.

11.3.1 STOP Mode

In STOP mode, the clock supply to the Always-On area and isolated area is stopped. In addition, stop all of the peripheral functions to which the clock supply is to be stopped before the transition to STOP mode is made.

The transition procedure (example) to STOP mode is shown below.

Preparation for stand-by

- Stop all of the peripheral functions that stop the clock supply.
- Disable the interrupt handling by the CPU instruction “DI”.
- Set the interrupt control registers.
  - Clear the interrupt flag (ICxxx.RFxxx = 0).
  - Mask the interrupt of non-wake-up factor (ICxxx.MKxxx = 1).
  - Release to mask the interrupt of Wake-up factor (ICxxx.MKxxx = 0).
- Set the wake-up related registers.
  - Clear the wake-up factor flag (the WUF0/WUFC20/WUFC_ISO0 registers).
  - Mask the non-wake-up factor (the WUFMSK0/WUFMSK20/WUFMSK_ISO0 registers).
  - Release to mask the wake-up factor (the WUFMSK0/WUFMSK20/WUFMSK_ISO0 registers).

CAUTION

When a wake-up factor is assigned to both wake-up factors 1 registers and wake-up factors 2 registers, it can be used only in one of them.

- Set the clock stop mask and select the clock domains to be stopped and to continue operating. (Set by the CKSC_xxx_STPM.xxxxSTPMSK bit.)
- Designate each clock source for oscillation or for stopping. In addition, set the clock mask and select which clock source is to stop and which clock source is to continue operation. (Set by the MOSCSTPMSK bit in the MOSCSTPM register, and the ROSCSTPMSK bit in the ROSCSTPM register.)

Start of stand-by

Set the STBC0STPTRG bit in the STBC0STPT register to 1 to shift to STOP mode.

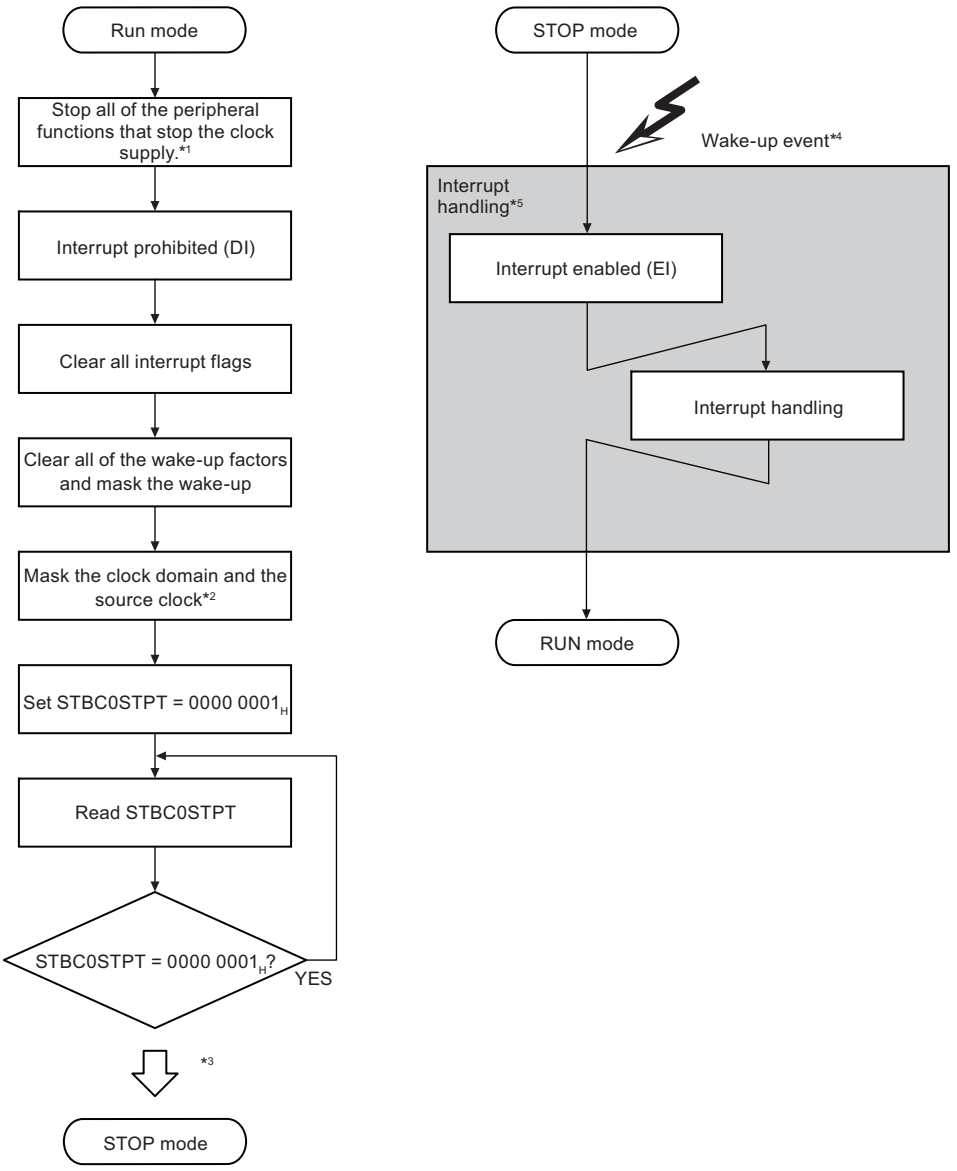
End of stand-by

When a wake-up event is generated, the microcontroller returns from STOP mode.

Wake-up handling

The wake-up factor is determined by the wake-up factor flag (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 that stop the clock supply.

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 shifts 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 event. The generated wake-up event can be checked by the WUF0 and WUF_ISO0 registers.

Note 5. This processing is optional. It is required to execute the interrupt handling after the wake-up.

Figure 11.2 Example of STOP Mode Transition

11.3.2 DEEPSTOP Mode

In DEEPSTOP mode, the clock supply to all areas and the power supply to the isolated area are stopped. However, supply of the clock signal to peripheral modules in the AWO (always-on) area can be made to continue 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 is made.

The transition procedure (example) to DEEPSTOP mode is shown below.

Preparation for stand-by

See the description on **“Preparation for stand-by”** in **Section 11.3.1, STOP Mode**.

Start of stand-by

Set the STBC0DISTRG bit in the STBC0PSC register to 1 to shift to DEEPSTOP mode.

End of stand-by

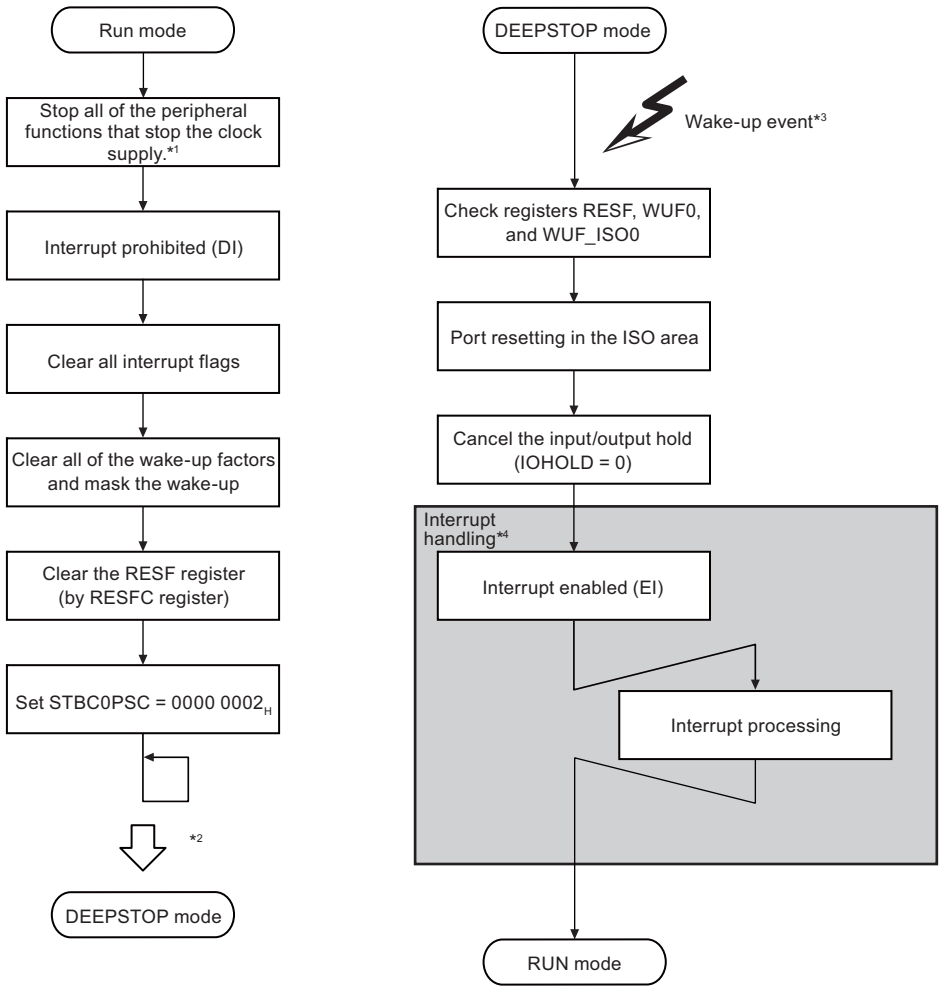
When a wake-up event is generated, the microcontroller returns from DEEPSTOP mode.

Wake-up handling

- When returned from DEEPSTOP mode due to wake-up factor 1, the microcontroller starts the operation from the reset address.
- The wake-up factor is determined by the wake-up factor flag (WUF0).
- The ports in the isolated area maintain the I/O buffer hold state.
  Release the I/O buffer hold state in the following order.

  1. Re-configure the peripheral functions and port functions.

  2. 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 is to 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 that stop the clock supply.

Note 2. After setting STBC0PSC = 0000 0002_H, wait for the transition to the DEEPSTOP mode by the unconditional loop.

Note 3. The CPU starts the program from the reset vector after the generation of a wake-up event. The return from the DEEPSTOP mode by a reset can be checked by the RESF register. In addition, the generated wake-up event can be checked by the WUF0 and WUF_ISO0 registers.

Note 4. This processing is optional. It is required to execute the interrupt handling after the wake-up.

Figure 11.3 Example of DEEPSTOP Mode Transition

11.3.3 Cyclic RUN Mode

In Cyclic RUN mode, the functions except the CPU, AWO area peripheral function, and RLIN3 are stopped. The transition procedure (example) to Cyclic RUN mode is shown below.

Preparation of Cyclic RUN

Arrange the program for Cyclic RUN in the Retention RAM.

At that time, the instruction to shift to DEEPSTOP mode should be arranged in the interrupt vector (exception vector) to be used as the source to return to RUN mode.

For details on the exception vector, see the *RH850 Family User's Manual: Software*.

CAUTION

Do not change the PSW.EBV bit during Cyclic RUN mode.

- Set the wake-up related registers.
  - Clear the wake-up factor flag (the WUFC20 register).
  - Mask the non-wake-up factor (the WUFMSK20 register).
  - Release to mask the wake-up factor (the WUFMSK20 register).
- The transition to DEEPSTOP mode is made. For details on how to transit to DEEPSTOP mode, see **Section 11.3.2, DEEPSTOP Mode**.

Start of Cyclic RUN

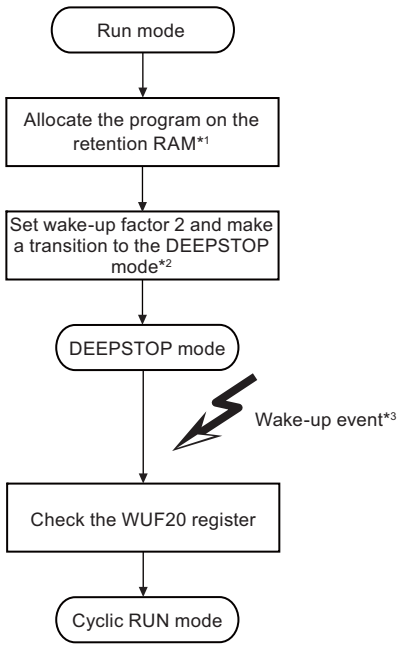
The operation shifts to Cyclic RUN mode at the generation of wake-up factor 2.

End of Cyclic RUN

The Cyclic RUN mode ends at the shift to the Cyclic STOP mode by setting the STBC0STPT.STBC0STPTRG bit to 1, or the shift to the DEEPSTOP mode by setting the STBC0PSC.STBC0DISTRG bit to 1.

Wake-up handling

The wake-up factor is determined by the wake-up factor flag (WUF20).



Note 1. When the mode shifts from the Cyclic RUN mode to the RUN mode by the wake-up event, the transition to the DEEPSTOP mode should be made in the processing of the interrupt vector for the wake-up event. In that case, allocate the interrupt processing program on the retention RAM.

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 by the WUFMSK20 register. Other part of procedure for the transition to the DEEPSTOP mode is as usual.

Note 3. The CPU starts the program from the top address (FEE0 0000_H) on the retention RAM after the generation of a wake-up event. The generated wake-up event can be checked by the WUF20 register.

Figure 11.4 Example of Cyclic RUN Mode Transition

11.3.4 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

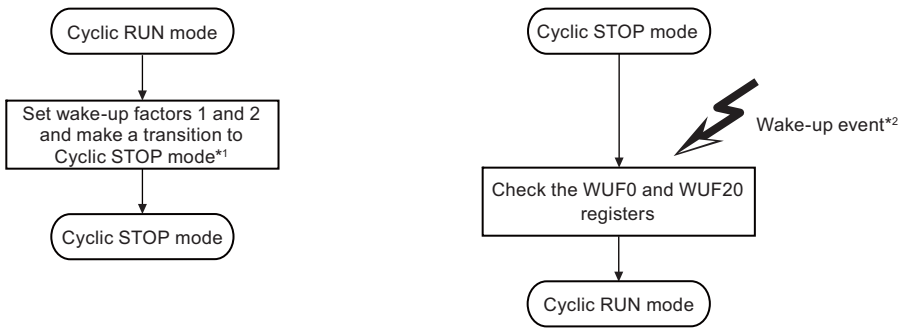
- The transition to Cyclic RUN is made.
- Set the wake-up related registers.
  - Clear the wake-up factor flag (the WUF0/WUFC20 register).
  - Mask the non-wake-up factor (the WUFMSK0/WUFMSK20 register).
  - Release to mask the wake-up factor (the WUFMSK0/WUFMSK20 register).

Start of Cyclic STOP

Set the STBC0STPT.STBC0STPTR bit to 1 to shift to Cyclic STOP mode.

End of Cyclic STOP

The operation shifts to Cyclic RUN mode at the generation of wake-up factor 1 or 2.



Note 1. The wake-up factors 1 and 2 are set to make a transition to RUN mode and Cyclic RUN mode, respectively. When the mode shifts 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 event is generated in Cyclic STOP mode, the mode shifts 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 11.5 Example of Cyclic STOP Mode Transition

11.4 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 the error programming operation.

For details on the write-protected sequence, see **Section 4, Write-Protected Registers**.

11.5 Cautions when Using Stand-By Modes

11.5.1 Cautions when Shifting to DEEPSTOP Mode during Debugger Use

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 waits in the interval between reset release and the DEEPSTOP execution instruction so that the debugger starts normally.

Section 12 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/F1L 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.

12.1 Features of RH850/F1L LPS

12.1.1 Number of Units

This microcontroller has the following number of units of the LPS.

Table 12.1 Units

Product	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	1	1	1	1
Name	LPSn (n = 0)	LPSn (n = 0)	LPSn (n = 0)	LPSn (n = 0)	LPSn (n = 0)	LPSn (n = 0)

Table 12.2 Unit Configurations and Channels

Unit Name LPSn	Channels per Unit	Function	Channel Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
LPS0	1	Digital port input m for port polling	DPINm	3 ch	8 ch	12 ch	17 ch	24 ch	24 ch
		Analog input m for A/D converter	ADCA0Im	8 ch	10 ch	11 ch	16 ch	16 ch	16 ch

Table 12.3 Index

Index	Meaning
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 letter “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”.

The following table shows values indicated by the indexes of each product.

Table 12.4 Indexes of Products

Indexes of Each Product	
48 pins	**64 pins, 80 pins, 100 pins, 144 pins, 176 pins**
k = —	k = 0 to 2
x = 0 to 3	x = 0 to 3

12.1.2 Register Base Address

LPS base addresses are listed in the following table.

LPS register addresses are given as offsets from the base addresses in general.

Table 12.5 Register Base Address

Base Address Name	Base Address
<LPS0_base>	FFBC 2000_H

12.1.3 Clock Supply

The LPS clock supply is shown in the following table.

Table 12.6 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
LPSn	PCLK	EMCLK

12.1.4 Interrupt Request

LPS interrupt requests are listed in the following table.

Table 12.7 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
LPS0			
INTCWEND	Port polling end interrupt (LPS)	104	—
INTADCA0I0*1	ADCA0 SG1 end interrupt	10	4 (channels 0 to 7)
INTADCA0I1*1	ADCA0 SG2 end interrupt	11	5 (channels 0 to 7)
INTADCA0I2*1	ADCA0 SG3 end interrupt	12	6 (channels 0 to 7)

Note 1. Output from ADCA0.

12.1.5 Reset Sources

LPS reset sources are listed in the following table. LPS is initialized by these reset sources.

Table 12.8 Reset Sources

Unit Name	Reset Source
LPS0	Reset sources except during transition to DEEPSTOP mode (AWORES)

12.1.6 External Input/Output Signals

External input/output signals of LPS are listed below.

Table 12.9 External Input/Output Signals

Unit Signal Name	Outline	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

12.1.7 Internal Input/Output Signals

Internal input/output signals for connecting the LPS and STBC or the LPS and TAUJ are listed below.

Table 12.10 Internal Input/Output Signals

Unit Signal Name	Outline	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

12.2 Overview

12.2.1 Functional Overview

To supervise the external input without consuming CPU resources, the low-power sampler (LPS) can check the digital input ports and analog input ports without the CPU. The figure below shows a connection example between the main components of the LPS and the external circuit.

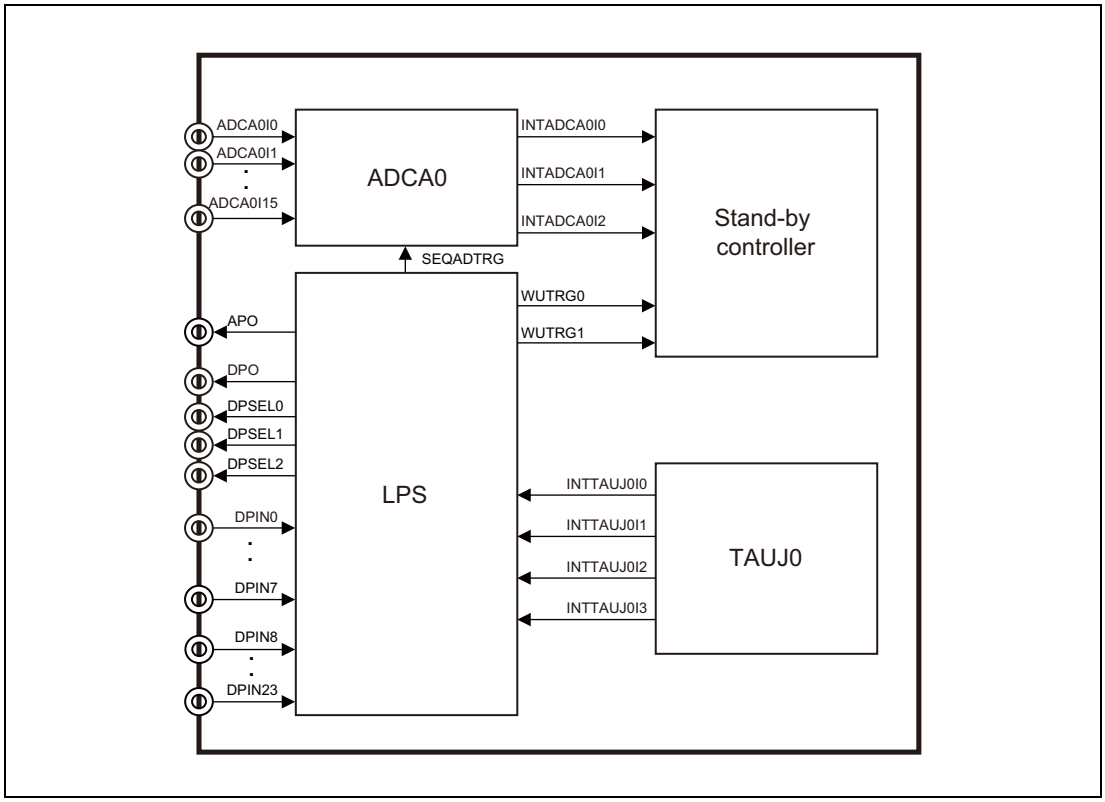


Figure 12.1 Block Diagram of the LPS

CAUTION

DPSEL2 to DPSEL0 and DPIN10 to DPIN8 are assigned to the same pins, thus cannot be used simultaneously.

12.3 Registers

12.3.1 List of Registers

LPS registers are listed in the following table.

For details about <LPS0_base>, see **Section 12.1.2, Register Base Address**.

Table 12.11 Registers

Module	Register	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

12.3.2 SCTLR — LPS Control Registers

This register makes settings for the LPS.

Access: This register can be read/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 12.12 SCTLR Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned. When writing to these bits, 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 SELDP[2:0] pins. **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.)
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 stopped 1: Analog input mode is enabled
0	DPEN	0: Digital input mode is stopped 1: Digital input mode is enabled

12.3.3 EVFR — Event Flag Register

This register indicates the result of comparison of the data (DPDIMR7 to DPDIMR0) which acquired the digital port input by the sequence operation with the compare target data (DPDSRH/DPDSRM/DPDSR0).

Access: This register can be read/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 12.13 EVFR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	DINEVF	This register indicates the result of comparison of the data (DPDIMR7 to DPDIMR0) which acquired the digital port input with the compare target data (DPDSRH/DPDSRM/DPDSR0). 0: The result of comparison is a match. 1: The result of comparison is a mismatch. This bit is set to 1 when a mismatch is detected even in one bit. Only 0 can be written to clear the bit.

12.3.4 DPSELRL0 — 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/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 12.14 DPSELRL0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
23 to 0	D0EN_n (n = 23 to 0)	These bits enable or disable comparison of the data (DPDIMR0) which acquired the first digital port input (DPINm) with the compare target data (DPDSR0). 0: Disables comparison. 1: Enables comparison.

12.3.5 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/written in 32-bit units.
DPSELRML and DPSELRMH can be read/written in 16-bit units.
DPSELR1, DPSELR2, DPSELR3, and DPSELR4 can be read/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 12.15 DPSELRM Register Contents

Bit Position	Bit Name	Function
31 to 24	D4EN_n (n = 7 to 0)	These bits enable or disable comparison of the data (DPDIMR4) which acquired the fifth digital port input (DPINm) with the compare target data (DPDSR4). 0: Disables comparison. 1: Enables comparison.
23 to 16	D3EN_n (n = 7 to 0)	These bits enable or disable comparison of the data (DPDIMR3) which acquired the fourth digital port input (DPINm) with the compare target data (DPDSR3). 0: Disables comparison. 1: Enables comparison.
15 to 8	D2EN_n (n = 7 to 0)	These bits enable or disable comparison of the data (DPDIMR2) which acquired the third digital port input (DPINm) with the compare target data (DPDSR2). 0: Disables comparison. 1: Enables comparison.
7 to 0	D1EN_n (n = 7 to 0)	These bits enable or disable comparison of the data (DPDIMR1) which acquired the second digital port input DPINm) with the compare target data (DPDSR1). 0: Disables comparison. 1: Enables comparison.

12.3.6 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/written in 32-bit units.
DPSELRHL and DPSELRHH can be read/written in 16-bit units.
DPSELR5, DPSELR6, and DPSELR7 can be read/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 12.16 DPDSRH Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
23 to 16	D7EN_n (n = 7 to 0)	These bits enable or disable comparison of the data (DPDIMR7) which acquired the eighth digital port input (DPINm) with the compare target data (DPDSR7). 0: Disables comparison. 1: Enables comparison.
15 to 8	D6EN_n (n = 7 to 0)	These bits enable or disable comparison of the data (DPDIMR6) which acquired the seventh digital port input (DPINm) with the compare target data (DPDSR6). 0: Disables comparison. 1: Enables comparison.
7 to 0	D5EN_n (n = 7 to 0)	These bits enable or disable comparison of the data (DPDIMR5) which acquired the sixth digital port input (DPINm) with the compare target data (DPDSR5). 0: Disables comparison. 1: Enables comparison.

12.3.7 DPDSR0 — DPIN Data Set Register 0

This register specifies the data to be compared with the data (DPDIMR0) which acquired the digital port input by the sequence operation.

Write to the DPDSR0 register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: This register can be read/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 12.17 DPDSR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing to these bits, 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)

12.3.8 DPDSRM — DPIN Data Set Register M

This register specifies the data to be compared with the data (DPDIMR4 to DPDIMR1) which acquired the digital port input by the sequence operation.

Write to the DPDSRM register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: DPDSRM can be read/written in 32-bit units.
DPDSRML and DPDSRMH can be read/written in 16-bit units.
DPDSR1, DPDSR2, DPDSR3, and DPDSR4 can be read/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 12.18 DPDIMRM 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)

12.3.9 DPDSRH — DPIN Data Set Register H

This register specifies the data to be compared with the data (DPDIMR7 to DPDIMR5) which acquired the digital port input by the sequence operation.

Write to the DPDSRH register before the sequence operation is started (when the SOSTR.SOF bit = 0).

Access: DPDSRH can be read/written in 32-bit units.
DPDSRHL and DPDSRHH can be read/written in 16-bit units.
DPDSR5, DPDSR6, and DPDSR7 can be read/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 12.19 DPDSRH Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing to these bits, 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)

12.3.10 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 can be read only 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 12.20 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

12.3.11 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 can be read only 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 12.21 DPDIMR1 Register Contents

Bit Position	Bit Name	Function
7 to 0	D1M_n (n = 7 to 0)	The second digital port input (DPINm) data

12.3.12 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 can be read only 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 12.22 DPDIMR2 Register Contents

Bit Position	Bit Name	Function
7 to 0	D2M_n (n = 7 to 0)	The third digital port input (DPINm) data

12.3.13 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 can be read only 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 12.23 DPDIMR3 Register Contents

Bit Position	Bit Name	Function
7 to 0	D3M_n (n = 7 to 0)	The fourth digital port input (DPINm) data

12.3.14 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 can be read only 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 12.24 DPDIMR4 Register Contents

Bit Position	Bit Name	Function
7 to 0	D4M_n (n = 7 to 0)	The fifth digital port input (DPINm) data

12.3.15 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 can be read only 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 12.25 DPDIMR5 Register Contents

Bit Position	Bit Name	Function
7 to 0	D5M_n (n = 7 to 0)	The sixth digital port input (DPINm) data

12.3.16 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 can be read only 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 12.26 DPDIMR6 Register Contents

Bit Position	Bit Name	Function
7 to 0	D6M_n (n = 7 to 0)	The seventh digital port input (DPINm) data

12.3.17 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 can be read only 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 12.27 DPDIMR7 Register Contents

Bit Position	Bit Name	Function
7 to 0	D7M_n (n = 7 to 0)	The eighth digital port input (DPINm) data

12.3.18 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 for the first time is acquired
- In analog mode
  The time from when the APO output is set to 1 to the time when the LPS outputs the AD conversion trigger to the ADCA0

The CNTVAL register should be written 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/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 12.28 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 = $(1/f_{RH}) \times 16 \times$ 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 = $(1/f_{RH}) \times 16 \times$ CNT0n (set value)

12.3.19 SOSTR — LPS Operation Status Register

This register indicates the operating state of the LPS.

Access: This register can be read only 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 12.29 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 will be canceled.

12.4 Digital Input Mode

With the digital input port DPINm and the externally connected multiplexer, up to 64 input ports can be supervised as shown in **Table 12.30, Combination of Supervised 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 12.30 Combination of Supervised Ports

Combination (Number of Ports x Number of Checking)	Port for Use	Total Number
Direct mode When input ports are checked simultaneously without the external multiplexer Up to 24 ports × 1	DPIN23-DPIN0	Up to 24
Multiplexer mode When input ports are checked with the small number of pins and the external multiplexer Up to 8 ports × 8	DPIN7-DPIN0 DPSEL2-DPSEL0	Up to 64
MIX mode When input ports are checked with the combination of the above two Up to 14 ports x 1 + Up to 7 ports x 7	DPIN7-DPIN0 DPIN16-DPIN11 DPSEL2-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.

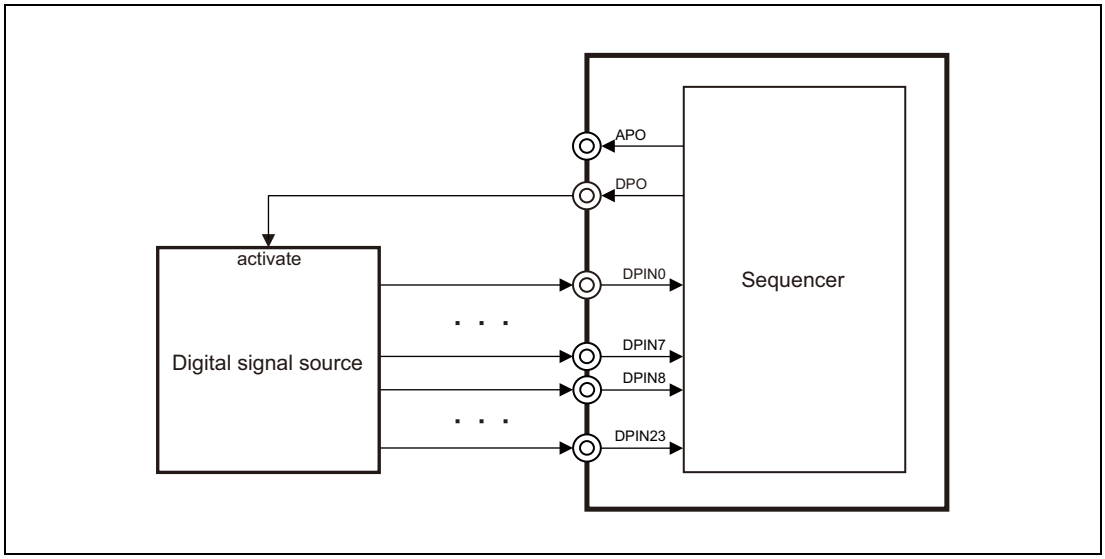


Figure 12.2 Direct Mode Connection Example

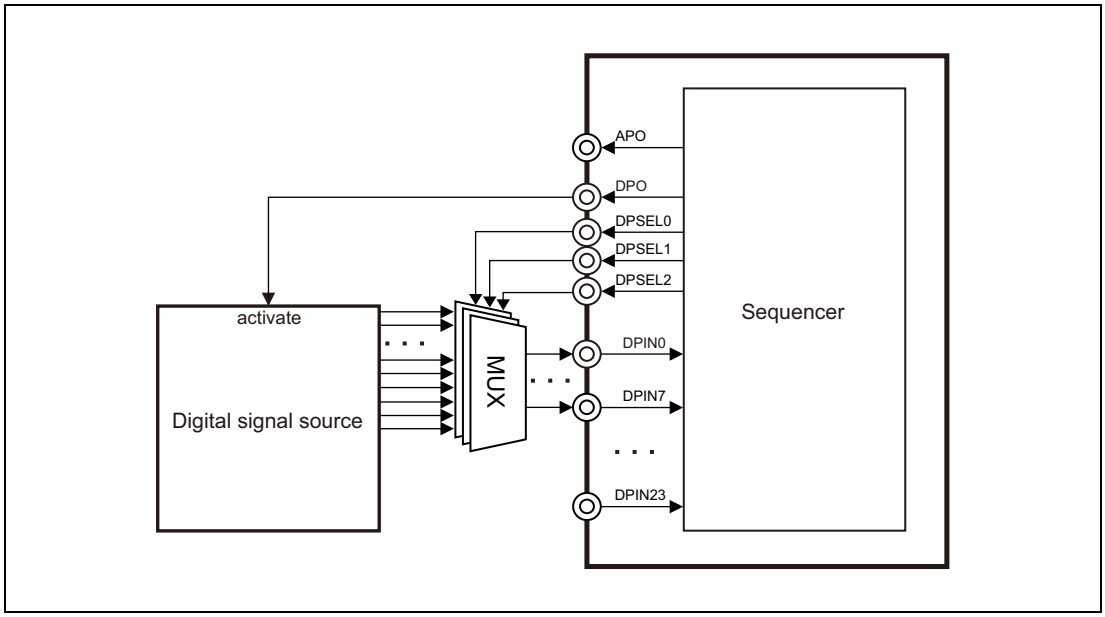


Figure 12.3 Multiplexer Mode Connection Example

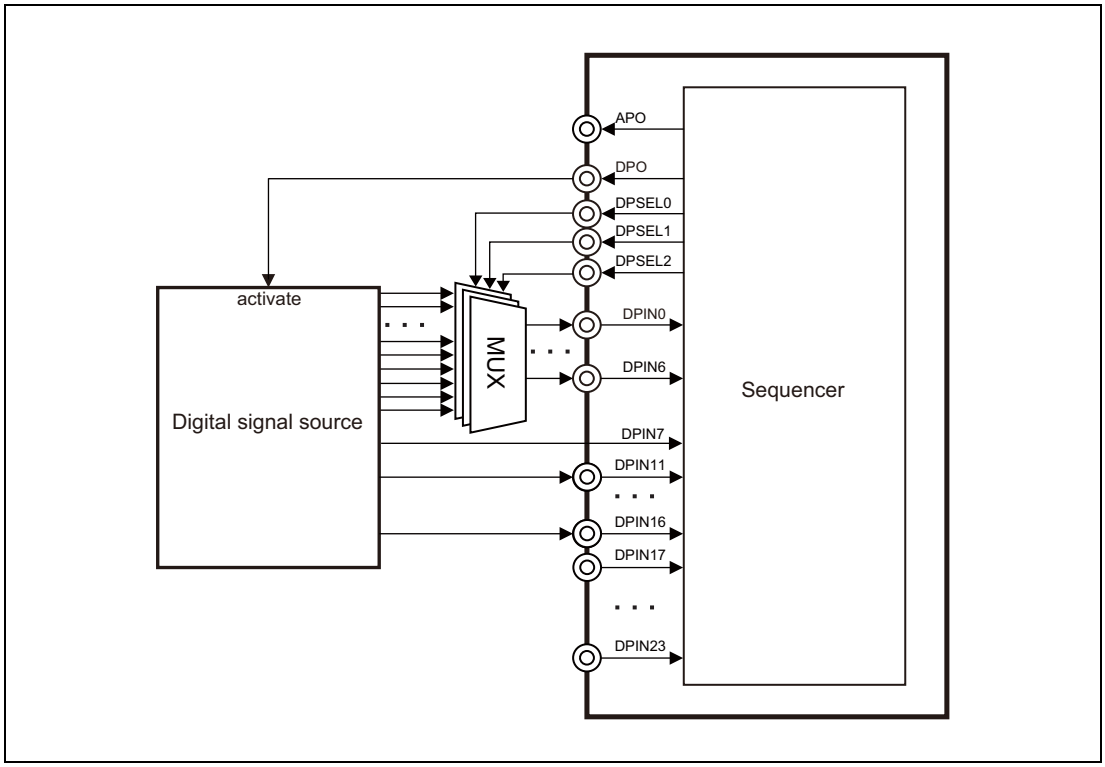


Figure 12.4 MIX Mode Connection Example

CAUTION

The DPSEL2-0 bits are assigned to the same alternate-function pins as DPIN10-8 and cannot be used as the same time.

Preparation

- Set the TAUJ0 interrupt to decide the timing and the number of times read by the multiplexor, by the SCTLR register.
- Set the interval timer to TAUJ0.
- Set the wait time of the digital signal source by the lower 8 bits in the CNTVAL register.
- Set expected values in the DPDSR0, DPDSRM and DPDSRH registers.
- Set the ports to be checked to the DPSELR0, DPSELRM, and DPSELRH registers.

Start

- Start the TAUJ0.
- Set the SCTLR.DPEN bit to 1.

After the operation starts, execute the port check by the interval set in TAUJ0. The operation continues regardless whether the mode is the RUN mode or power save mode. When the HS IntOsc stops in stand-by mode, the operation of the HS IntOsc will be resumed 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 figure shows an example of the operation in digital input mode.

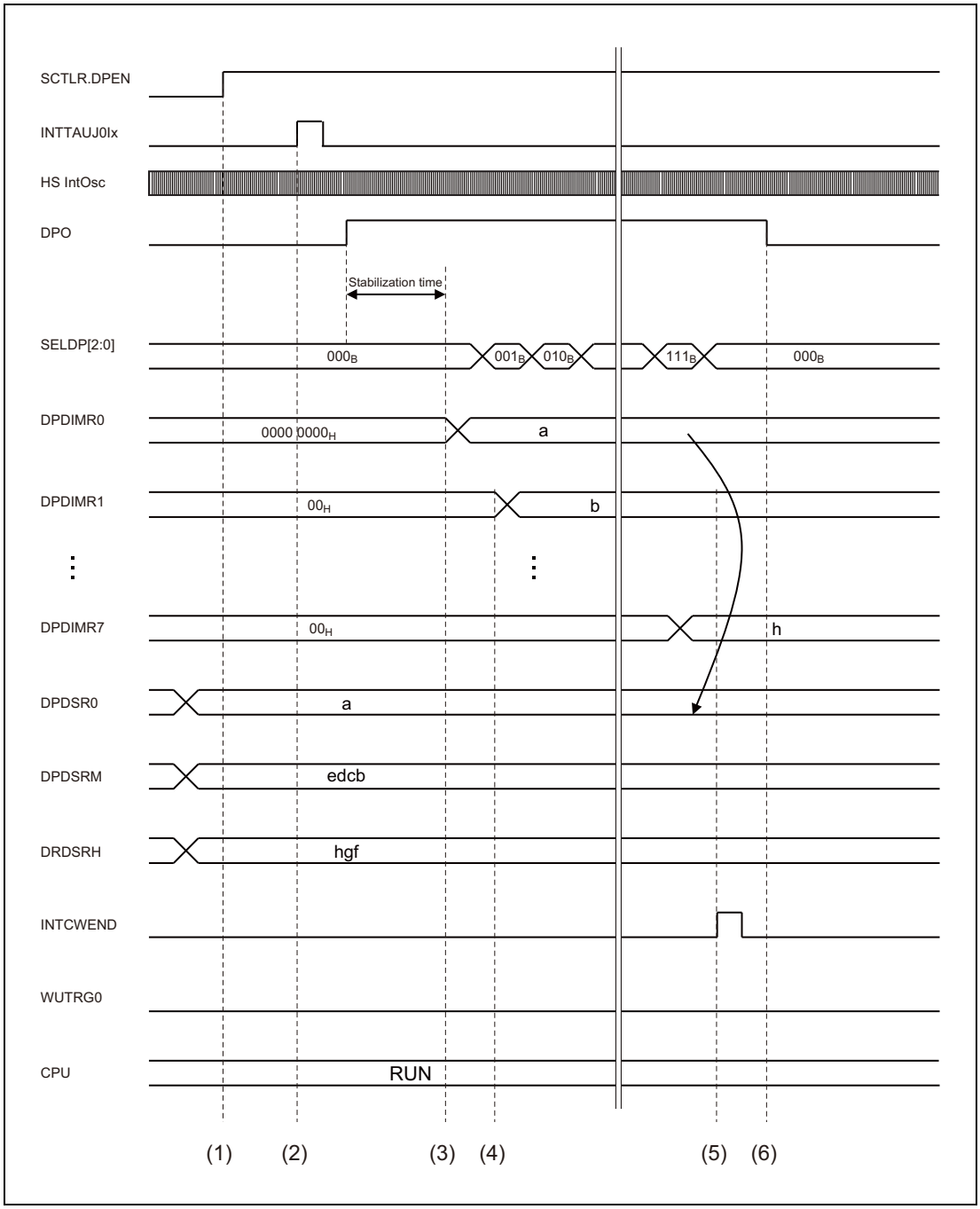


Figure 12.5 Operation of Digital Input Mode (8 Ports × 8) when the Input Value is not Changed (RUN Mode)

(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 SELDP[2:0] pins to switch the external multiplexer.

(4) After the switching of the SELDP[2:0] pins, the sequencer sequentially stores the value to the DPDIMR1 register and later and continues to increment the SELDP[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.

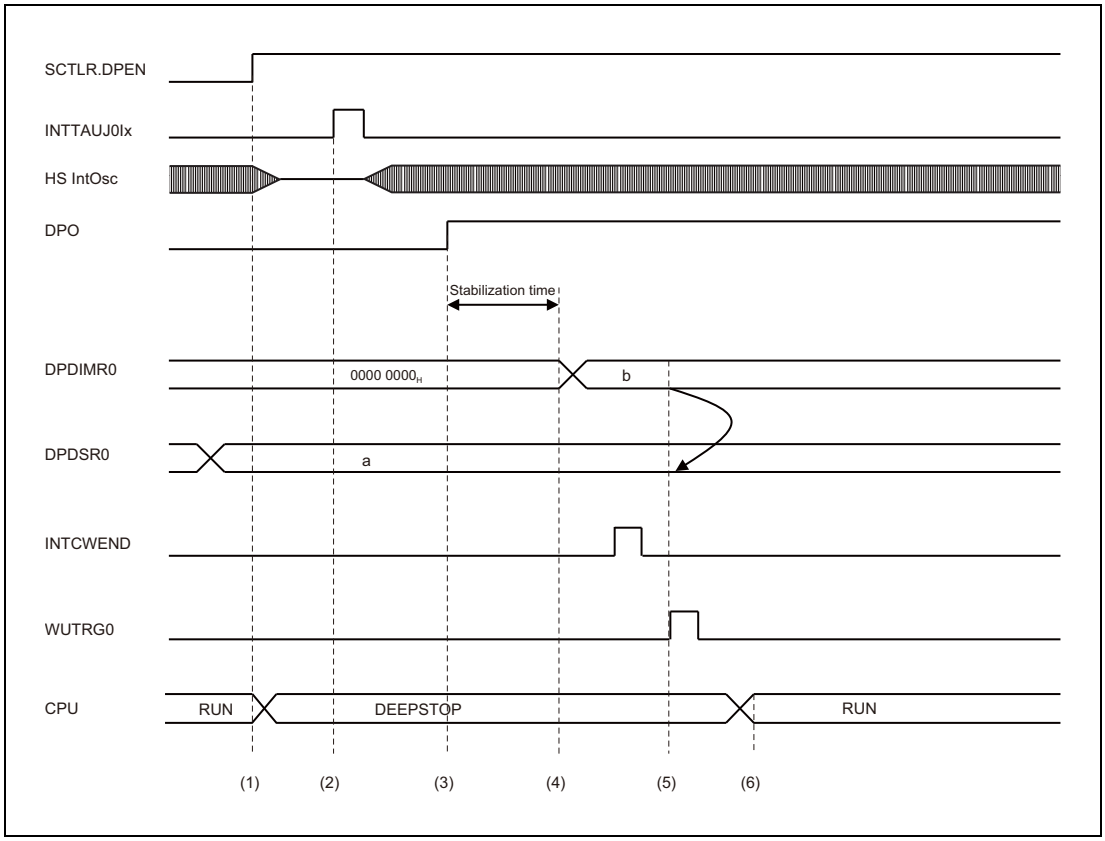


Figure 12.6 Operation of Digital Input Mode (24 Ports × 1) when the Input Value is Changed (DEEPSTOP Mode)

(1) Set the STBC0PSC.STBC0DISTRG bit to 1 to shift 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.

12.5 Analog Input Mode

The analog input port ADCA0Im (m = 0 to 15) can be supervised.

TAUJ0 is used to set the timing to check the value input to the port.

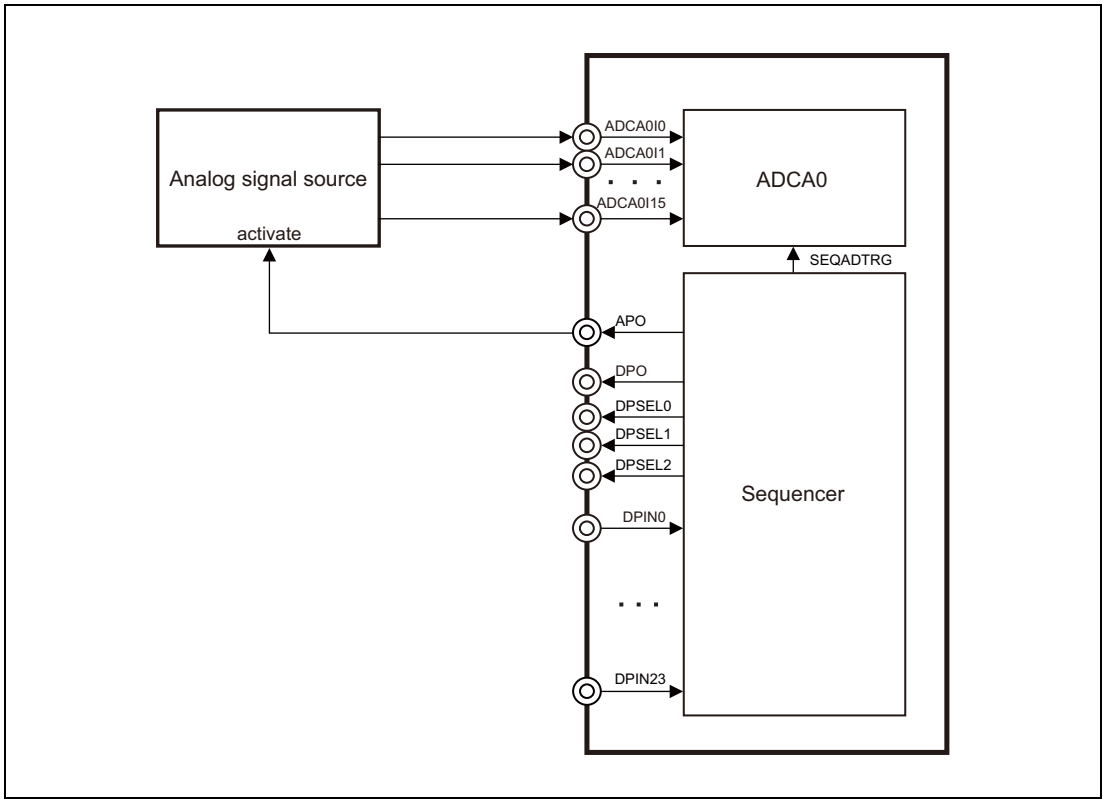


Figure 12.7 Analog Input Mode Connection Example

Preparation

- Set the TAUJ0 interrupt to decide the timing by the SCTLR register.
- Set the interval timer to TAUJ0.
- Set the wait time of the analog signal source by the upper 8 bits in the CNTVAL register.
- Set the ADCA0.

Start

- Start the TAUJ0.
- Set the SCTLR.ADEN bit to 1.

After the operation starts, execute the port check by the interval set in TAUJ0. The operation continues regardless whether the mode is the RUN mode or power save mode. When the HS IntOsc stops in stand-by mode, the operation of the HS IntOsc will be resumed while the sequencer is running.

To detect that the value input differs from the expected value, use the A/D error interrupt request (INTADCA0ERR) of the A/D converter.

In addition, if the 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 29.4.13, A/D Error Interrupt Request**[*1]. The following figure shows an example of the operation in analog input mode.

Note 1. In **Section 29 A/D Converter (ADCA)**, the name of the A/D error interrupt request is described as “INT_ADE”.

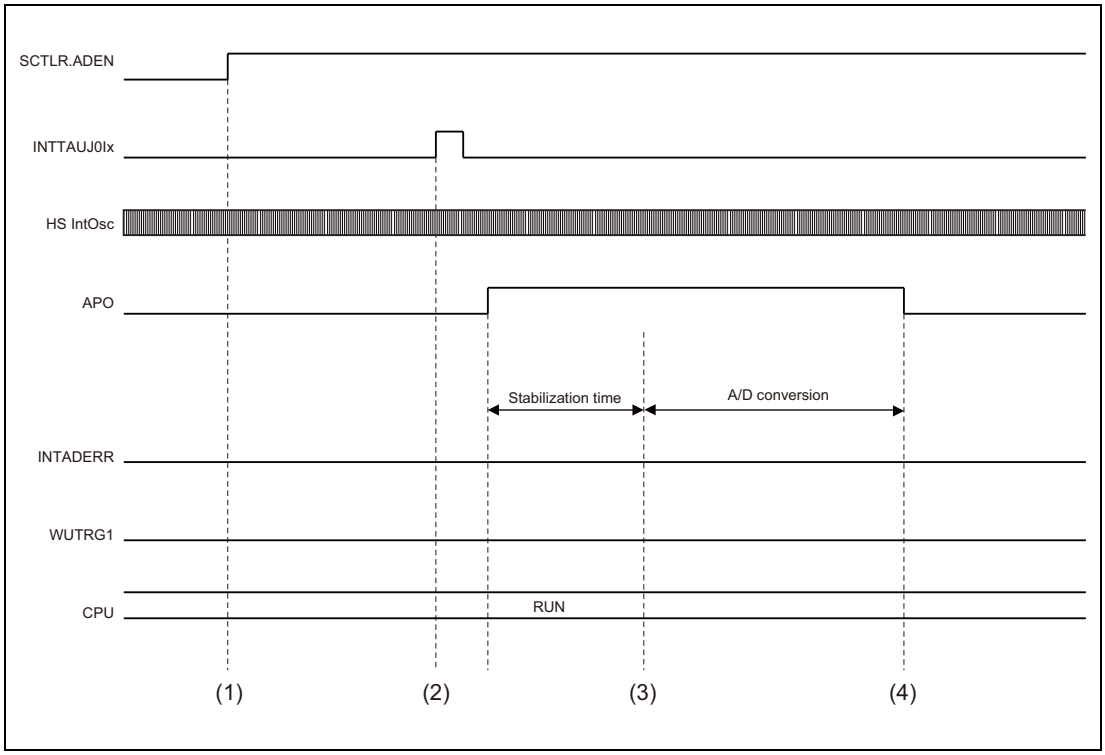


Figure 12.8 Operation of Analog Input Mode when the Conversion Result is within the Expected Range (RUN Mode)

(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.

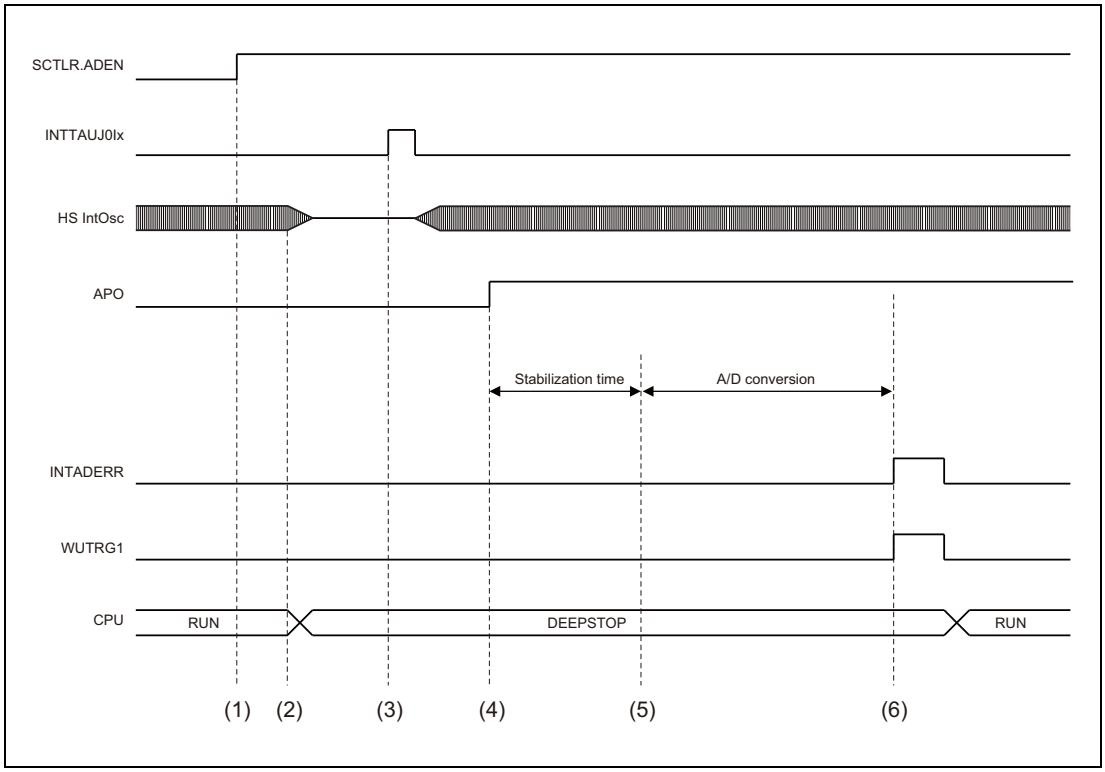


Figure 12.9 Operation of Analog Input Mode when the Conversion Result is not within the Expected Range (DEEPSTOP Mode)

(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 shift 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 13 External Memory Access (MEMC)

This section contains a generic description of the external memory access (MEMC).

The first part in this section describes the features specific to RH850/F1L microcontrollers, including register base addresses and input/output signals. The ensuing sections describe the functions and registers of MEMC.

13.1 Features of MEMC

13.1.1 Products that Incorporate MEMC

MEMC is incorporated in the following products.

Table 13.1 Products that Incorporate MEMC

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH 850/F1L 176 pins
Number of Units	—	—	—	—	—	1
Name	—	—	—	—	—	MEMCn (n = 0)

Table 13.2 Index

Index	Meaning
n	In this section, number of MEMC units are identified by “n” (n = 0).
x	In 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 13.3 External Memory Access Functions

Control Signal Name	Number of Control Signals
Address output	20 bits
Chip select output	4
Data bus	8/16 bits

Note 1. The lower 16 bits of the address output are multiplexed with the data bus.

13.1.2 Register Addresses

The MEMC base address is shown in the following table.

The MEMC register address is expressed as an offset of the base address.

Table 13.4 Register Base Addresses

Base Address Name	Base Address
<MEMC0_base>	FFFF 8200_H

13.1.3 Clock Supply

The MEMC clock supply is shown in the following table.

Table 13.5 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
MEMC	MEMC0CLK	CKSCLK_ICPUCLK*[1]

Note 1. CKSCLK_ICPUCLK/2 or CKSCLK_ICPUCLK/4 is selected as the clock using the option bytes.

13.1.4 Reset Sources

The MEMC reset sources are shown below. MEMC is initialized by the following reset sources.

Table 13.6 Reset Sources

Unit Name	Reset Source
MEMC	All reset sources (ISORES)

13.1.5 External Input/Output Signals

External input/output signals of MEMC are listed below.

Table 13.7 External Input/Output Signals

Unit Signal Name	Description	Alternative Port Pin Signal Name
MEMC		
MEMC0A[19:16]	Address bus output signal	MEMC0Am[19:16]
MEMC0AD[15:0]	Address/data bus I/O signal	MEMC0ADm[15:0]
MEMC0ASTB	Address strobe output signal	$\overline{\text{MEMC0ASTB}}$
MEMC0BEN[1:0]	Byte enable output signal	$\overline{\text{MEMC0BENm[1:0]}}$
MEMC0CLK	Bus clock output signal	MEMC0CLK
MEMC0CS[3:0]	Chip select output signal	$\overline{\text{MEMC0CSm[3:0]}}$
MEMC0RD	Read strobe output signal	$\overline{\text{MEMC0RD}}$
MEMC0WAIT	External wait request input signal	$\overline{\text{MEMC0WAIT}}$
MEMC0WR	Write strobe output signal	$\overline{\text{MEMC0WR}}$

13.2 Overview

The external memory access controller provides four chip select areas, and the bus width and wait time are selectable in each chip select area.

Moreover, the bus clock (CPU clock/2, CPU clock/4) is selectable using the option bytes. For details, see **Section 35.11.2, OPBT0 — Option Byte 0 Register**.

13.2.1 Functional Overview

The main features of the external memory access controller:

- Multiplexed bus mode
- Four chip select areas
- Selectable data bus width for each chip select area (8 or 16 bits)
- The data endian format can be selected for each chip select area individually.
- The following wait functions can be set individually for each chip select area.
  - External wait on SRAM access cycles

13.2.2 Operation Mode, Connectable Memory Types

13.2.2.1 Multiplexed Bus

This is an operation mode that connects address output and data input/output to external memory using the same signal line. It is possible to reduce the number of pins required for external memory connection.

13.2.3 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.

13.2.4 Bus Sizing Function

The data bus size can be selected from 8 bits or 16 bits for each chip select area. The bus size can be selected from 8 bits or 16 bits for each chip select area. To execute access when the data size exceeds the selected bus width, divide the data into sizes smaller than the bus width by using the bus sizing function.

13.2.5 Data Endian Setting Function

The data endian (little endian/big endian) can be specified for the chip select areas.

13.2.6 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
- Address setup wait
- Address hold wait
- Idle cycle (after read cycle)

13.2.7 External Wait Function

When accessing SRAM data in a write cycle, 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.

13.3 Registers

13.3.1 List of Registers

The MEMC registers are listed in the following table.

For details on <MEMCn_base>, see **Section 13.1.2, Register Addresses**.

Table 13.8 List of Registers

Module Name	Register Name	Symbol	Address
MEMC	Data bus width configuration register	BSC	<MEMCn_base> + 00_H
	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> + $0C_H$
	Address wait configuration register 0	AWC	<MEMCn_base> + 10_H
	Idle cycle configuration register	ICC	<MEMCn_base> + 14_H

13.3.2 BSC — Data Bus Width Configuration Register

The BSC register is used to set the data bus width of the external bus for each chip select area.

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

Address: <MEMCn_base> + 00_H

Value after reset: 0055_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	BS3	—	BS2	—	BS1	—	BS0
Value after reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W

Table 13.9 BSC Register Contents

Bit Position	Bit Name	Function
15 to 7, 5, 3, 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
6, 4, 2, 0	BSx	Bus Size Setting These bits set the bus width of each chip select area. 0: 8 bits 1: 16 bits

The relationship between each chip select area and the control bit is shown below.

Table 13.10 Relationship between BSx Bit and Chip Select Area

Chip Select Area	BSx Bit
CS3 area	BS3
CS2 area	BS2
CS1 area	BS1
CS0 area	BS0

13.3.3 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 13.11 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 to these bits, 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 13.12 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

13.3.4 DWC — Data Wait Configuration Register

The DWC register is used to set the number of data wait states of the external bus.

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

Address: <MEMCn_base> + 08_H

Value after reset: 7777_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DW32	DW31	DW30	—	DW22	DW21	DW20	—	DW12	DW11	DW10	—	DW02	DW01	DW00
Value after reset	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
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 13.13 DWC Register Contents

Bit Position	Bit Name	Function
15, 11, 7, 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
14 to 12, 10 to 8, 6 to 4, 2 to 0	DWx2, DWx1, DWx0	Data Wait Setting These bits set the number of data wait states for each chip select area.

DWx2	DWx1	DWx0	Number of Data Wait States
0	0	0	No data wait
0	0	1	1 clock
0	1	0	2 clocks
0	1	1	3 clocks
1	0	0	4 clocks
1	0	1	5 clocks
1	1	0	6 clocks
1	1	1	7 clocks

The relationship between each chip select area and the control bit is shown below.

Table 13.14 Relationship between DWx2 to DWx0 Bits and Chip Select Area

Chip Select Area	DWx2 to DWx0 Bits
CS3 area	DW32, DW31, DW30
CS2 area	DW22, DW21, DW20
CS1 area	DW12, DW11, DW10
CS0 area	DW02, DW01, DW00

13.3.5 DHC — Data Hold Wait Configuration Register

The DHC register is used to set the number of data hold wait states for each chip select area in the write cycle of the external bus.

The data hold wait of (the DHC register setting + one cycle) is always 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: 0055_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	DH3	—	DH2	—	DH1	—	DH0
Value after reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R	R/W

Table 13.15 DHC Register Contents

Bit Position	Bit Name	Function
15 to 7, 5, 3, 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
6, 4, 2, 0	DHx	Data Hold Wait Setting These bits set the number of data hold wait states for each chip select area. 0: No data hold wait 1: 1 clock

The relationship between each chip select area and the control bit is shown below.

Table 13.16 Relationship between DHx Bits and Chip Select Area

Chip Select Area	DHx Bits
CS3 area	DH3
CS2 area	DH2
CS1 area	DH1
CS0 area	DH0

13.3.6 AWC — Address Wait Configuration Register

The AWC registers are used to set the address wait period of the external bus for each chip select area.

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

Address: <MEMCn_base> + 10_H

Value after reset: 5555_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	AHW3	—	ASW3	—	AHW2	—	ASW2	—	AHW1	—	ASW1	—	AHW0	—	ASW0
Value after reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W	R	R/W

Table 13.17 AWC Register Contents

Bit Position	Bit Name	Function
15, 13, 11, 9, 7, 5, 3, 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
14, 10, 6, 2	AHWx	Address Hold Wait Setting These bits set the number of address hold wait states for each chip select area. 0: No address hold wait 1: 1 clock
12, 8, 4, 0	ASWx	Address Setup Wait Setting These bits set the number of address setup wait states for each chip select area. 0: No address setup wait 1: 1 clock

The relationship between each chip select area and the control bit is shown below.

Table 13.18 Relationship between ASWx and AHWx Bits, and Chip Select Area

Chip Select Area	ASWx and AHWx Bits
CS3 area	ASW3, AHW3
CS2 area	ASW2, AHW2
CS1 area	ASW1, AHW1
CS0 area	ASW0, AHW0

13.3.7 ICC — Idle Cycle Configuration Register

The ICC registers are 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: These registers can be read or written in 16-bit units.

Address: <MEMCn_base> + 14_H

Value after reset: 1111_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	RIC3	—	—	—	RIC2	—	—	—	RIC1	—	—	—	RIC0
Value after reset	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Table 13.19 ICC Register Contents

Bit Position	Bit Name	Function
15 to 13, 11 to 9, 7 to 5, 3 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12, 8, 4, 0	RICx	Idle Cycle Setting Bits after Read Cycle These bits set the number of idle cycles for each chip select area after a read cycle. 0: No idle cycle 1: 1 clock

13.4 Functions

13.4.1 Bus Control Functions

13.4.1.1 Chip Select Output Function

The connected external memory area is divided into 4 chip select areas and managed, as shown in **Figure 13.1, 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 bus size and 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 for 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 next.

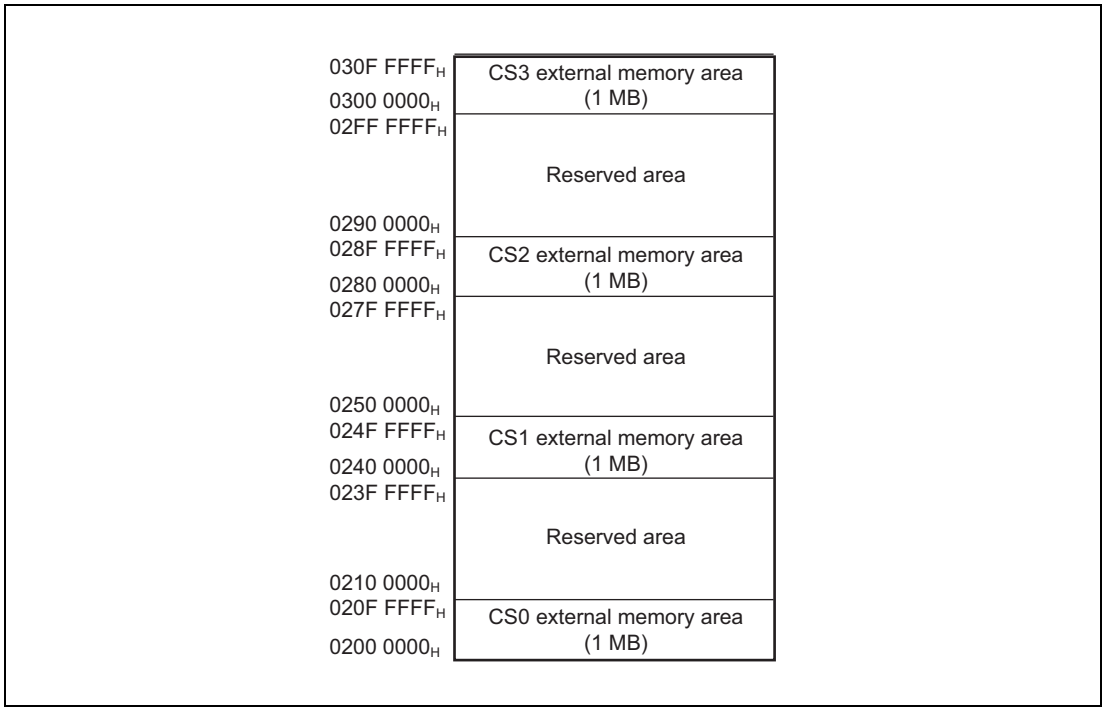
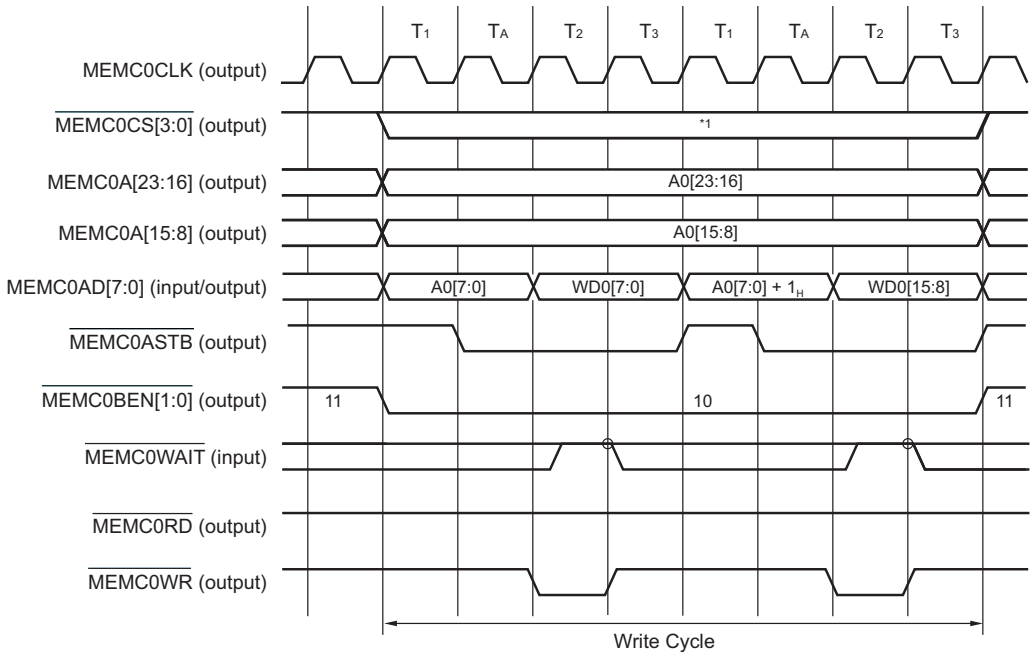


Figure 13.1 External Memory Map

13.4.1.2 Bus Size Setting 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.

The bit width of the external bus can be selected from 16 and 8 bits for each chip select area by setting the BSC 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_3: Data hold wait period

Figure 13.2 Bus Cycles when Making a 16-Bit Write Access to the 8-Bit External Bus

13.4.1.3 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. **The misaligned access with big endian is not supported.**

13.4.2 Wait Functions

Wait functions are listed below.

Table 13.20 Wait Functions

Wait Function	Data Wait		Data Hold Wait	Address Wait	Idle
	Programmable	External Wait			
Read	√	√	—	√	√
Write	√	√	√	√	—
Setting registers	DWC	—	DHC	AWC	ICC
Max. number of waits	7	—	1	1	1

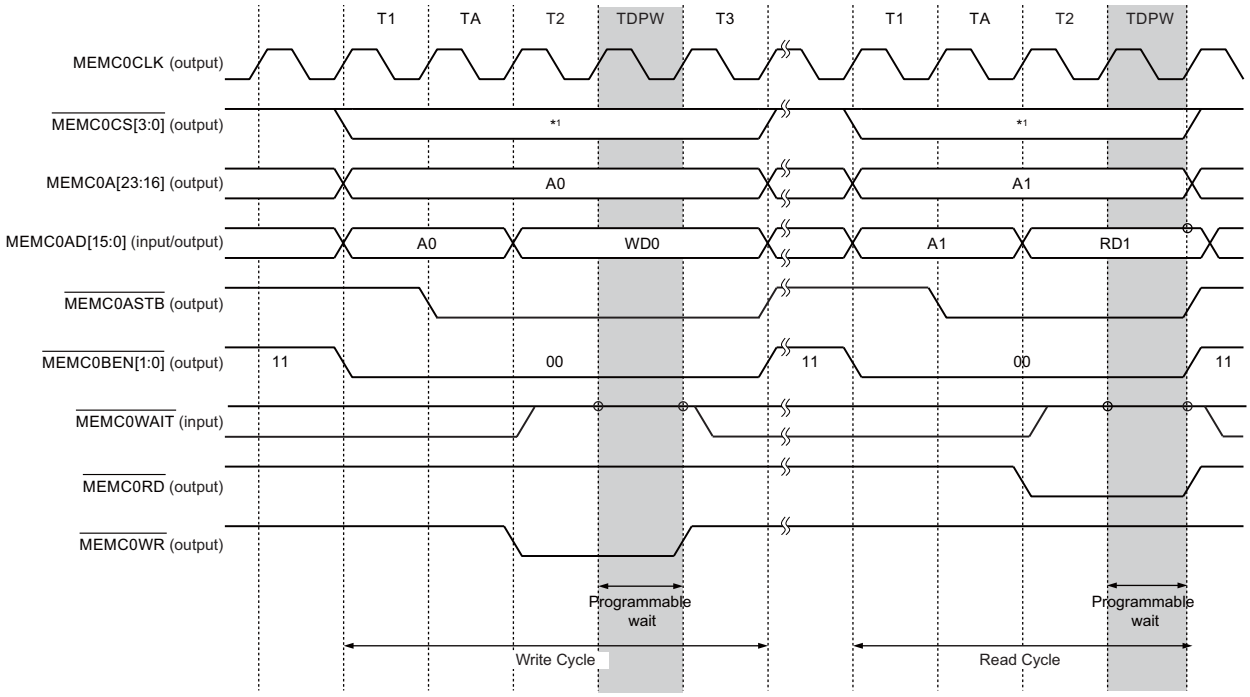
13.4.2.1 Programmable Data Wait Function

This wait function is for delaying the data latch timing by extending the read strobe and write strobe periods.

This function is enabled during data transfer.

Up to 7 cycles can be inserted.

Setting individual chip select areas with the DWC register is possible.



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_3: Data hold wait period

Figure 13.3 Bus Cycles when Inserting Programmable Data Wait

13.4.2.2 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_2 cycle and the T_{DPW}, T_{DEW} cycles.

Data wait cycles obtained by ORing the programmable data wait set by data wait control registers 0 and 1 (DWC0, DWC1 registers) and the external wait specification set by the $\overline{\text{MEMC0WAIT}}$ pin input, are inserted.

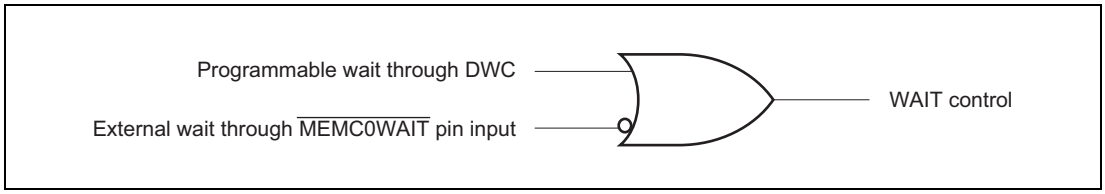
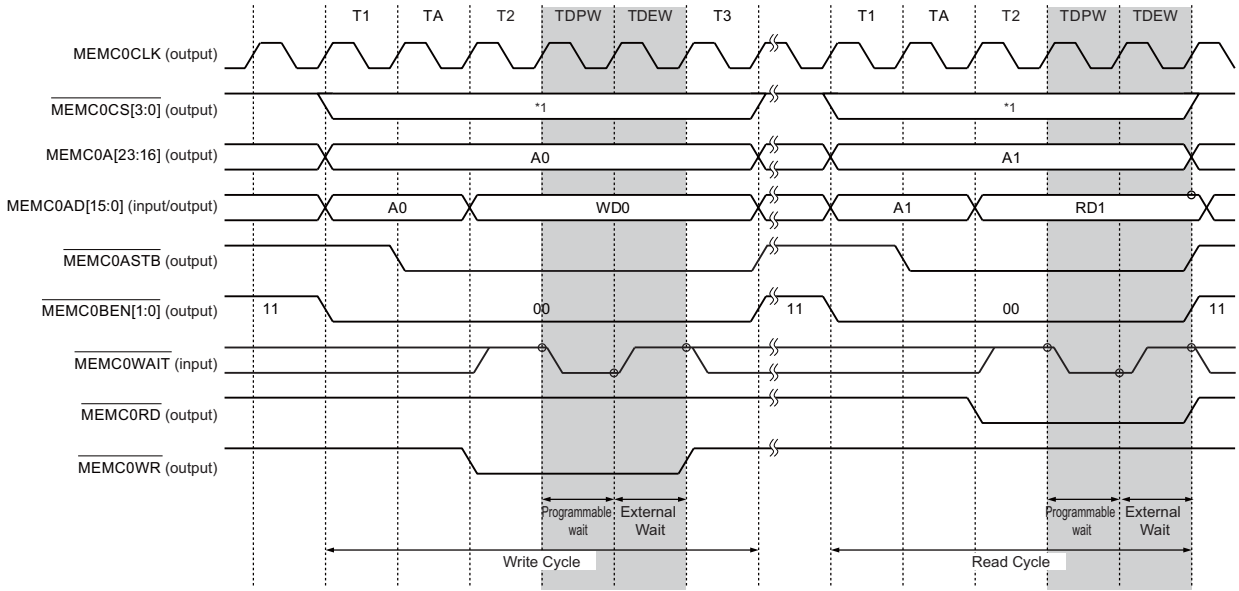


Figure 13.4 Internal Data Wait Generator



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_3: Data hold wait period

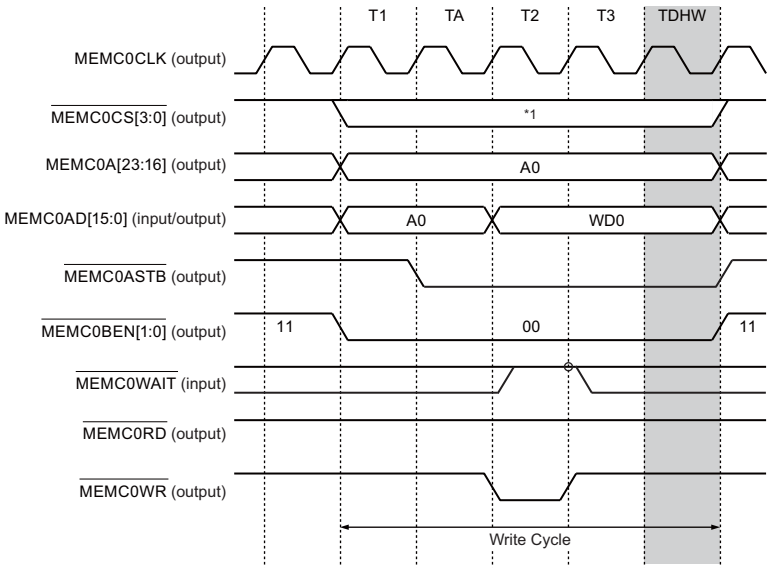
Figure 13.5 Bus Cycles when Inserting Programmable Data Wait and External Wait (while DWC = 1)

13.4.2.3 Data Hold Wait Function

This function inserts a wait for 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 extends the DHC register setting by 1 cycle, allowing insertion of 2 cycles.

The number of data hold wait extensions 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_3: Data hold period
T_{DHW}: Data hold wait period

Figure 13.6 Bus Cycles when Inserting Data Hold Wait

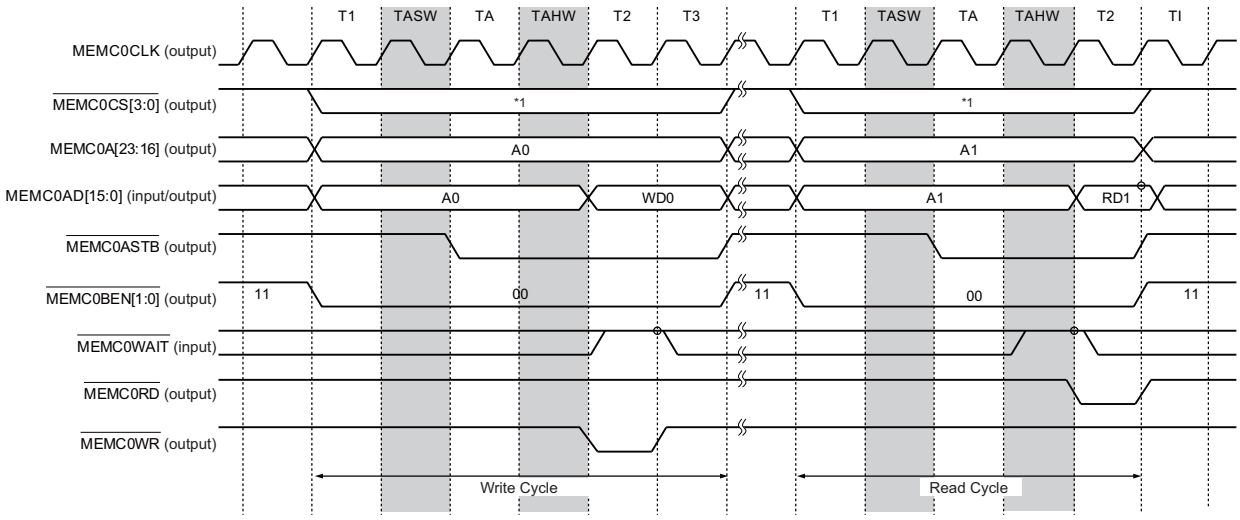
13.4.2.4 Address Setup Wait and Address Hold Wait Functions

The address setup wait function inserts a wait before the address transfer state in order to secure the setup time for the address strobe.

The address hold wait function inserts a wait in order to secure the hold time for the address strobe.

Up to 1 cycle can be inserted.

Setting each chip select area with the AWC register is possible.



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_{AHW}: Address hold wait period
T_2: Read (write) strobe period
T_3: Data hold period
T_I: Idle state period

Figure 13.7 Bus Cycles when Inserting Address Setup Wait and Address Hold Wait

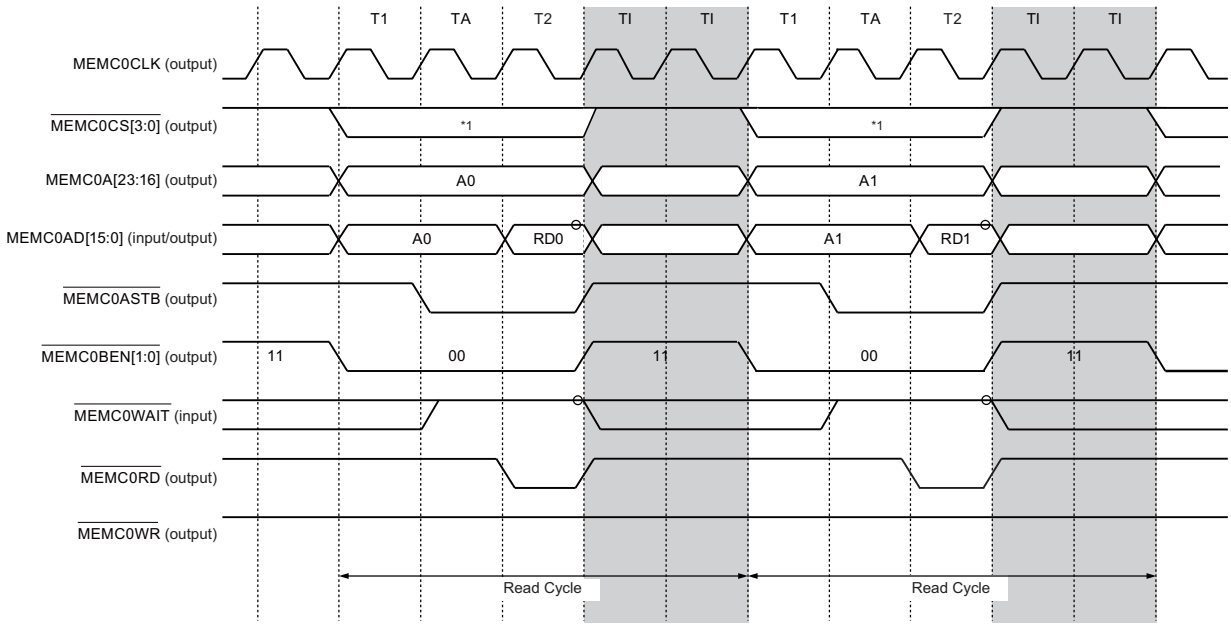
13.4.2.5 Idle Insertion Function

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 for each chip select area by setting the ICC registers.

CAUTION

The interval from completion of a bus cycle until the occurrence of subsequent bus cycles from the CPU (or DMA) lasts 1 cycle, regardless of the idle cycle setting. Therefore, a 1-cycle interval occurs between bus cycles even if the setting is no idle cycle.



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

Figure 13.8 Bus Cycles when Inserting Idle State

13.4.3 Memory Connection Example

13.4.3.1 SRAM Connection Example

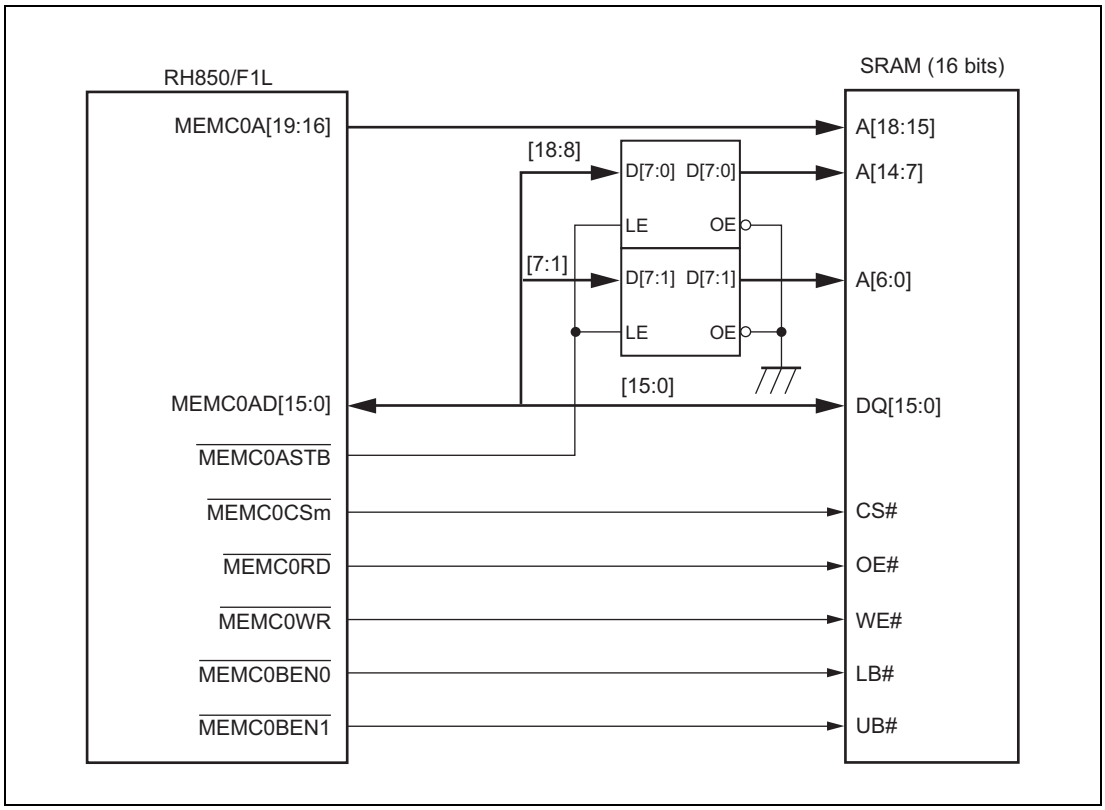


Figure 13.9 SRAM (16 Bits) Connection Example

13.4.4 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 the misaligned access, the CPU performs the division and combination of data. The data width and start address decide whether the access is misaligned or not.

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 of not a multiple of 4

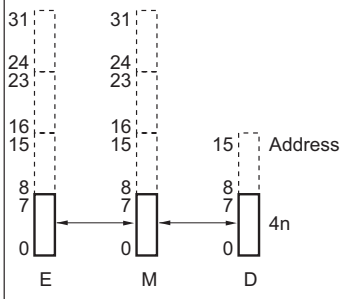
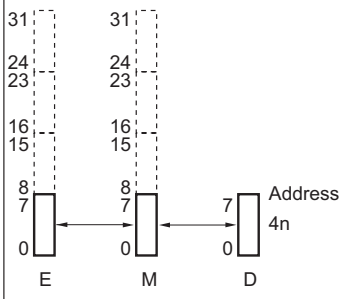
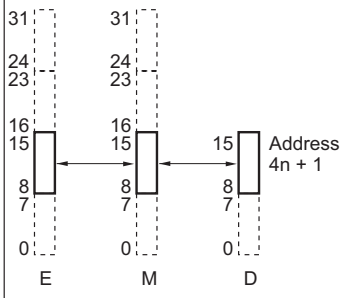
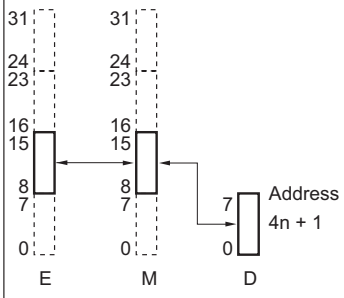
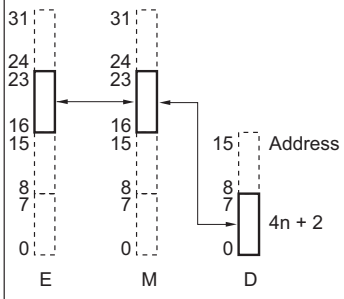
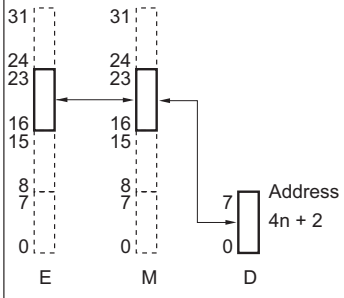
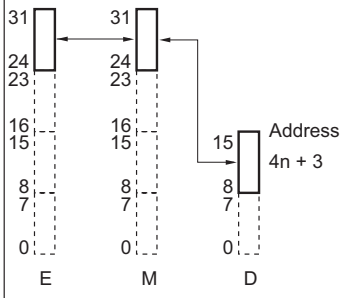
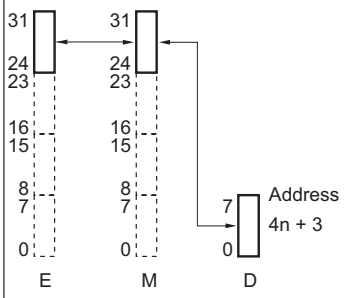
Table 13.21 Misaligned Access Generation Conditions

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 respective conditions are shown in the following pages.

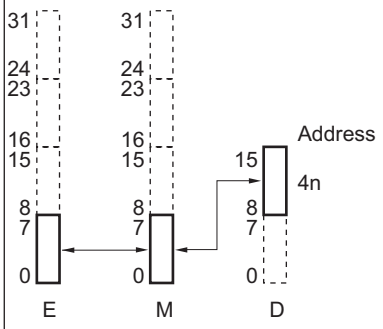
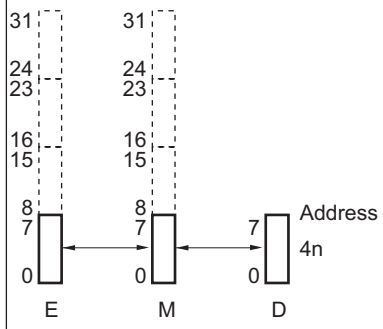
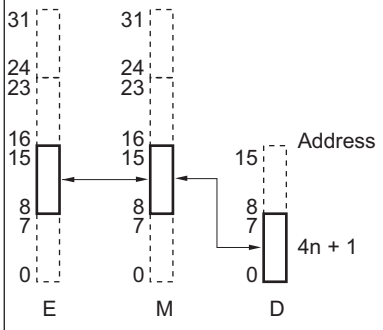
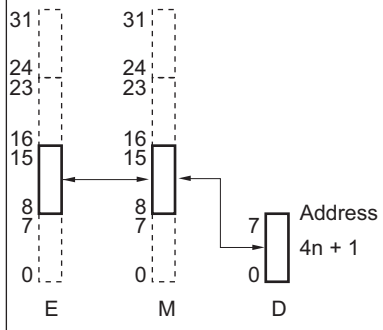
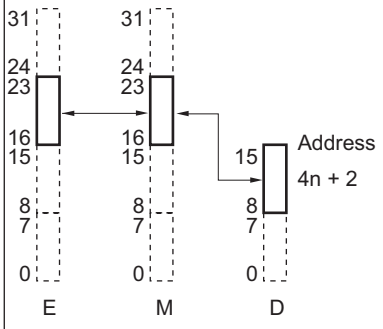
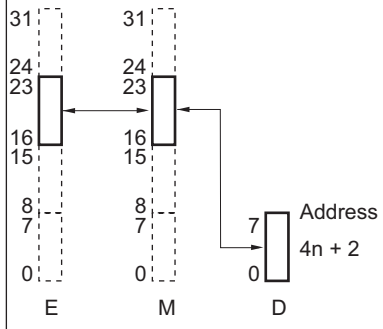
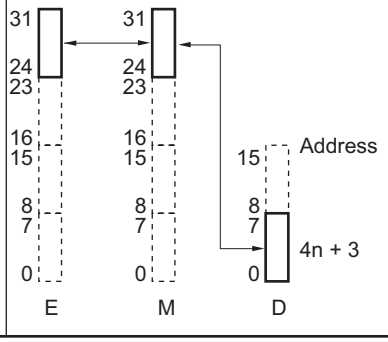
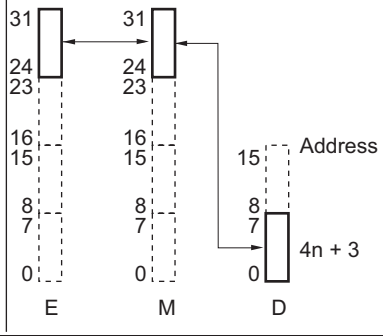
13.4.4.1 Data Flow for Byte Access (for Reading and Writing)

Table 13.22 Data Flow for Byte Access (Little Endian)

Address to be Accessed	Data Transfer Flow	
	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n		
4n + 1		
4n + 2		
4n + 3		

Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

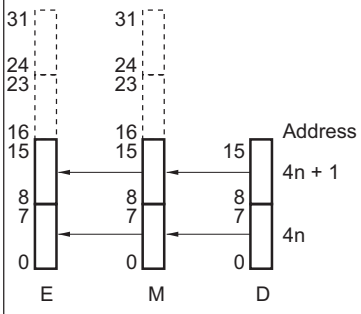
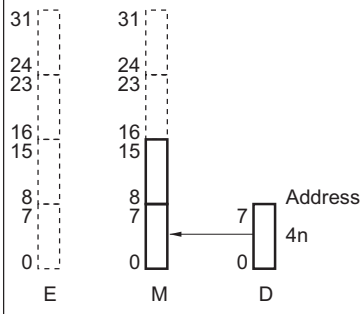
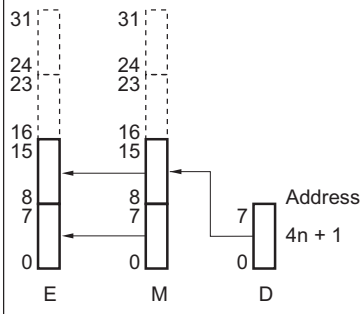
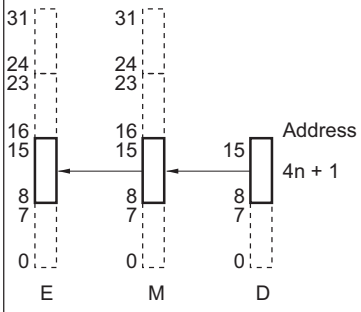
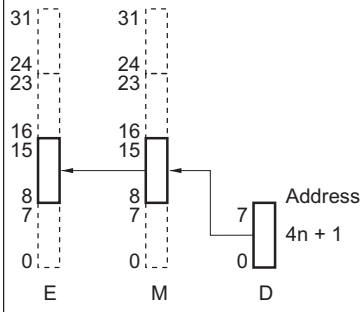
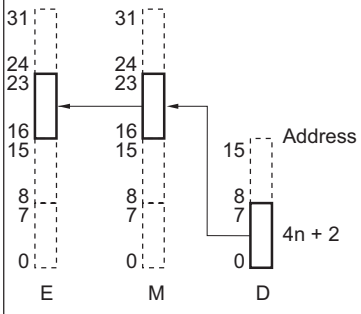
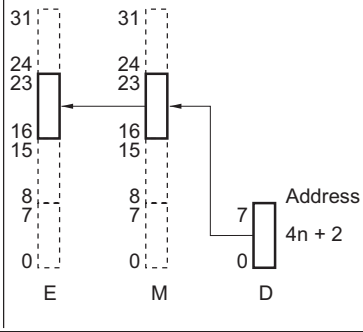
Table 13.23 Data Flow for Byte Access (Big Endian)

Address to be Accessed	Data Transfer Flow	
	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n		
4n + 1		
4n + 2		
4n + 3		

Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

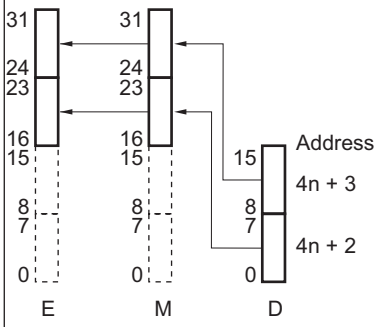
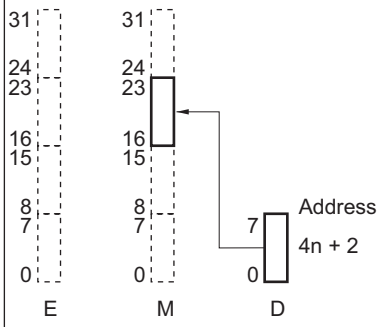
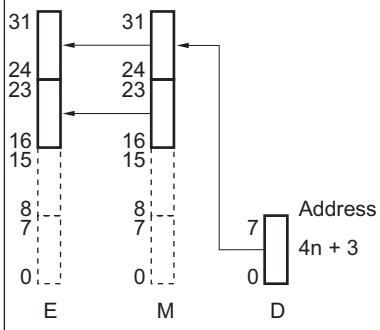
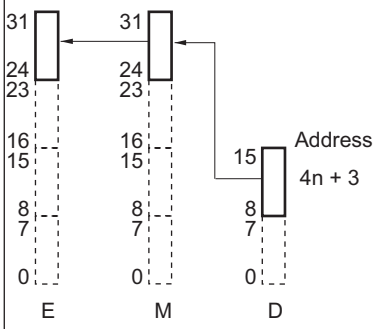
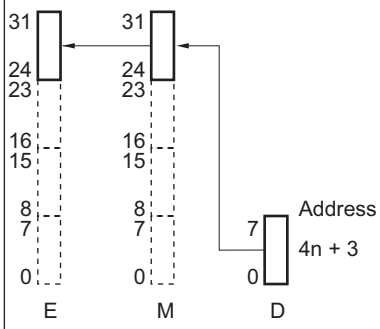
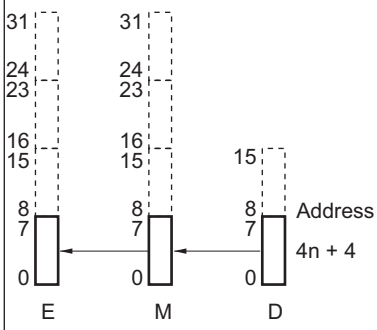
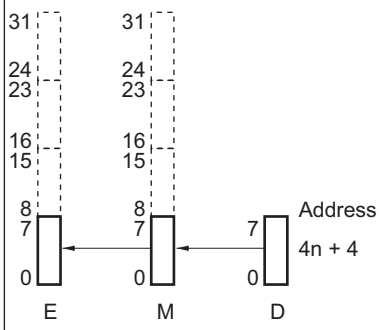
13.4.4.2 Data Flow for Half-Word Read Access

Table 13.24 Data Flow for Half-Word Read Access (Little Endian) (1/2)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st		
	2nd	—	
4n+1	1st		
	2nd		

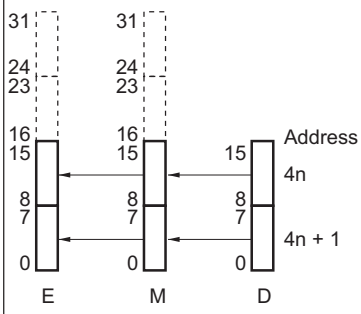
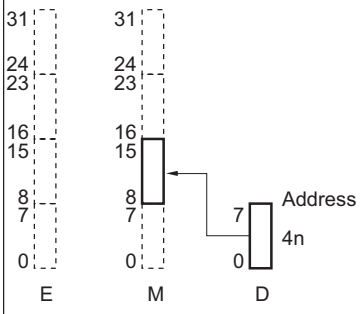
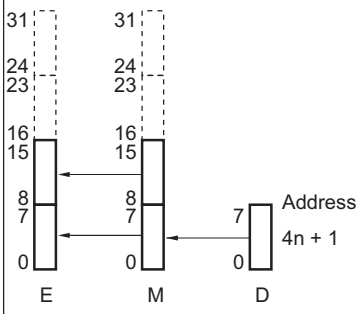
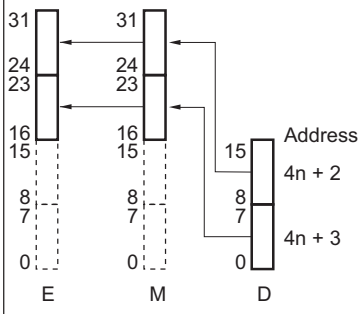
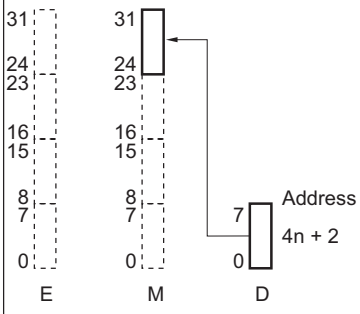
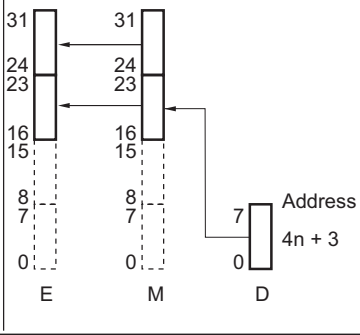
Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.24 Data Flow for Half-Word Read Access (Little Endian) (2/2)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+2	1st		
	2nd	—	
4n+3	1st		
	2nd		

Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.25 Data Flow for Half-Word Read Access (Big Endian)

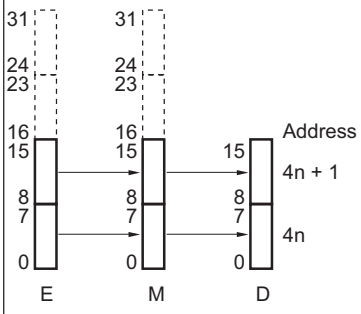
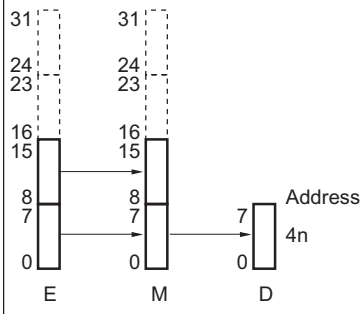
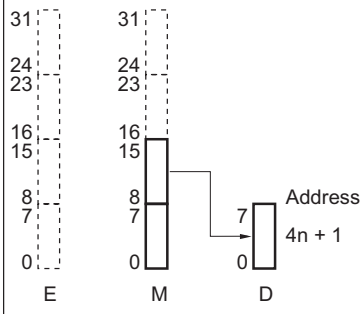
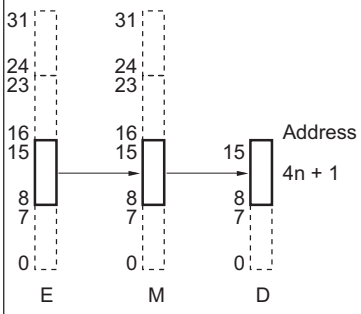
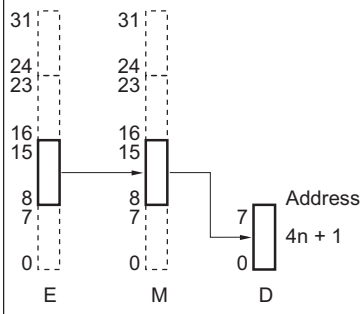
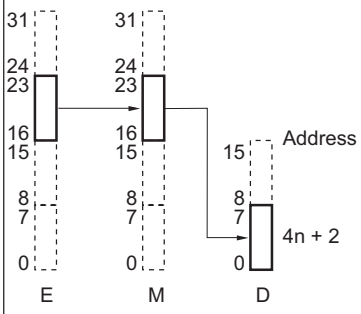
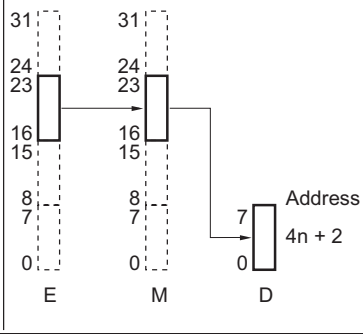
Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st		
	2nd	—	
4n+2	1st		
	2nd	—	

Note 1. E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing the address starting with 4n + 1 or 4n + 3 is prohibited.

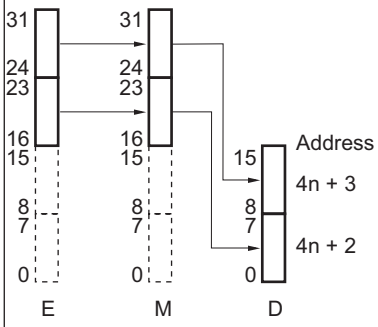
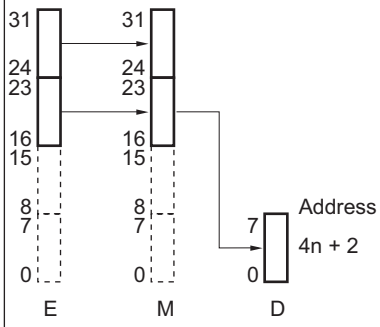
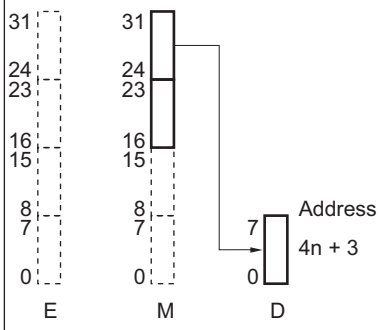
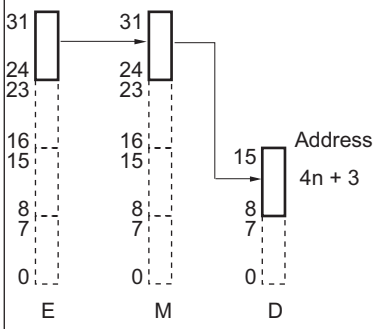
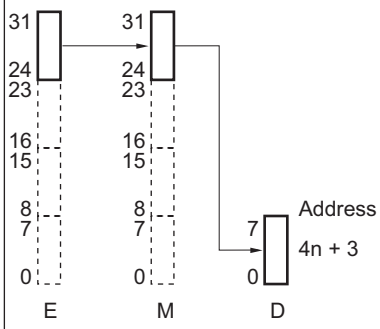
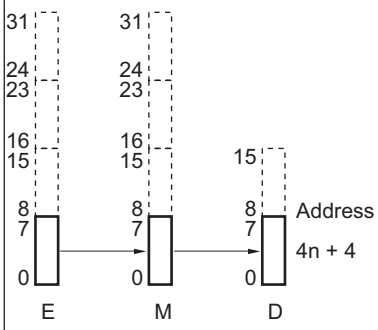
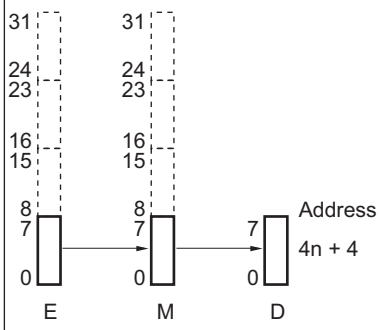
13.4.4.3 Data Flow for Half-Word Write Access

Table 13.26 Data Flow for Half-Word Write Access (Little Endian) (1/2)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st		
	2nd	—	
4n+1	1st		
	1st		

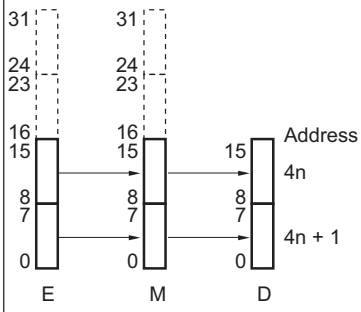
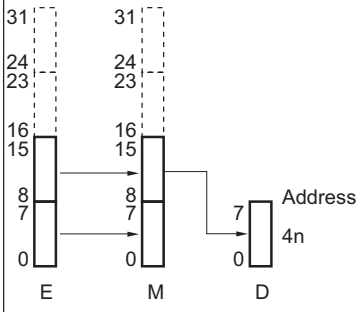
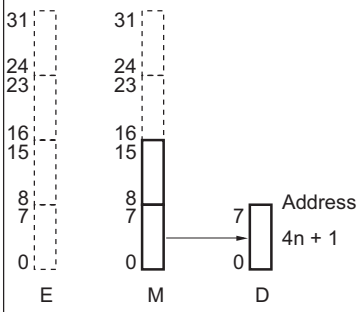
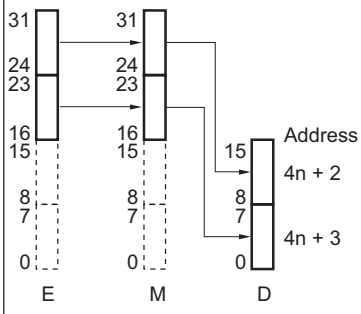
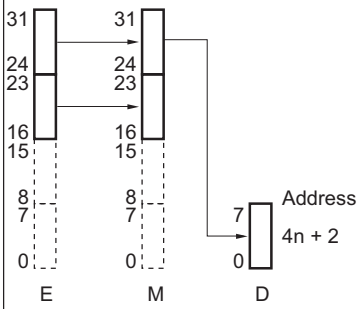
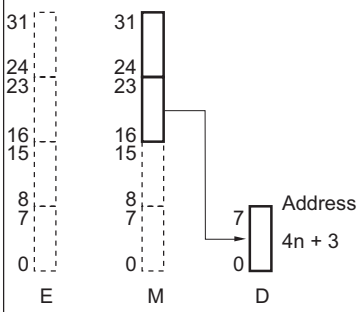
Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.26 Data Flow for Half-Word Write Access (Little Endian) (2/2)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+2	1st		
	2nd	—	
4n+3	1st		
	2nd		

Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.27 Data Flow for Half-Word Write Access (Big Endian)

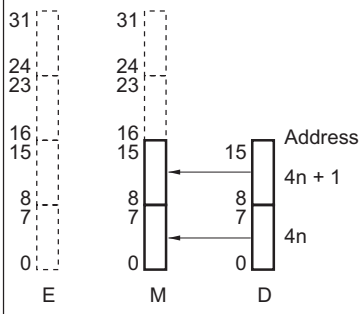
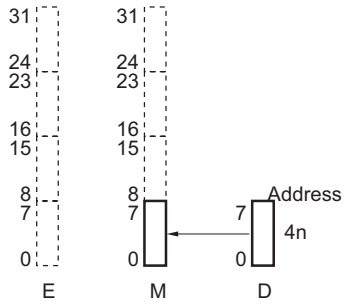
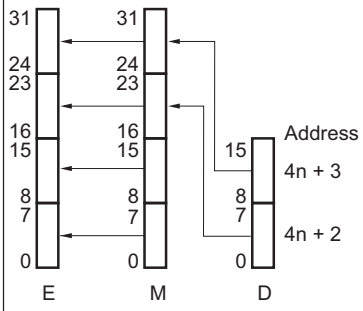
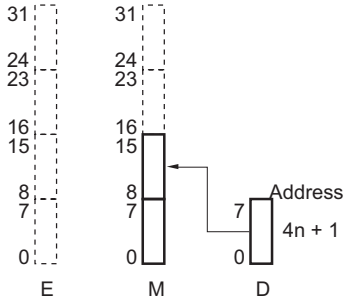
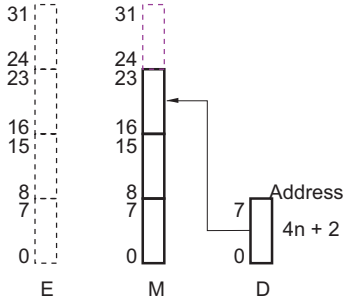
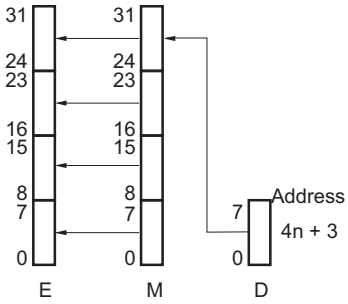
Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st		
	2nd	———	
4n+2	1st		
	2nd	———	

Note 1. E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing the address starting with 4n + 1 or 4n + 3 is prohibited.

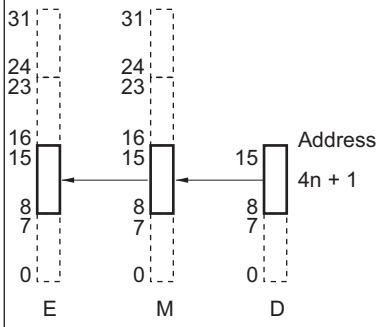
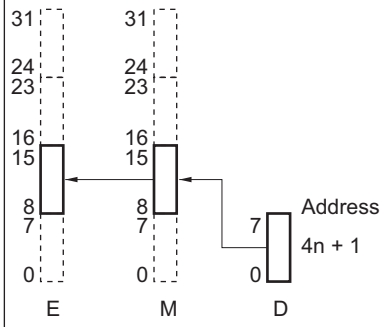
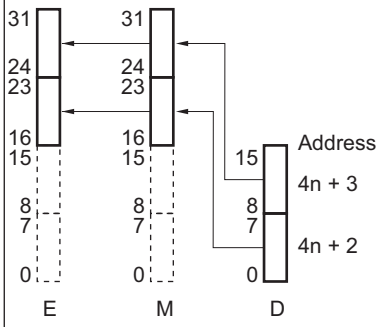
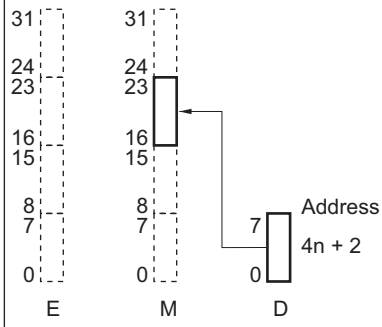
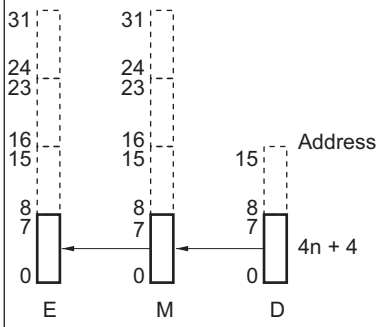
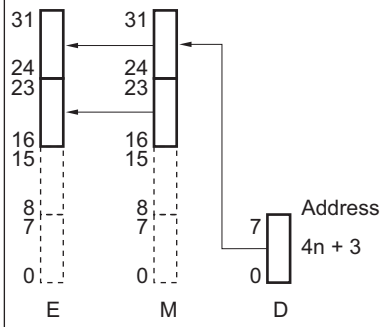
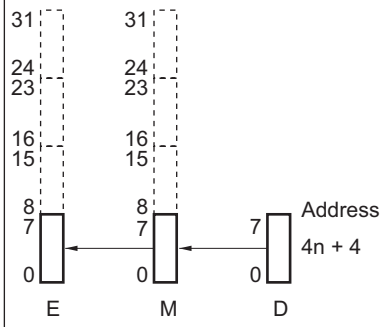
13.4.4.4 Data Flow for Word Read Access

Table 13.28 Data Flow for Word Read Access (Little Endian) (1/4)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st		
	2nd		
	3rd	—	
	4th	—	

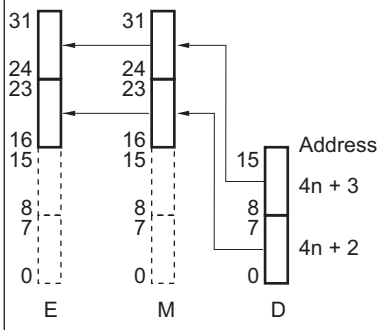
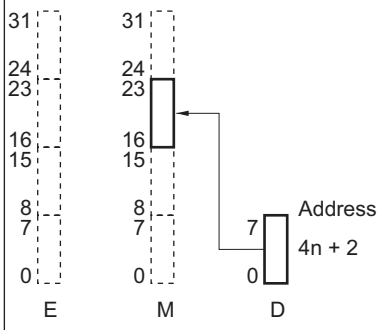
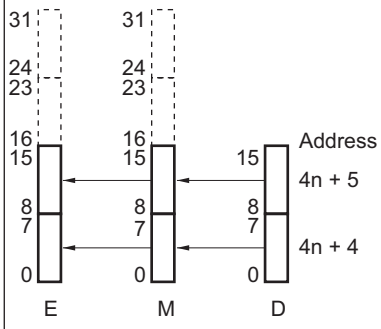
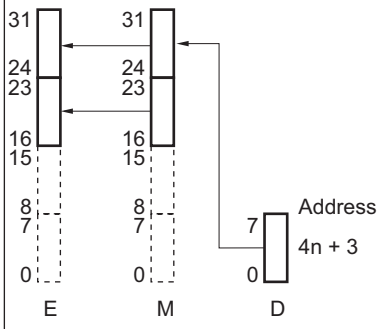
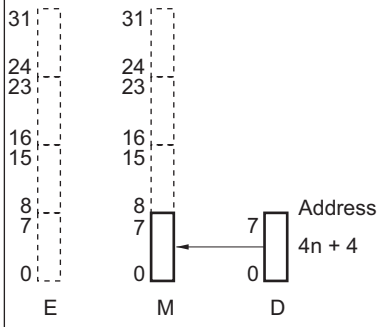
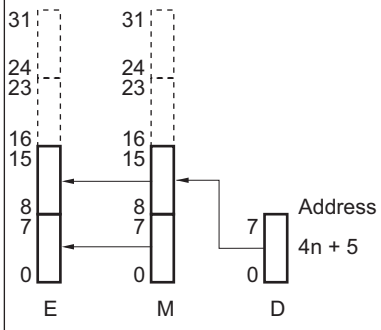
Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.28 Data Flow for Word Read Access (Little Endian) (2/4)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+1	1st		
	2nd		
	3rd		
	4th	———	

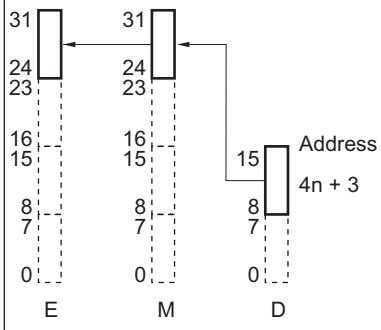
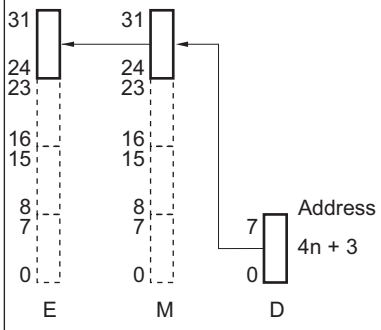
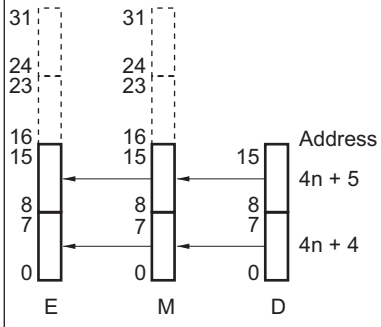
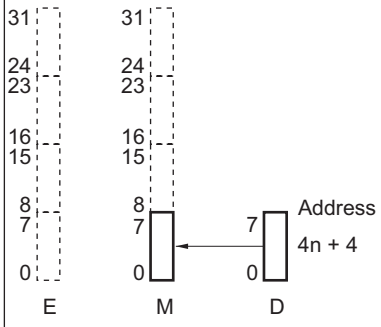
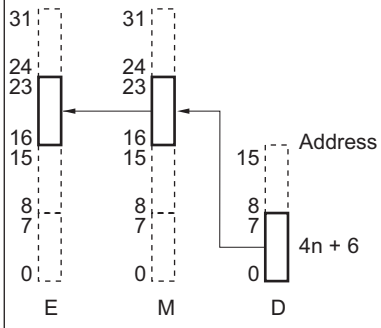
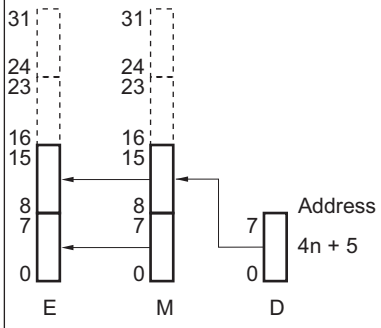
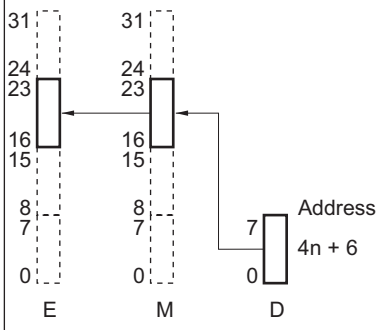
Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.28 Data Flow for Word Read Access (Little Endian) (3/4)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+2	1st		
	2nd		
	3rd	—	
	4th	—	

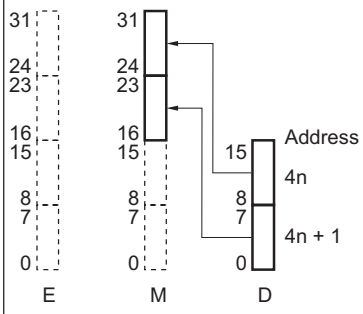
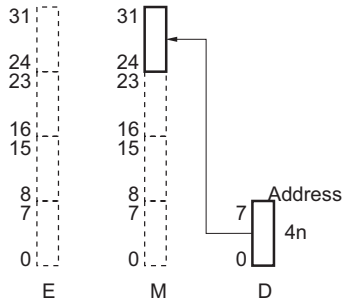
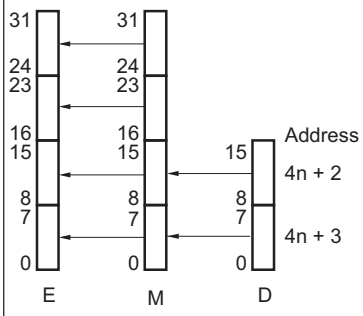
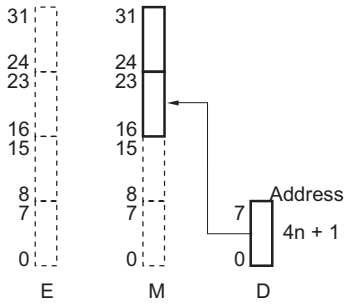
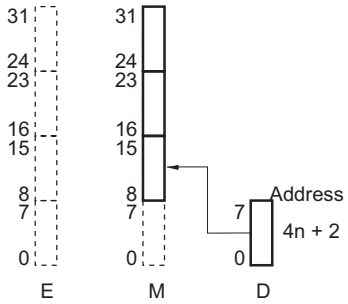
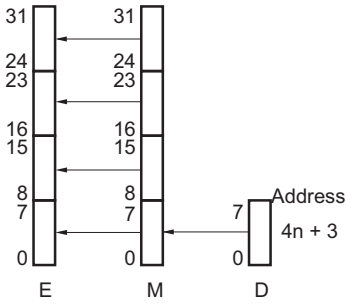
Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.28 Data Flow for Word Read Access (Little Endian) (4/4)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+3	1st		
	2nd		
	3rd		
	4th	—	

Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.29 Data Flow for Word Read Access (Big Endian)

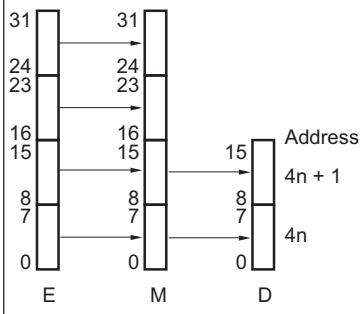
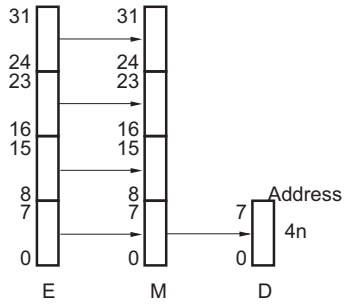
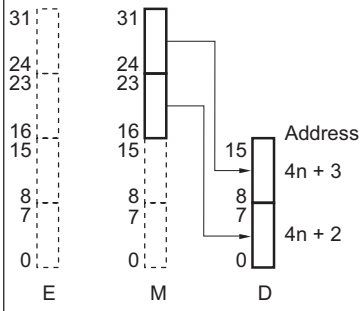
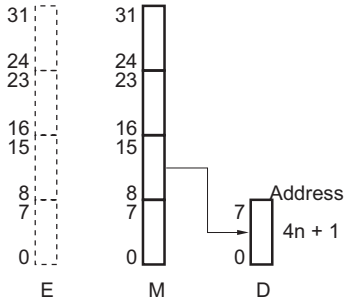
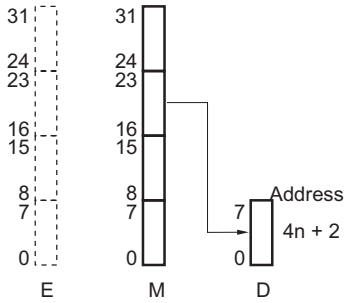
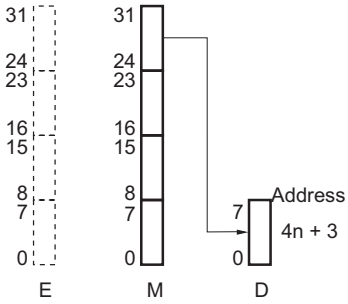
Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st		
	2nd		
	3rd	———	
	4th	———	

Note 1. E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing the address starting with 4n + 1, 4n + 2, or 4n + 3 is prohibited.

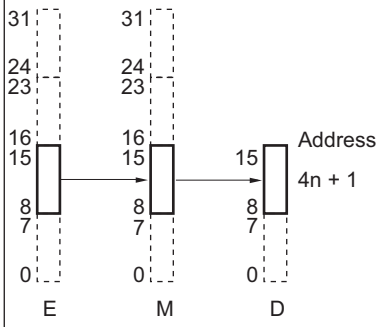
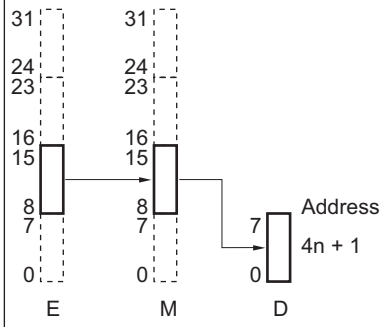
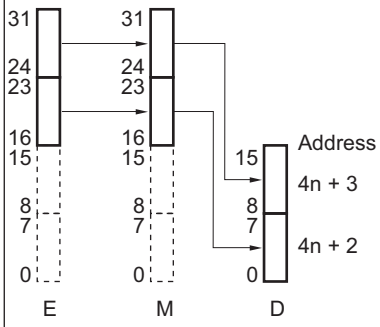
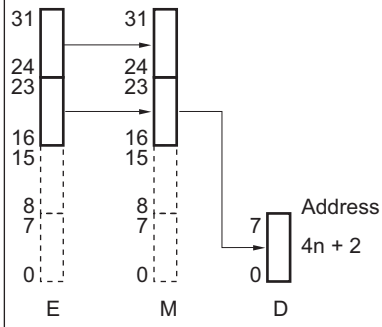
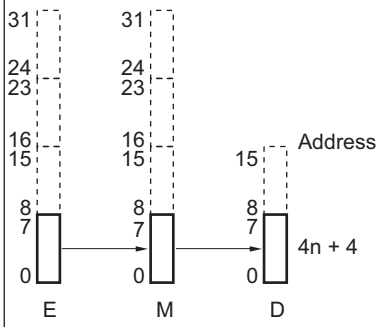
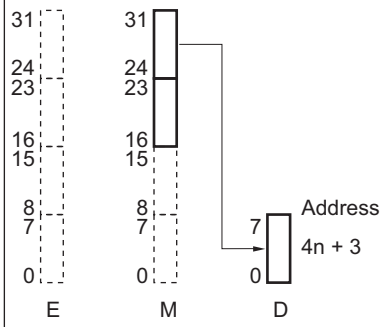
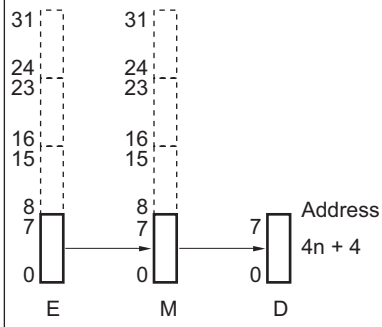
13.4.4.5 Data Flow for Word Write Access

Table 13.30 Data Flow for Word Write Access (Little Endian)(1/4)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st		
	2nd		
	3rd	—	
	4th	—	

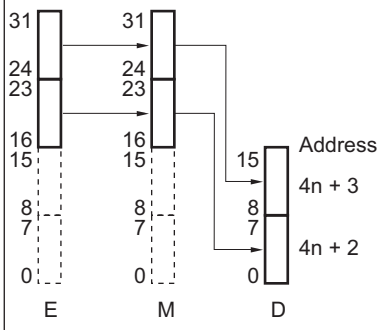
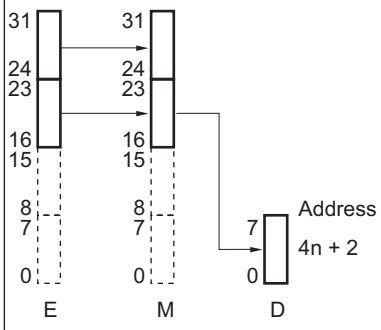
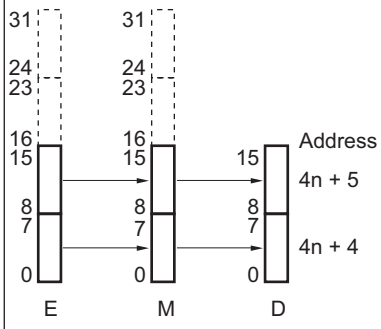
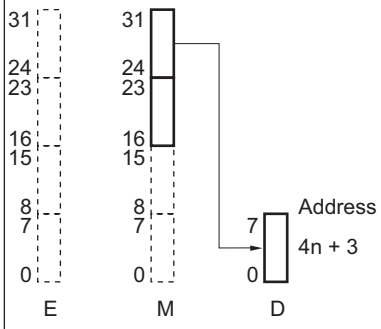
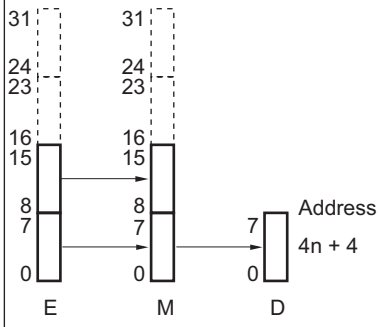
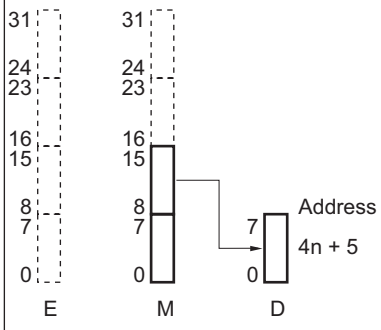
Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.30 Data Flow for Word Write Access (Little Endian) (2/4)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+1	1st		
	2nd		
	3rd		
	4th	—	

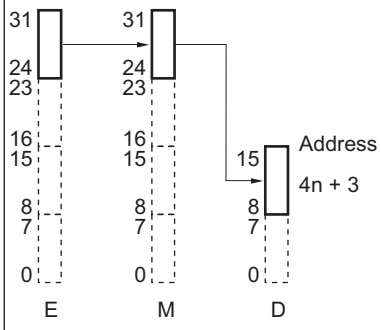
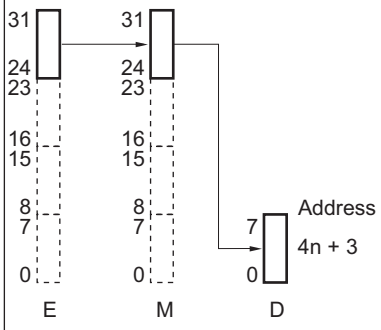
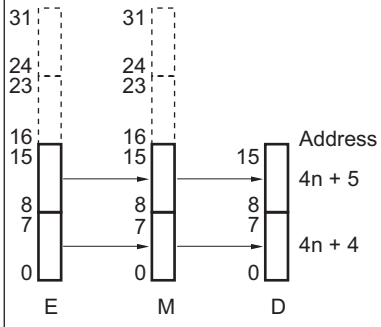
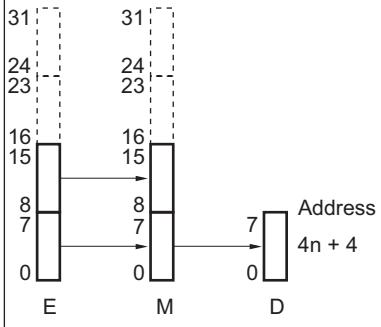
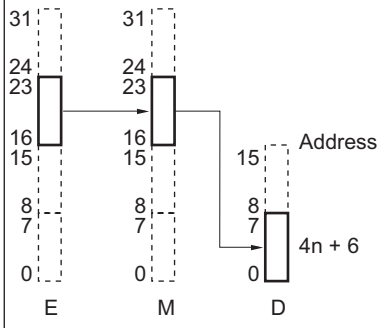
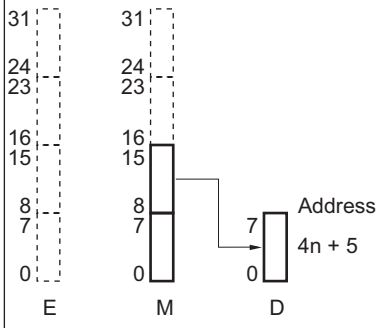
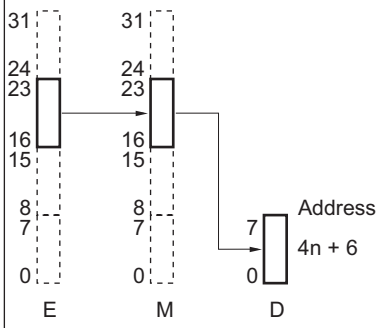
Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.30 Data Flow for Word Write Access (Little Endian) (3/4)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+2	1st		
	2nd		
	3rd	—	
	4th	—	

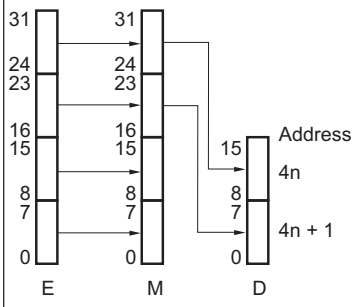
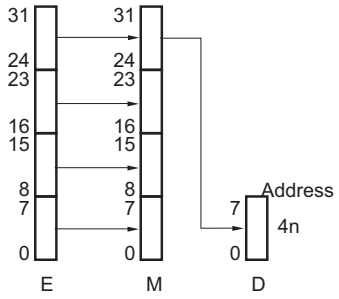
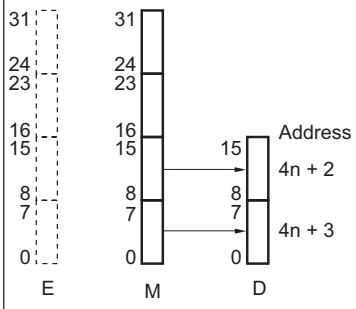
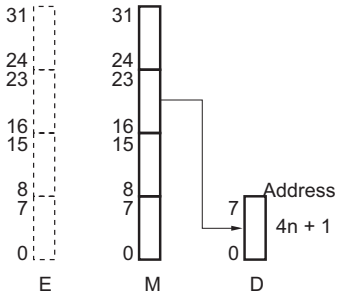
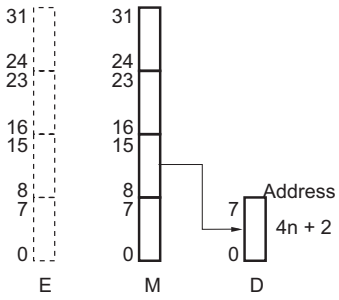
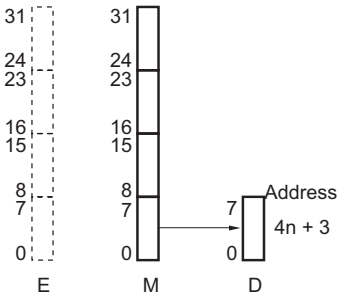
Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.30 Data Flow for Word Write Access (Little Endian) (4/4)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+3	1st		
	2nd		
	3rd		
	4th	———	

Note: E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Table 13.31 Data Flow for Word Write Access (Big Endian)

Address to be Accessed		Data Transfer Flow	
		External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st		
	2nd		
	3rd	———	
	4th	———	

Note 1. E: Internal bus
M: Memory controller data buffer
D: External data bus
n = 0, 1, 2, 3, ...

Note 2. Accessing the address starting with 4n + 1 or 4n + 3 is prohibited.

Section 14 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 all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the CSIG.

14.1 Features of RH850/F1L CSIG

14.1.1 Number of Units

This microcontroller has the following number of units of the CSIG.

Each CSIG unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 14.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	1	1	2	2
Name	CSIGn (n = 0)	CSIGn (n = 0)	CSIGn (n = 0)	CSIGn (n = 0)	CSIGn (n = 0, 1)	CSIGn (n = 0, 1)

Table 14.2 Index

Index	Meaning
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.

14.1.2 Register Base Address

CSIG base addresses are listed in the following table.

CSIG register addresses are given as offsets from the base addresses in general.

Table 14.3 Register Base Address

Base Address Name	Base Address
<CSIG0_base>	FFDB 0000_H
<CSIG1_base>	FFDB 2000_H

14.1.3 Clock Supply

The CSIG clock supply is shown in the following table.

Table 14.4 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
CSIGn	PCLK	CKSCLK_ICSI

14.1.4 Interrupt Request

CSIG interrupt requests are listed in the following table.

Table 14.5 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
CSIG0			
INTCSIGTIC	Communication status interrupt	19,110	8 (channels 0 to 7)
INTCSIGTIR	Receive status interrupt	20,111	9 (channels 0 to 7)
INTCSIGTIRE	Communication error interrupt	49	—
CSIG1			
INTCSIGTIC	Communication status interrupt	215	45 (channels 8 to 15)
INTCSIGTIR	Receive status interrupt	216	46 (channels 8 to 15)
INTCSIGTIRE	Communication error interrupt	217	—

14.1.5 Reset Sources

CSIG reset sources are listed in the following table. CSIG is initialized by these reset sources.

Table 14.6 Reset Sources

Unit Name	Reset Source
CSIGn	All reset sources (ISORES)

14.1.6 External Input/Output Signals

External input/output signals of CSIG are listed below.

Table 14.7 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
CSIG0		
CSIGTSCK	Serial clock signal	CSIG0SC
CSIGTSI	Serial data input signal	CSIG0SI
CSIGTSO	Serial data output signal	CSIG0SO
CSIGTSSO	Serial data output control signal	CSIG0SO
$\overline{\text{CSIGTSSI}}$	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
CSIGTSI	Serial data input signal	CSIG1SI
CSIGTSO	Serial data output signal	CSIG1SO
CSIGTSSO	Serial data output control signal	CSIG1SO
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG1SSI}}$
CSIGTRYI	Ready / busy input signal	CSIG1RYI
CSIGTRYO	Ready / busy output signal	CSIG1RYO

14.1.7 Data Consistency Check

The following table lists the port pins on which CSIGnSO pin functions are multiplexed and data consistency checking. See **Section 14.5.10, Error Detection** for details on data consistency checking.

Table 14.8 Support for Data Consistency Check

Unit Signal Name	Port Pin Name	Alternative Function
CSIG0		
CSIGTSO	P0_13	ALT_OUT4
	P10_6	ALT_OUT2
CSIG1		
CSIGTSO	P11_9	ALT_OUT1

14.2 Overview

14.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Selection of master mode and slave mode
- Slave select input signal ($\overline{\text{CSIGTSSI}}$) 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:
  - In master mode: 10.0 MHz (however, it must be up to PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be up to PCLK/6)
- Selectable clock phase and data phase
- Data transfer selectable from MSB or LSB first
- Transfer data length 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
- Error detection (data consistency check, parity, overrun) included
- Three different interrupt request signals (INTCSIGTIC, INTCSIGTIR, INTCSIGTIRE)
- LBM (Loop Back Mode) function for self test included

14.2.2 Functional Overview Description

The CSIG uses three signals for communication:

- Transmission clock CSIGTSCK
- Serial data output signal CSIGTSO
- Serial data input signal CSIGTSI

The CSIGnCTL2 register is used to select whether the CSIG should operated in master mode or slave mode.

Data transmission is bit-wise and serial and synchronous to the transmission clock.

The most important registers for setting up the CSIG are:

Register	Function
CSIGnCTL0	Provides and stops operating clock and enables/disables data transmission and data reception.
CSIGnCTL1	Controls options like 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.

14.2.3 Block Diagram

The block diagram shows the main components of the CSIG.

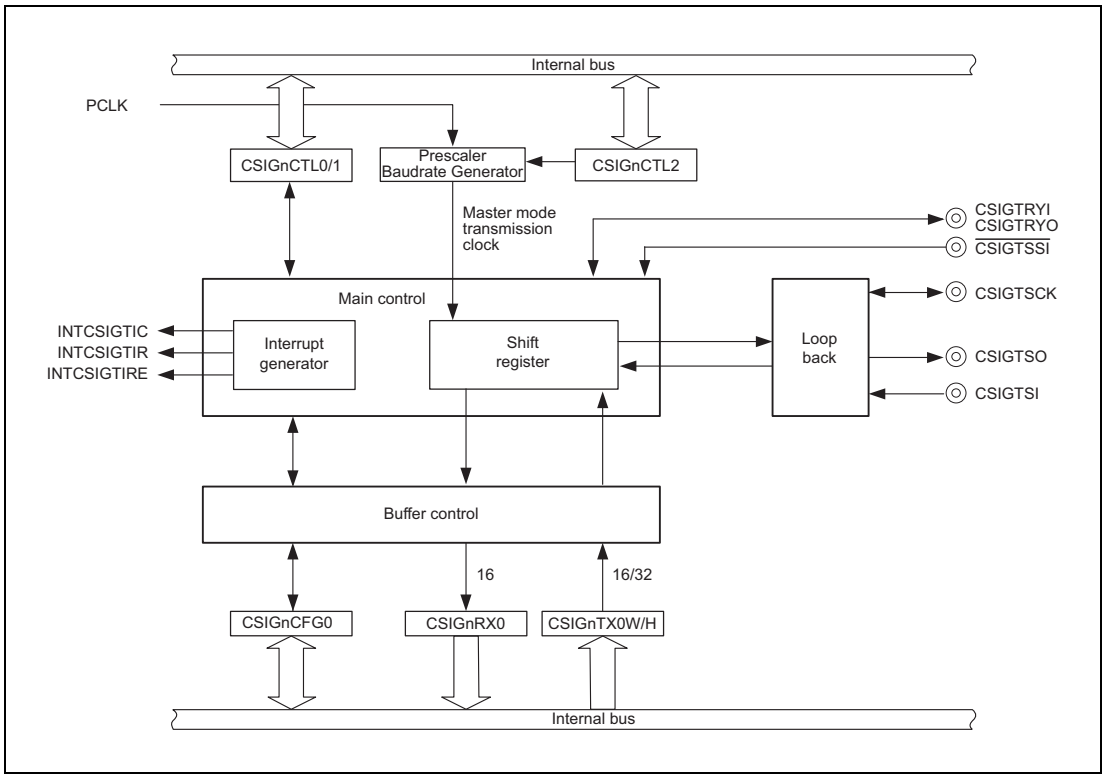


Figure 14.1 CSIG Block Diagram

In master mode, the transmission clock CSIGTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is received from an external source.

14.3 Registers

14.3.1 List of Registers

CSIG registers are listed in the following table.

For details on <CSIGn_base>, see **Section 14.1.2, Register Base Address**.

Table 14.9 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 Rx-only mode control register 0	CSIGnBCTL0	<CSIGn_base> + 1000_H
CSIGn	CSIGn configuration register 0	CSIGnCFG0	<CSIGn_base> + 1010_H
CSIGn	CSIGn transmission data register 0 for word access	CSIGnTX0W	<CSIGn_base> + 1004_H
CSIGn	CSIGn transmission data register 0 for half word access	CSIGnTX0H	<CSIGn_base> + 1008_H
CSIGn	CSIGn reception data register 0	CSIGnRX0	<CSIGn_base> + $100C_H$
CSIGn	CSIGn emulation register	CSIGnEMU	<CSIGn_base> + 0018_H

14.3.2 CSIGnCTL0 — CSIGn Control Register 0

This register controls the operation clock, and enables or disables transmission/reception.

Access: This register can be read/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 14.10 CSIGnCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIGnPWR	Controls operation clock. 0: Stops operation clock. 1: Provides operation clock. Clearing CSIGnPWR to 0 resets the internal circuits, stops operation, and sets the CSIG to standby state. No clock is provided to internal circuits. If CSIGnPWR is cleared during communication, ongoing communication is aborted. A restart of the communication setting is then required.
6	CSIGnTXE	Enables/disables transmission. 0: Transmission disabled 1: Transmission enabled
5	CSIGnRXE	Enables/disables reception. 0: Receive disabled 1: Receive enabled
4 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	CSIGnMBS	Be sure to set 1 to this bit (the value after reset is 0).

CAUTION

When setting this register, see **Table 14.23, List of Cautions when Setting Registers**.

14.3.3 CSIGnCTL1 — CSIGn Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It enables/disables extended data length control, data consistency check, loop-back mode, handshake function, and slave select function.

Access: This register can be read/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 14.11 CSIGnCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing to these bits, 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 14.3.8, 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 free for next data. For details, see **14.4.2, INTCSIGTIC (Communication Status Interrupt)**
15 to 8	Reserved	When read, the value after reset is returned. When writing to these bits, 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 **14.5.5.2, Data Length Selection with Extended Data Length**.
6	Reserved	When read, the value after reset is returned. When writing to this bit, 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 **14.5.10.1, Data Consistency Check**.
4	Reserved	When read, the value after reset is returned. When writing to this bit, 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 14.5.9, Loop-Back Mode**.

Table 14.11 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 14.4.1, Interrupt Delay**.
1	CSIGnHSE	Enables/disables handshake mode. 0: Handshake function disabled 1: Handshake function enabled For details, see **Section 14.5.8, Handshake Function**.
0	CSIGnSSE	Enables/disables slave select function. 0: Input signal $\overline{\text{CSIGTSSI}}$ is ignored. 1: Input signal $\overline{\text{CSIGTSSI}}$ is enabled. If the slave select function is not used, this bit must be set to 0 (see also **Section 14.5.2, Master/Slave Connections**).

Details about CSIGnCTL1.CSIGnSSE:

Table 14.12 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 14.13 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 14.23, List of Cautions when Setting Registers**.

14.3.4 CSIGnCTL2 — CSIGn Control Register 2

This register selects the communication clock.

Access: This register can be read/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 14.14 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 to this bit, 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 14.23, List of Cautions when Setting Registers**.

14.3.5 CSIGnSTR0 — CSIGn Status Register 0

This register indicates the status of the CSIG.

Access: This register can only 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 14.15 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: Transmission is in progress or being prepared Setting and clearing of this bit is as follows:
6 to 4	Reserved	When read, the value after reset is returned.
3	CSIGnDCE	Data Consistency Check Error Flag 0: No data consistency error detected 1: Data consistency 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, clearing to 0 by CSIGnSTCR0.CSIGnDCEC takes precedence over setting to 1. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 1 to 0 or from 0 to 1.
2	Reserved	When read, the value after reset is returned.

Master Mode	Set by	Cleared by
Tx-only mode	Writing to transmit register	Within a half clock cycle from the last serial clock edge
Tx/Rx mode		
Rx-only mode	Reading from receive register	

Slave Mode	Set by	Cleared by
Tx-only mode	Writing to transmit register	Within a half clock cycle from the last serial clock edge
Tx/Rx mode		
Rx-only mode	CSIGTSCK input	

Table 14.15 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, clearing to 0 by CSIGnSTCR0.CSIGnPEC takes precedence over setting to 1. This bit is initialized by the value of 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, clearing to 0 by CSIGnSTCR0.CSIGnOVEC takes precedence over setting to 1. 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 14.23, List of Cautions when Setting Registers**.

14.3.6 CSIGnSTCR0 — CSIGn Status Clear Register 0

This register clears the status flags of the CSIGnSTR0 status register.

Access: This register can be read/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 14.16 CSIGnSTCR0 Register Contents

Bit Position	Bit Name	Function
15 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	CSIGnDCEC	Controls the data consistency error flag clear command. 0: No operation. Read value is always 0. 1: Clear data consistency check error flag (CSIGnSTR0.CSIGnDCE).
2	Reserved	When read, the value after reset is returned. When writing to this bit, 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).

14.3.7 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/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 14.17 CSIGnBCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	CSIGnSCE	Disables/enables next data reception start by reading CSIGnRX0. 0: Next reception disabled 1: Next reception enabled For details, see **14.5.4.2, Receive-Only Mode** and **Section 14.5.7, Communication in Slave Mode**.

CAUTION

When setting this register, see **Table 14.23, List of Cautions when Setting Registers**.

14.3.8 CSIGnCFG0 — CSIGn Configuration Register 0

This register configures the communication protocol – data length, parity, transfer direction, clock phase, and data phase.

Access: This register can be read/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 14.18 CSIGnCFG0 Register Contents (1/2)

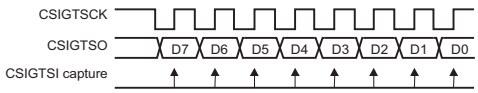
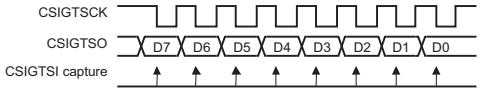
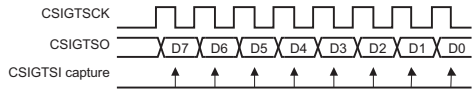
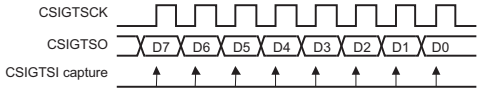
Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
29, 28	CSIGnPS[1:0]	Specifies parity.

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.

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] for a 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 to these bits, write the value after reset.
18	CSIGnDIR	Selects the serial data direction. 0: Data is sent/received with MSB first 1: Data is sent/received with LSB first
17	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.

Table 14.18 CSIGnCFG0 Register Contents (2/2)

Bit Position	Bit Name	Function
16	CSIGnDAP	Data Phase Selection Select data phase in line with CSIGnCTL1.CSIGnCKR bit. See below for clock and data phase. .
15 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

CSIGnCTL1.CSIGnCKR	CSIGnDAP	Clock and Data Phase Selection
0	0	
0	1	
1	0	
1	1	

CAUTION

When setting this register, see **Table 14.23, List of Cautions when Setting Registers**.

14.3.9 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/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 14.19 CSIGnTX0W Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, write the value after reset.
15 to 0	CSIGnTX[15:0]	Data to be transmitted

CAUTION

When setting this register, see **Table 14.23, List of Cautions when Setting Registers**.

14.3.10 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 16 high-order bits of CSIGnTX0W are applied for transfer.

Access: This register can be read/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 14.20 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 14.23, List of Cautions when Setting Registers**.

14.3.11 CSIGnRX0 — CSIGn Reception Register 0

This register stores the received data.

Access: This register can only 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 14.21 CSIGnRX0 Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIGnRX [15:0]	Received Data This register is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0. When reading, the values of the register must be read at least 1 clock before the generation of CSIGTIR interrupt.

CAUTION

When setting this register, see **Table 14.23, List of Cautions when Setting Registers**.

14.3.12 CSIGnEMU — CSIGn Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read/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 14.22 CSIGnEMU Register Contents

Bit Position	Bit Name	Function
7	CSIGnSVSDIS	Selects 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 to these bits, write the value after reset.

CAUTION

When setting this register, see **Table 14.23, List of Cautions when Setting Registers**.

14.3.13 List of Cautions

Table 14.23 List of Cautions when Setting Registers

Register Name	Bit Name	Contents
CSIGnCTL0	CSIGnPWR	If this bit is cleared during communication, ongoing communication is suspended. After the communication is suspended, 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 at the same time 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 suspended.
CSIGnCTL0	CSIGnMBS	When writing, always write 0. (The value after reset is “0”.) This bit must be modified at the same time 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 this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting of the max transfer clock frequency is as follows. • Master mode: 10.0 MHz (however, it must be up to PCLK/4) • Slave mode: 5.0 MHz (however, it must be up to 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 at least one clock before the generation of the CSIGnTIR interrupt. Fix the CSIGnSCE bit to 0 when transfer is in transmission or 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		This bit is initialized when CSIGnRX0 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.

14.4 Interrupt Sources

CSIG can generate the following interrupts:

- INTCSIGTIC (communication status interrupt)
- INTCSIGTIR (reception status interrupt)
- INTCSIGTIRE (communication error interrupt)

14.4.1 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by one half period of the transmission clock CSIGTSCK. This is not possible in slave mode.

The delay is specified by setting bit 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), CSIGnCFG0.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits).

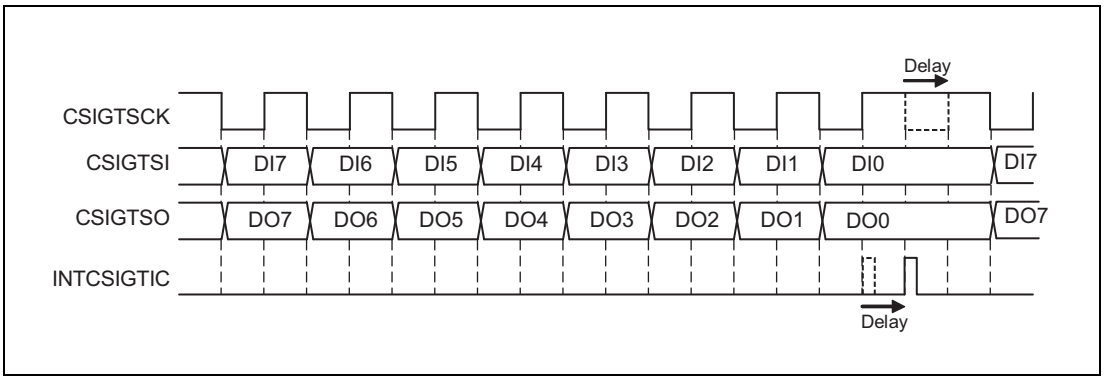


Figure 14.2 Interrupt Delay Function (CSIGnCTL1.CSIGnSIT = 1)

14.4.2 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), CSIGnCFG0.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).

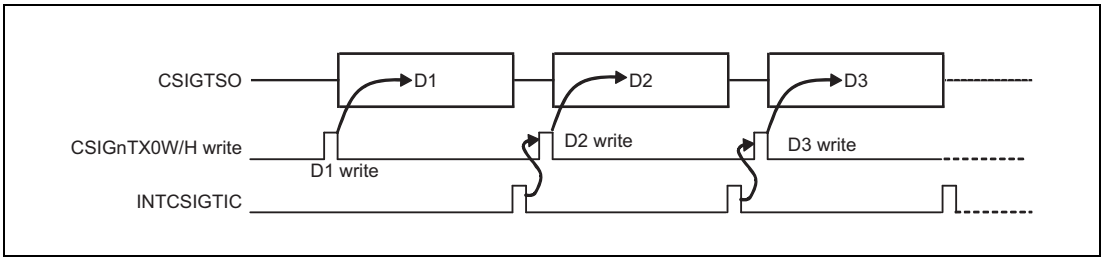


Figure 14.3 Generation of INTCSIGTIC after Communication (CSIGnCTL1.CSIGnSLIT = 0)

However, INTCSIGTIC can also be set up to occur when the CSIGnTX0W/H register is free for 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.

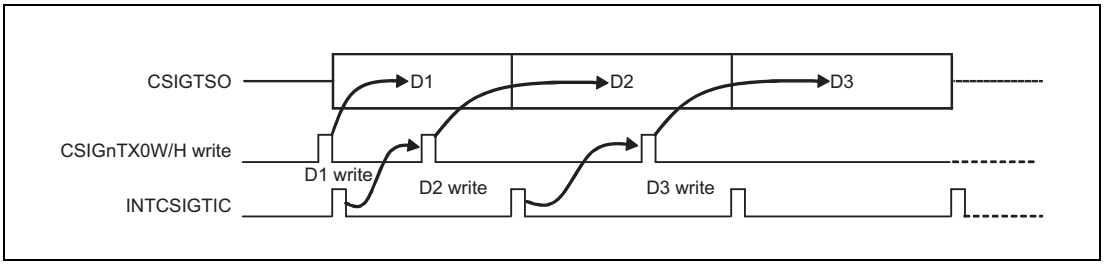


Figure 14.4 Generation of INTCSIGTIC at the Beginning of Communication

14.4.3 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 register CSIGnRX0.

The following example assumes master mode and a setting of CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay), CSIGnCFG0.CSIGnCKP = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits).

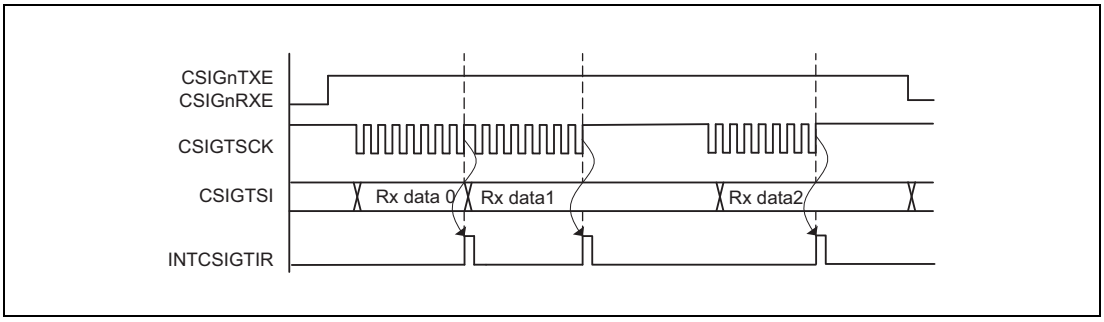


Figure 14.5 Generation of INTCSIGTIR

14.4.4 INTCSIGTIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

Table 14.24 Data Error Types

Error Type	Communication Status After Error Interrupt
Parity error	Interrupt is generated and communication continues.
Data consistency error	Interrupt is generated and communication continues.
Overrun error	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. Interrupt is not generated and overrun errors do 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 14.5.10, Error Detection**.

14.5 Operation

14.5.1 Master/Slave Mode

CSIG operation 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 too.

14.5.1.1 Master Mode

In master mode, the serial communication clock is generated by the internal baud rate generator (BRG) and provided by signal CSIGTSCK.

Master mode is enabled by setting bits CSIGnCTL2.CSIGnPRS[2:0] to anything but 111_B. In master mode, the frequency setting of the BRG becomes effective by setting bits CSIGnCTL2.CSIGnPRS[2:0] and bits CSIGnCTL2.CSIGnBRS[11:0].

The default level of CSIGTSCK depends on the clock phase selection bit: it is high when CSIGnCFG0.CSIGnCKR = 0, and is low when CSIGnCFG0.CSIGnCKR = 1.

The example below shows the communication in master mode for 8 data bits, CSIGnCFG0.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

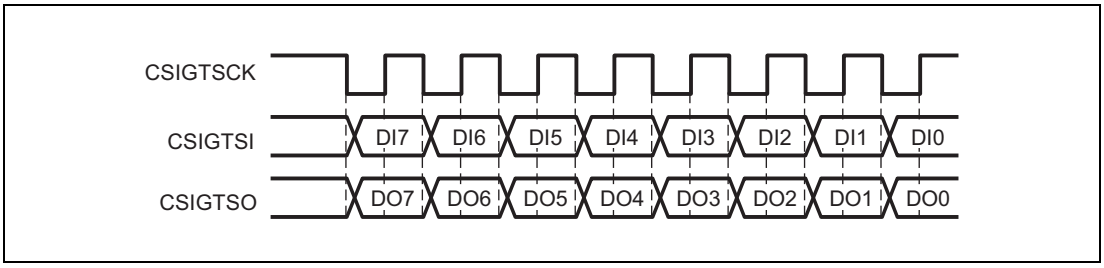


Figure 14.6 Transmission/Reception in Master Mode

14.5.1.2 Slave Mode

In slave mode, another device is the communication master. The external clock is received by signal CSIGTSCK. Send/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. When the BRG is disabled, CSIGTSCK stays on the level specified by CSIGnCTL1.CSIGnCKR.

The example below shows the communication in slave mode for 8 data bits, CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

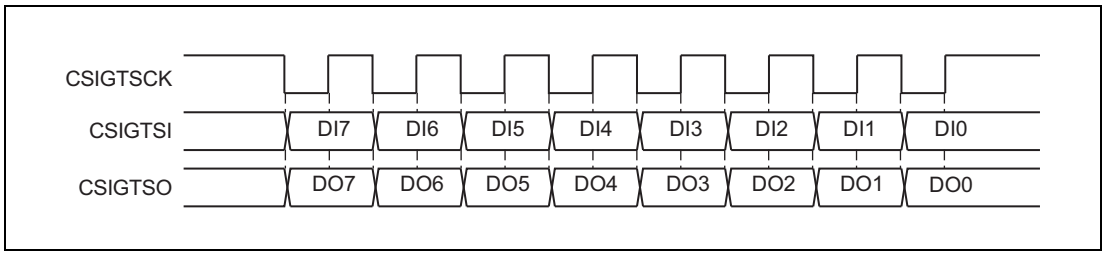


Figure 14.7 Transmission/Reception in Slave Mode

14.5.2 Master/Slave Connections

14.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

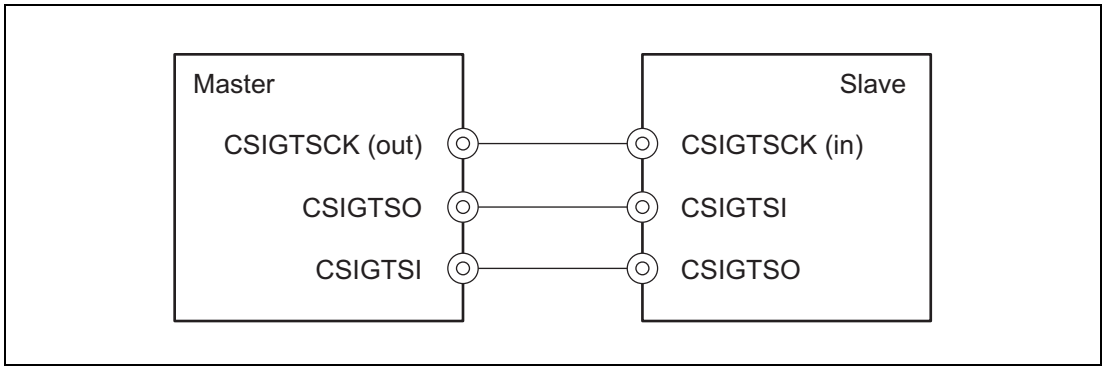


Figure 14.8 Direct Master/Slave Connection

14.5.2.2 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/disabled by bit CSIGnCTL1.CSIGnSSE.

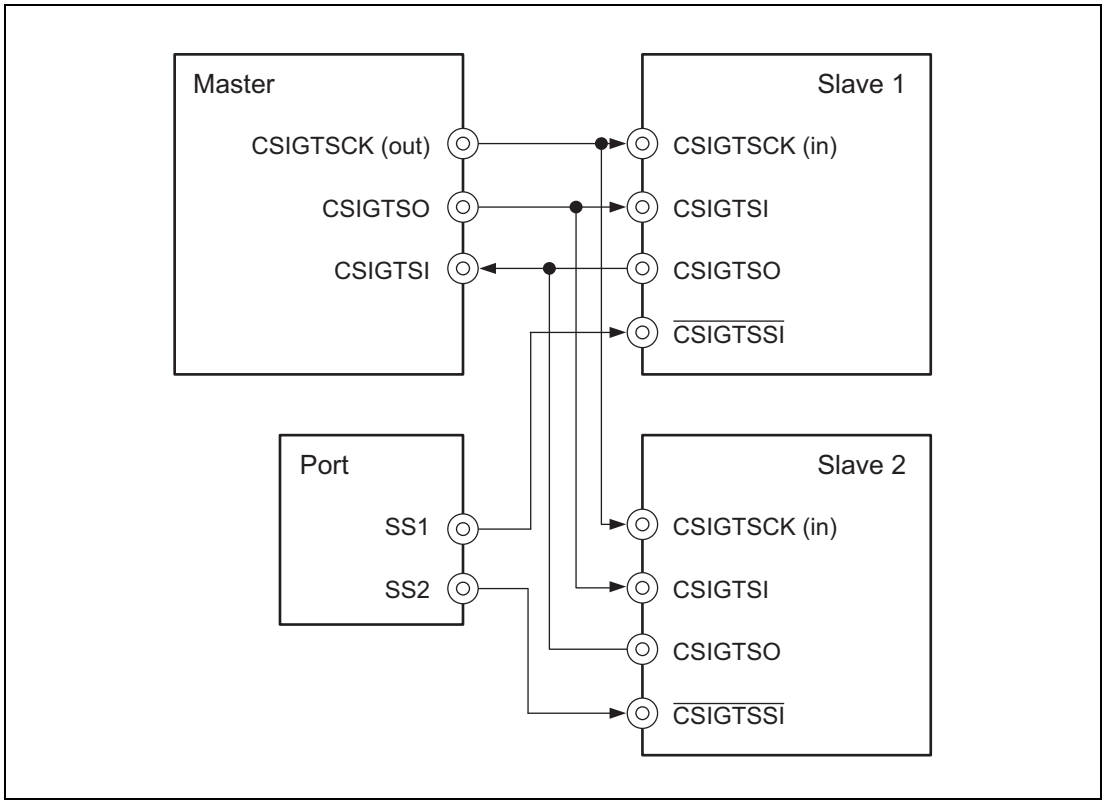


Figure 14.9 Master to Multiple Slaves Connection

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, the CSIGTSO output buffer is disabled and set to input mode in order to avoid interference with the output of another slave which was selected.

14.5.3 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.

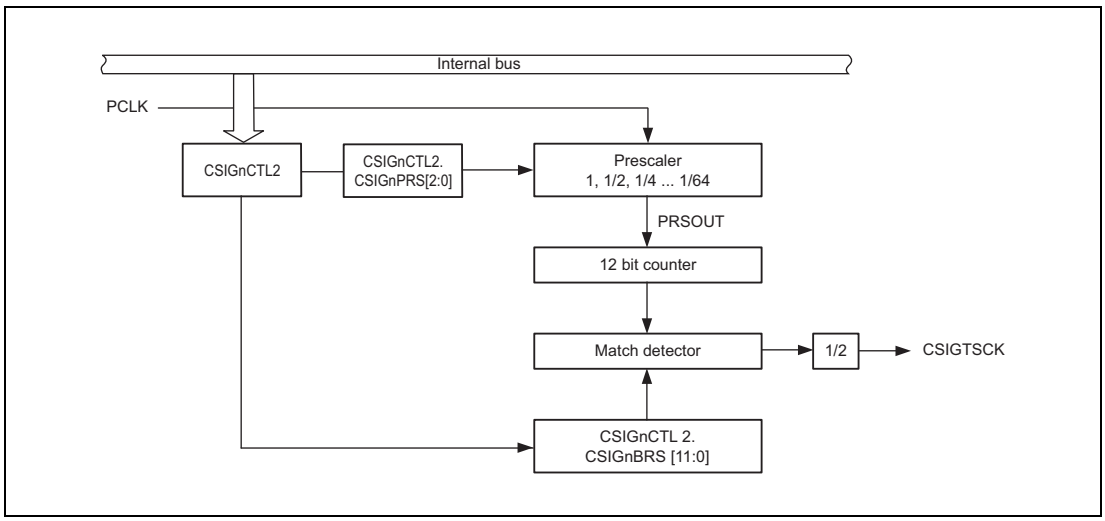


Figure 14.10 BRG Block Diagram

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

q = CSIGnCTL2.CSIGnPRS[2:0] = 0 to 6

k = CSIGnCTL2.CSIGnBRS[11:0] = 1 to 4095

Transfer clock frequency upper and lower limits

When setting the transfer clock frequency, please note the followings.

- For the maximum transfer clock frequency of this product in master mode or slave mode, see the CSIG timing shown in the electrical characteristics. In addition, in either mode, set it within the specified 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 up to PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be up to PCLK/6)

14.5.4 Data Transfer Modes

14.5.4.1 Transmit-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 0 puts the CSIG in transmit-only mode. Transmission starts when transmit data 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.

14.5.4.2 Receive-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 0 and CSIGnCTL0.CSIGnRXE = 1 puts the CSIG in receive-only mode.

In master mode, reception starts when dummy data is read from the CSIGnRX0 register.
All following receptions are triggered by a read from the receive data register CSIGnRX0, as long as CSIGnBCTL0.CSIGnSCE = 1.

In slave mode, reception starts when the communication clock CSIGTSCK from the master is received. In this case, it is not necessary to write data to the CSIGnTX0W or CSIGnTX0H register of the slave.

NOTE

In receive-only mode, any previously received data must be read from the reception register CSIGnRX0 in order to avoid any overwrite situation.

Moreover the communication start bit CSIGnBCTL0.CSIGnSCE has to be set to 1 and has to set back 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 of further data receptions continue at step 3, continue to read it until all data has been received.
   Before reading the last received data from CSIGnRX0, set CSIGnBCTL0.CSIGnSCE = 0.

14.5.4.3 Transmit/Receive Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 1 puts the CSIG in transmit/receive mode.

Data transfer (transmission and reception) starts when transmit data is written to the CSIGnTX0W or CSIGnTX0H register.

14.5.5 Data Length Selection

14.5.5.1 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):

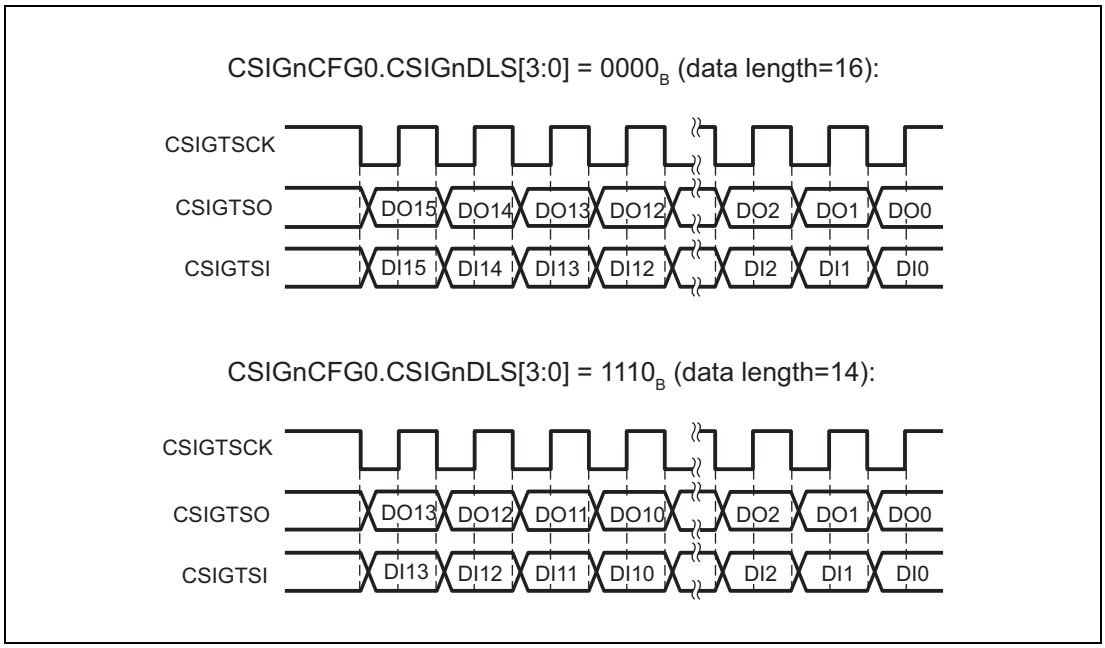


Figure 14.11 Data Length Select Function

14.5.5.2 Data Length Selection with Extended Data Length

If the data to be sent/received exceeds 16 bits, the extended data length (EDL) feature can be used.

The EDL function is enabled by setting bit CSIGnCTL1.CSIGnEDLE to 1.

The following describes how the EDL function works and how to specify the 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 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 is complete after the data with the specified data length (the remainder with CSIGnTX0W.CSIGnEDL = 0) has been sent.

Example

Example for sending 40-bit data, for example the data $123456789A_H$:

40 bits are split into 2 × 16 bits plus 8 bits.

- Initialize CSIGnCFG0.CSIGnDLS[3:0] = 8_D.
- To send 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.

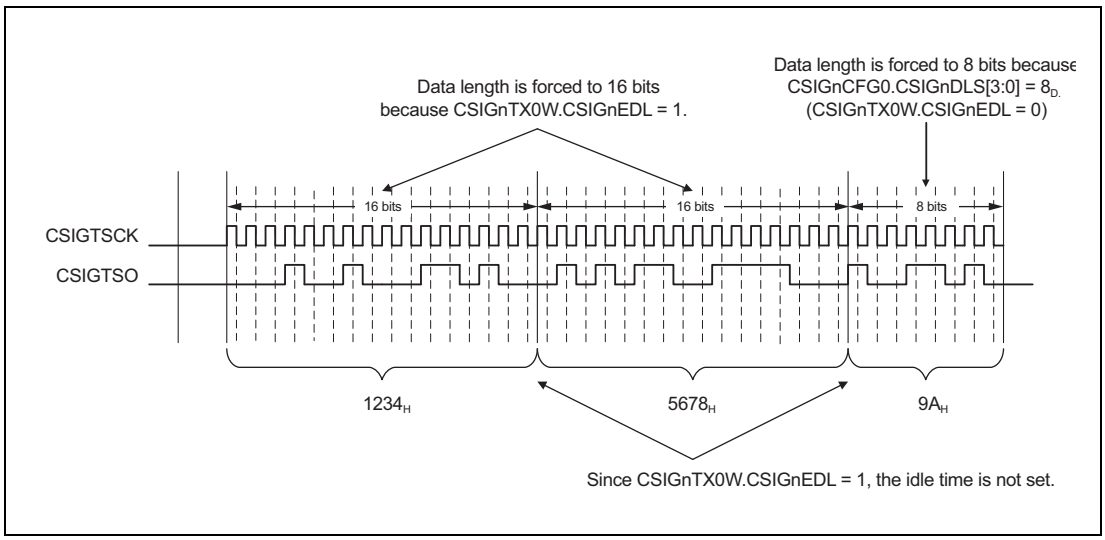


Figure 14.12 EDL Timing Diagram

NOTES

1. Data length with less than 7 bits can be set only when EDL mode is used.
2. It is not possible to send 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. Explanation of data direction is provided in the following example.
   - 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 exclusive for receive.
   (CSIGnCTL2.CSIGnPRS[2:0] = 111B, CSIGnCTL0.CSIGnTXE = 0, CSIGnCTL0.CSIGnRXE = 1)

14.5.6 Serial Data Direction Select 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):

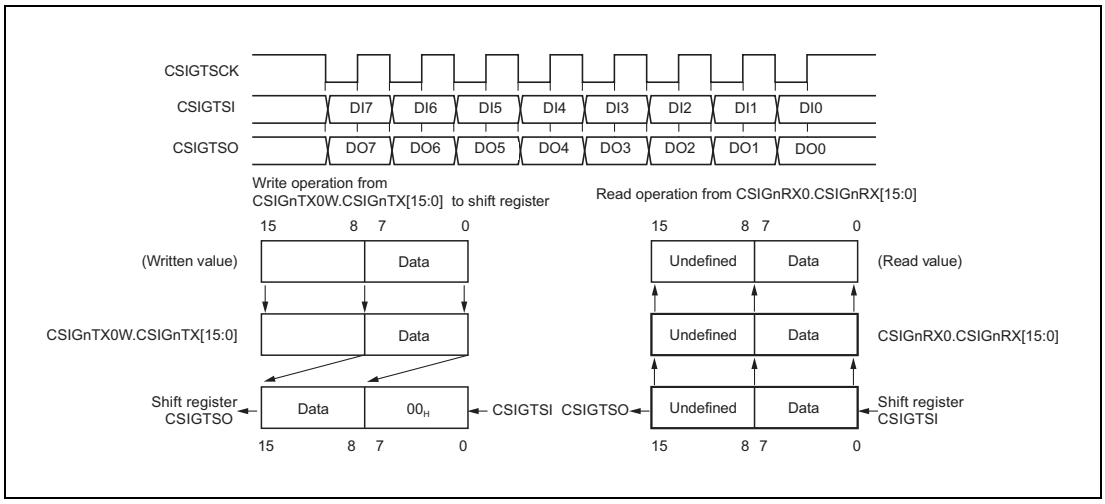


Figure 14.13 Serial Data Direction Select Function — MSB First (CSIGnDIR = 0)

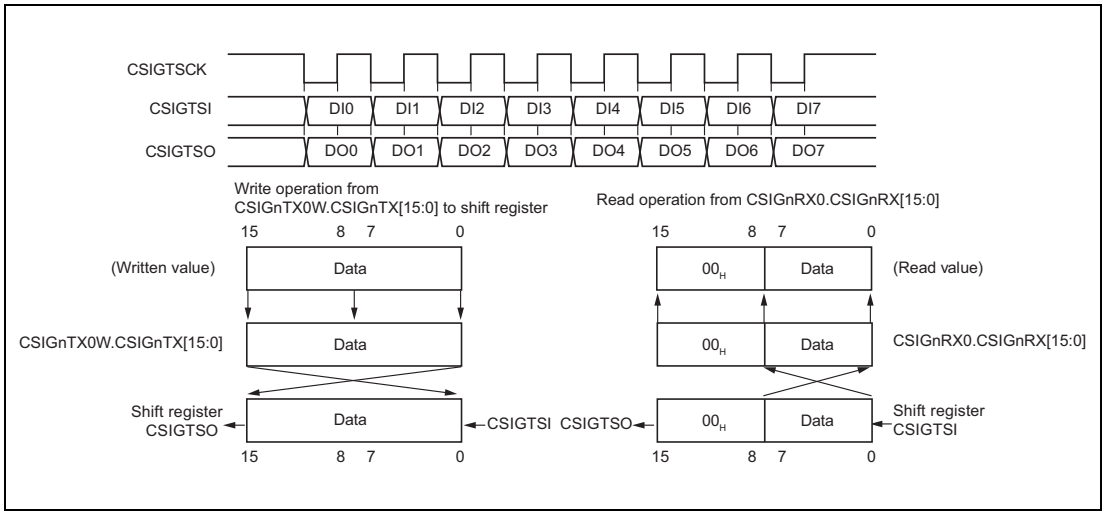


Figure 14.14 Serial Data Direction Select Function — LSB First (CSIGnDIR = 1)

14.5.7 Communication in Slave Mode

The following figure illustrates the communication signals and timings in slave mode.

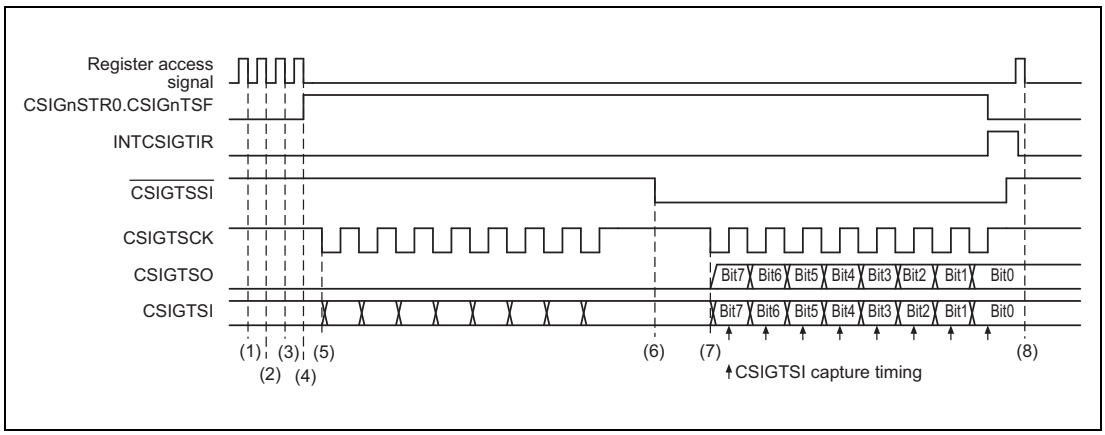


Figure 14.15 Rx/Tx Communication Timing in Slave Mode

1. CSIG is put into slave mode by setting CSIGnCTL2.CSIGnPRS[2:0] = 111_B. The $\overline{\text{CSIGTSSI}}$ signal is enabled (CSIGnCTL1.CSIGnSSE = 1) and the clock phase is at the high level (CSIGnCTL1.CSIGnCKR = 1).
2. Data length is 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B).
   Data direction is 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 “transmission in progress” flag CSIGnSTR0.CSIGnTSF is automatically set and the CSIG waits until signal $\overline{\text{CSIGTSSI}}$ goes low.
5. As long as signal $\overline{\text{CSIGTSSI}}$ is high, transmission/reception is not started even if the serial clock in input. CSIGTSO retains the values and input at CSIGTSI is ignored.
6. As soon as $\overline{\text{CSIGTSSI}}$ falls to low level, the chip serial data output is enabled and ready for transmission.
7. If the serial clock is input to the CSIG while $\overline{\text{CSIGTSSI}}$ is low, data is sent to CSIGTSO in synchronization with the serial clock, and at the same time, data is received from CSIGTSI.
8. The register CSIGnRX0 is read.

14.5.8 Handshake Function

CSIG features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by bit CSIGnCTL1.CSIGnHSE. For handshake, the signals CSIGTRYI and CSIGTRYO are used.

The timing depends on the data phase selection bit CSIGnCFG0.CSIGnDAP.

14.5.8.1 Slave Mode

When CSIGnCTL1.CSIGnHSE = 1 and the slave is busy, a low-level CSIGTSHSG signal is output. This happens when previous receive data is still in the CSIGnRX0 register, and new data cannot be copied from the shift register to CSIGnRX0 (CSIGnRX0 full condition).

The following examples assume a data length of 8 bits.

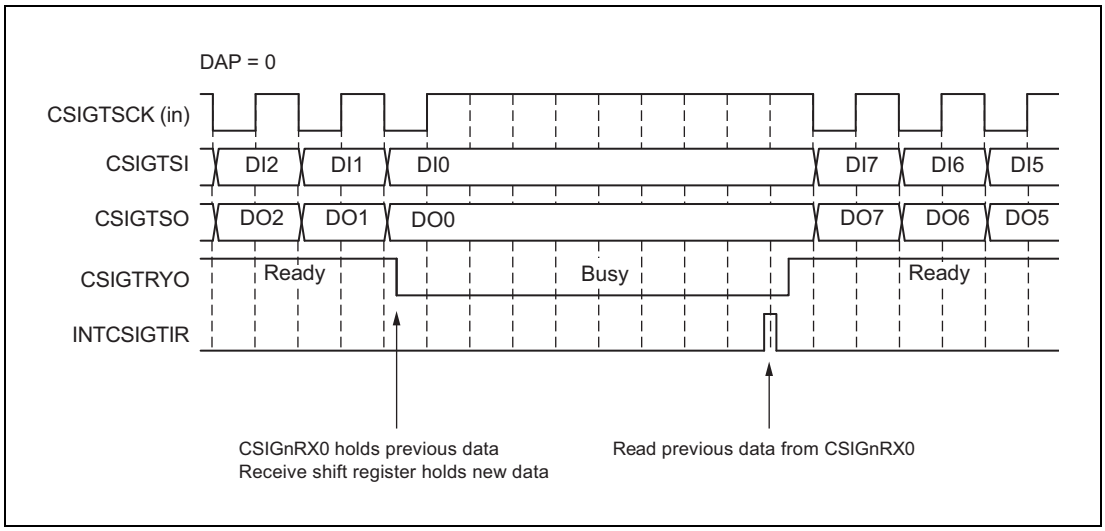


Figure 14.16 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 0)

As long as 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 reception register CSIGnRX0 has been read.

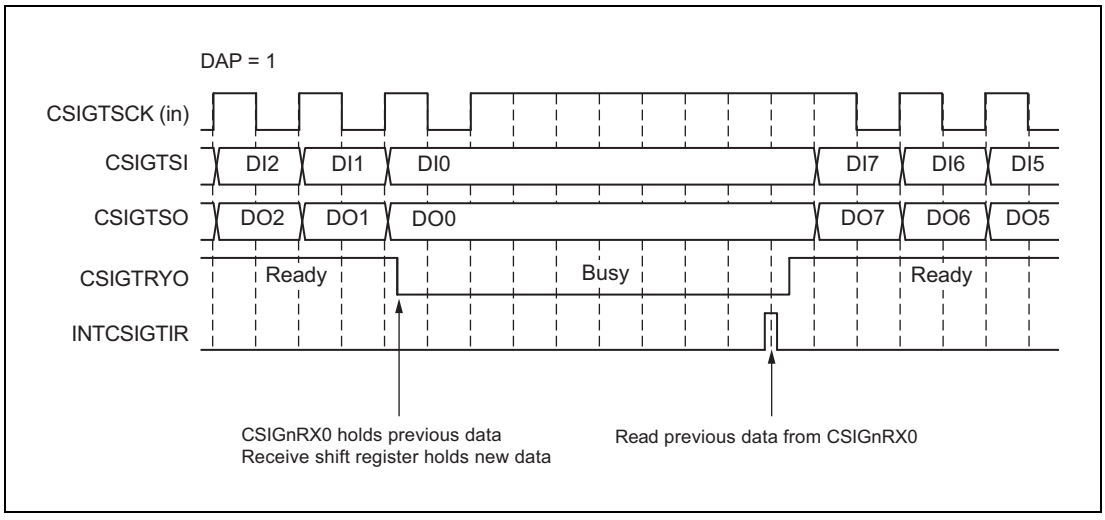


Figure 14.17 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 1)

14.5.8.2 Master Mode

When the master detects low level of the CSIGTRYI, the following transfer is put on hold, and the master goes into wait status. It suspends the CSIGTSCK clock.

The CSIGTRYI level is checked at each half clock cycle of CSIGTSCK.

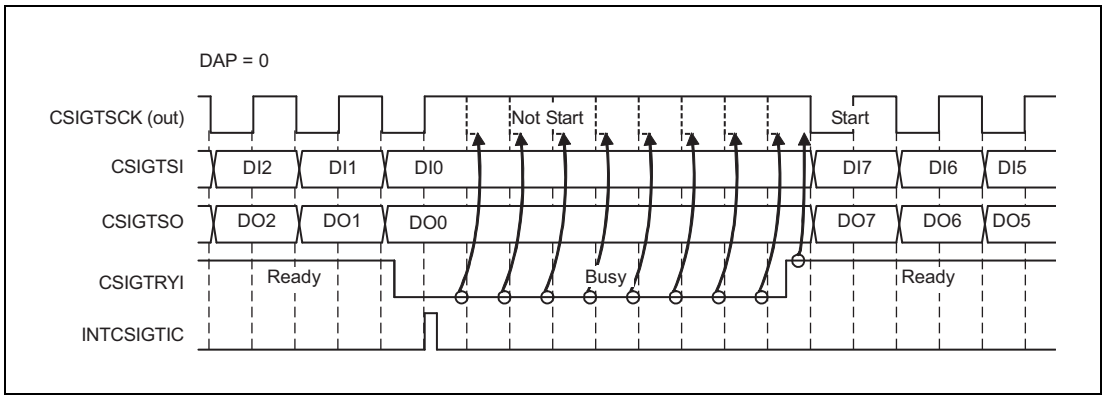


Figure 14.18 Master's Reaction to CSIGTRYI (CSIGnCFG0.CSIGnDAP = 0)

If the CSIGTRYI low signal comes from the slave 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").

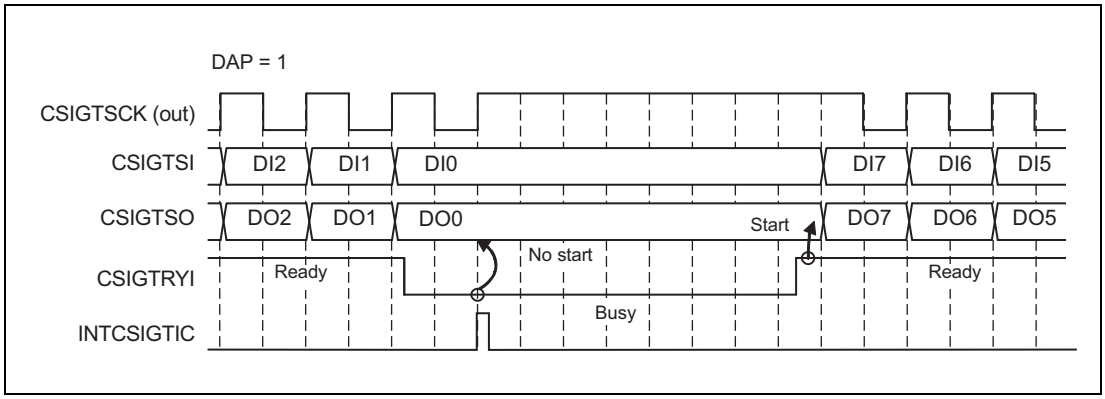


Figure 14.19 Master's Reaction to CSIGTRYI (CSIGnCFG0.CSIGnDAP = 1)

CAUTION

If multiple slaves are connected, the master must only detect the CSIGTRYI signal of the slave it has selected for communication.

CSIGTRYI 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 is finished.

14.5.9 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 (CSIG0CTL1.CSIG0LBM = 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 in loop-back mode and carry out normal transfer operations. Then check that the received data is the same as the transmitted data.

Table 14.25 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)

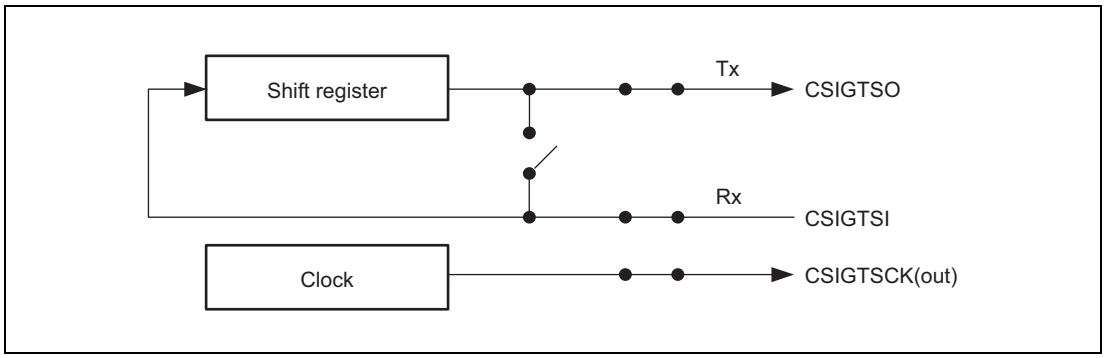


Figure 14.20 Normal Operation

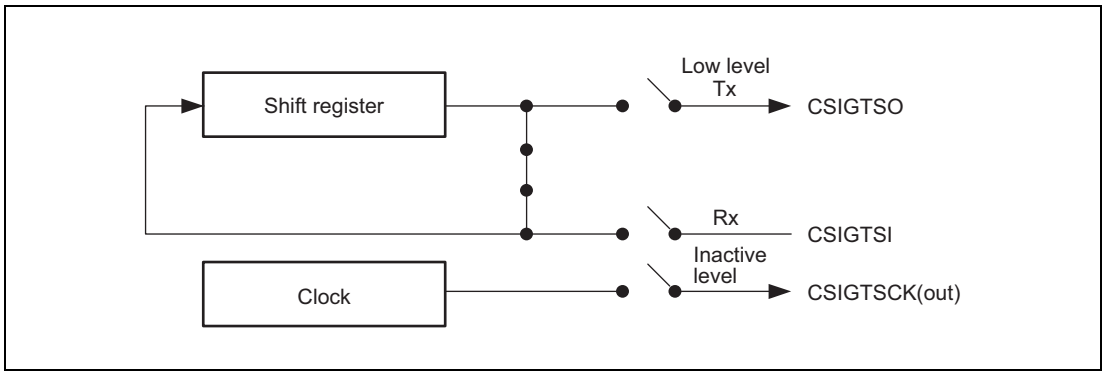


Figure 14.21 Operation in Loop-Back Mode

14.5.10 Error Detection

CSIG can detect three error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error

Data consistency error and parity error check functions are independently enabled or disabled.

If one of these errors is detected, the interrupt INTCSIGTIRE is generated.

14.5.10.1 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 bit CSIGnCTL1.CSIGnDCS (When checking data consistency, make sure that PIPCn.PIPCn_m = 1.). It is not active if data transmission is disabled (CSIGnCTL0.CSIGnTXE = 0).

When the data consistency check is active, the data transferred from CSIGnTX0W or CSIGnTX0H to the shift register is copied to a separate register. In addition, the physical levels at CSIGTSO are capture and the logical interpretation is written to an own 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:

- Interrupt INTCSIGTIRE is generated.
- Bit CSIGnSTR0.CSIGnDCE is set.

The function is illustrated in the following block diagram.

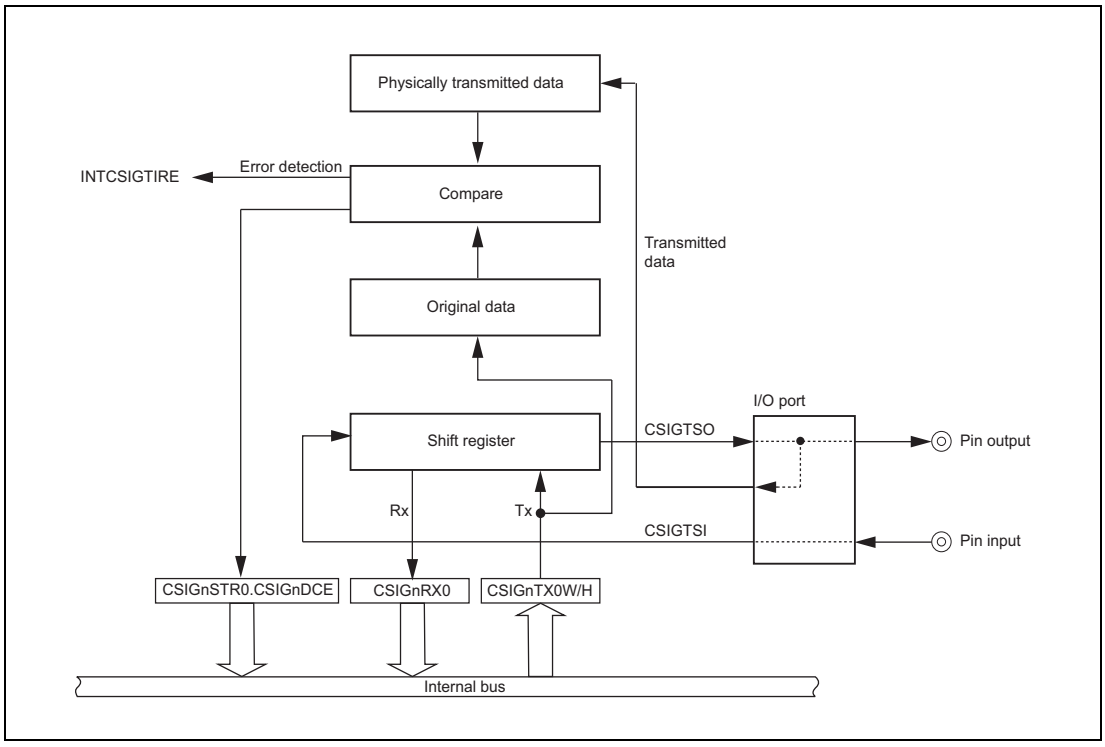


Figure 14.22 Functional Block Diagram of the Data Consistency Check

14.5.10.2 Parity Check

Parity is a common mean to detect a single bit failure during data transmission. CSIG 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 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.
- Bit CSIGnSTR0.CSIGnPE 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.

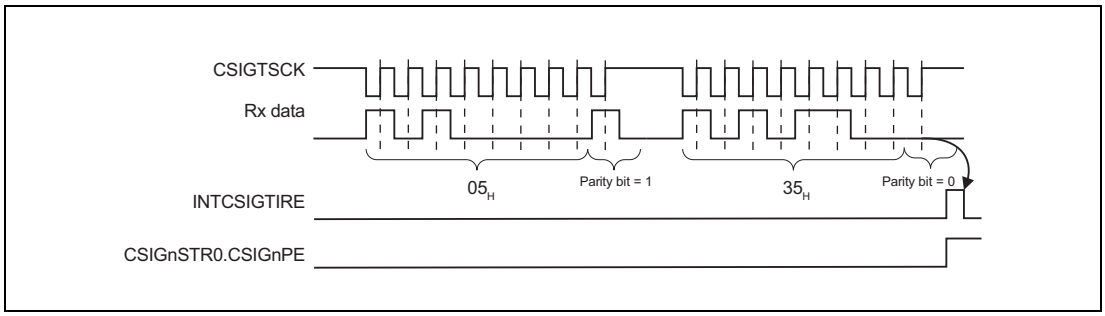


Figure 14.23 Parity Check Example

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.

14.5.10.3 Overrun Error

This error occurs when previously received data still resides in the reception register CSIGnRX0, because it wasn't 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
- Bit CSIGnSTR0.CSIGnOVE is set
- Received data is overwritten and communication continues.

The following figure illustrates the function.

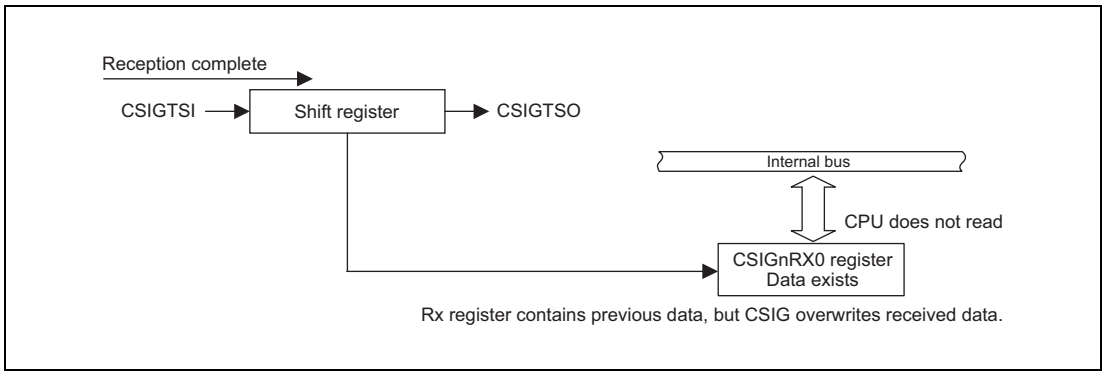


Figure 14.24 Overrun Error Detection

The following figure illustrates an example where:

- Rx data 3 was not read
- Rx data 4 was received, and data is overwritten.

Thus an overrun error occurs.

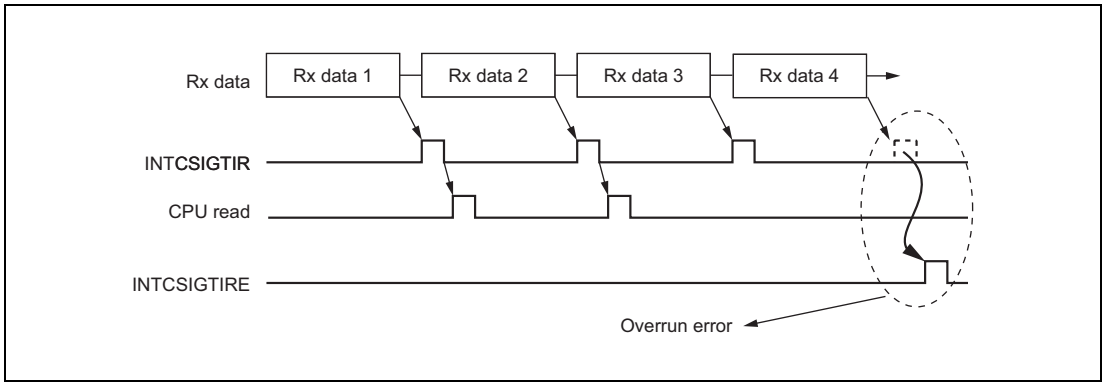


Figure 14.25 Overrun Error Detection - Example

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 has read its reception register and is ready again.

For details see **Section 14.5.8, Handshake Function**.

14.6 Operating Procedure

14.6.1 Master Mode Transmission/Reception by DMA

In the following a transmit/receive example in master mode in combination with a DMA is described.

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 at the end of the transfer (CSIGnCTL1.CSIGnSLIT = 0)
- Normal clock and data phase (CSIGnCFG0.CSIGnCKP = 0, CSIGnCFG0.CSIGnDAP = 0)
- The number of data is 10 (0 to 9)

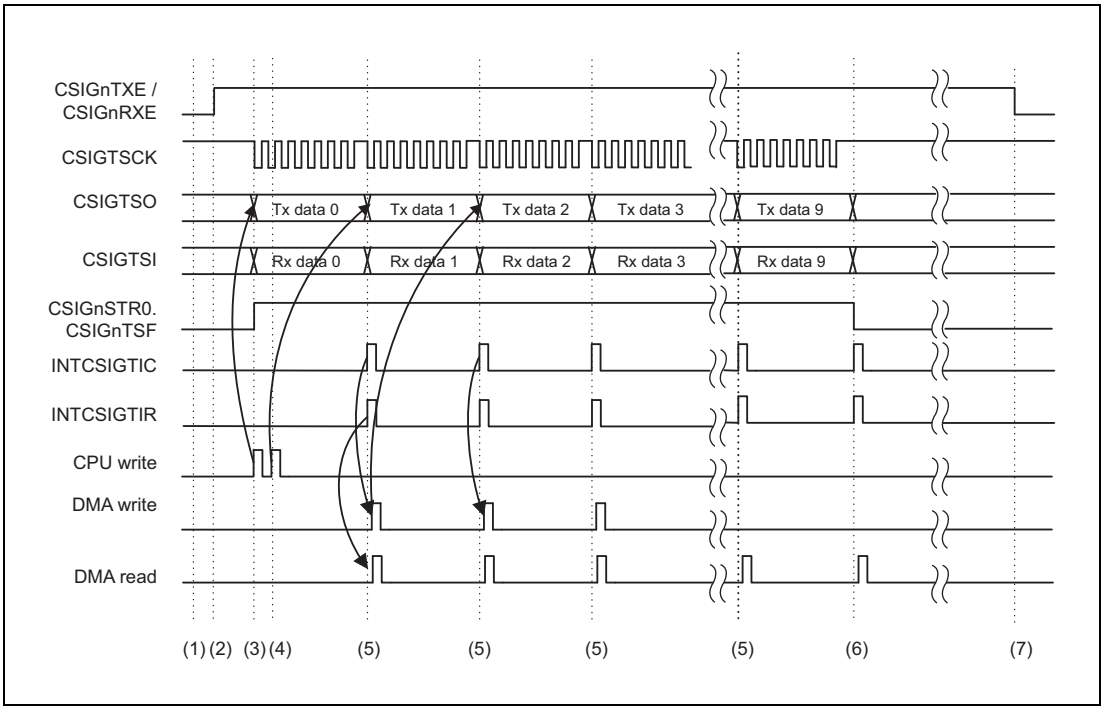


Figure 14.26 Communication in Master Mode

Procedure:

1. Configure the communication protocol in register CSIGnCFG0. Interrupt timing and operation mode are specified by setting the corresponding bits of the CSIGnCTL1 register and CSIGnCTL2 register.
2. In the CSIGnCTL0 register, set bits CSIGnPWR = 1 (enable the clock), CSIGnTXE = 1 (enable transmission), CSIGnRXE = 1 (enable reception).
3. Write the first data to be sent to the transmission register CSIGnTX0H. Transmission starts automatically when the first data is 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 every data that has been transmitted or received, the interrupts INTCSIGTIC and INTCSIGTIR are generated. INTCSIGTIC indicates that the next data can be written to

CSIGnTX0H. INTCSIGTIC indicates that the reception register CSIGnRX0 must be read.
In this example, CPU write and DMA write are equivalent.

6. No more write action is required after completion of writing data 8. Data 9 (the last data) has been written in advance.
However, the reception register CSIGnRX0 must be read after completion of data 8 and 9.

7. To finally 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 15 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 all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of CSIH.

15.1 Features of RH850/F1L CSIH

15.1.1 Number of Units

This microcontroller has the following number of CSIH units.

Each CSIH unit has one channel interface.

Table 15.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of units	1	1	3	4	4	4
Name	CSIHn (n = 0)	CSIHn (n = 0)	CSIHn (n = 0 to 2)	CSIHn (n = 0 to 3)	CSIHn (n = 0 to 3)	CSIHn (n = 0 to 3)

Table 15.2 Index

Index	Meaning
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 up to 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 a non-specified baud rate setting register of CSIHn.

The following table shows values indicated by the indexes of each product.

Table 15.3 Indexes of Products

Indexes of Each Product				
48 pins	**64 pins**	**80 pins**	**100 pins**	**144 pins, 176 pins**
For the value of x, see **Table 15.4, Number of Chip Select Signals**.				
y = 0 to 3	y = 0 to 3	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 is listed in the following table.

Table 15.4 Number of Chip Select Signals

Unit Name	Chip Select Index					
	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
CSIH0	CSx (x = 0, 1)	CSx (x = 0 to 3)	CSx (x = 0 to 7)	CSx (x = 0 to 7)	CSx (x = 0 to 7)	CSx (x = 0 to 7)
CSIH1	—	—	CSx (x = 0 to 3)	CSx (x = 0 to 5)	CSx (x = 0 to 5)	CSx (x = 0 to 5)
CSIH2	—	—	CSx (x = 0 to 3)	CSx (x = 0 to 3)	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)

15.1.2 Register Base Address

CSIH base addresses are listed in the following table.

CSIH register addresses are given as offsets from the base addresses in general.

Table 15.5 Register Base Address

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

15.1.3 Clock Supply

The CSIH clock supply is shown in the following table.

Table 15.6 Clock Supply

Unit Name	Clock for the Unit	Internal Clock Signal
CSIHn	PCLK	CKSCLK_ICSI

15.1.4 Interrupt Requests

CSIH interrupt requests are listed in the following table.

Table 15.7 Interrupt Requests

Unit Interrupt Name	Outline	Interrupt Number	DMA Trigger Number
CSIH0			
INTCSIHTIC	Communication status interrupt	21	6 (channels 8 to 15)
INTCSIHTIR	Receive status interrupt	22	7 (channels 8 to 15)
INTCSIHTIRE	Communication error interrupt	23	—
INTCSIHTIJC	Job completion interrupt	24	8 (channels 8 to 15)
CSIH1			
INTCSIHTIC	Communication status interrupt	8, 108	28 (channels 0 to 7)
INTCSIHTIR	Receive status interrupt	9, 109	29 (channels 0 to 7)
INTCSIHTIRE	Communication error interrupt	19, 110	—
INTCSIHTIJC	Job completion interrupt	20, 111	30 (channels 0 to 7)
CSIH2			
INTCSIHTIC	Communication status interrupt	0, 124	25 (channels 8 to 15)
INTCSIHTIR	Receive status interrupt	29, 125	26 (channels 8 to 15)
INTCSIHTIRE	Communication error interrupt	30, 126	—
INTCSIHTIJC	Job completion interrupt	31, 127	27 (channels 8 to 15)
CSIH3			
INTCSIHTIC	Communication status interrupt	1, 150	41 (channels 0 to 7)
INTCSIHTIR	Receive status interrupt	5, 151	42 (channels 0 to 7)
INTCSIHTIRE	Communication error interrupt	6, 152	—
INTCSIHTIJC	Job completion interrupt	7, 153	43 (channels 0 to 7)

15.1.5 Reset Sources

CSIH reset sources are listed in the following table. CSIH is initialized by these reset sources.

Table 15.8 Reset Sources

Unit Name	Reset Source
CSIHn	All reset sources (ISORES)

15.1.6 External Input/Output Signals

External input/output signals of CSIH are listed below.

Table 15.9 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
CSIH0		
CSIHTSCK	Serial clock signal	CSIH0SC
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
CSIHTRYO	Ready/busy output signal	CSIH0RYO
CSIHTCSS[7:0]*1	Chip select signal	CSIH0CSS[7:0]*1
CSIH1		
CSIHTSCK	Serial clock signal	CSIH1SC
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
CSIHTRYO	Ready/busy output signal	CSIH1RYO
CSIHTCSS[5:0]*1	Chip select signal	CSIH1CSS[5:0]*1
CSIH2		
CSIHTSCK	Serial clock signal	CSIH2SC
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
CSIHTRYO	Ready/busy output signal	CSIH2RYO
CSIHTCSS[5:0]*1	Chip select signal	CSIH2CSS[5:0]*1
CSIH3		
CSIHTSCK	Serial clock signal	CSIH3SC
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
CSIHTRYO	Ready/busy output signal	CSIH3RYO
CSIHTCSS[3:0]	Chip select signal	CSIH3CSS[3:0]

Note 1. For the number of chip select signals, see **Table 15.4, Number of Chip Select Signals**.

15.1.7 Data Consistency Check

The following table lists the port pins on which CSIHnSO pin functions are multiplexed and data consistency checking. See **Section 15.5.12, Error Detection** for details on data consistency checking.

Table 15.10 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

15.2 Overview

15.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Master mode and slave mode selectable
- Multiple slaves configuration plus RCB (Recessive Configuration for Broadcasting) due to eight configurable chip select output signals
- Slave select input signal ($\overline{\text{CSIHTSSI}}$) is usable
- Built-in baud rate generators
- 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 up to PCLK/4)
  - Slave mode: 5.0 MHz (however, it must be up to PCLK/6)
- Phase of clock and data selectable
- Data transfer with MSB or LSB first selectable
- Transfer data length selectable from 2 to 16 bits in 1-bit units
- EDL (Extended Data Length) function for transferring data with more than 16 bits is included
- Three selectable transfer modes:
  - transmit-only mode
  - receive-only mode
  - transmit/receive mode
- Built-in handshake function
- Error detection (data consistency check, parity, time-out, overflow, and overrun) is included
- Support of job concept
- 128 words I/O buffer memory
- Selectable direct access mode and memory mode (FIFO, dual buffer, and transmit-only buffer)
- Four different interrupt request signals (INTCSIHTIC, INTCSIHTIR, INTCSIHTIRE, INTCSIHTIJC)
- LBM (Loop Back Mode) function for self test is included
- CPU-controlled high-priority communication function
- Enforced chip select idle setting
- RCB (Recessive Configuration for Broadcasting) bit is included
- JOB enable control bit for AUTOSAR is included.

15.2.2 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 synchronous to the transmission clock.

The following table shows the most important registers for setting up the CSIH.

Register	Function
CSIHnCTL0	Enables/disables serial clock, and permits/ prohibits data transmission and data reception. Defines end-of-job behavior and enables/disables buffering (bypass of the buffer).
CSIHnCTL1	Controls options like 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	Registers to configure the communication protocol for each chip select signal

15.2.3 Block Diagram

The block diagram shows the main components of the CSIH.

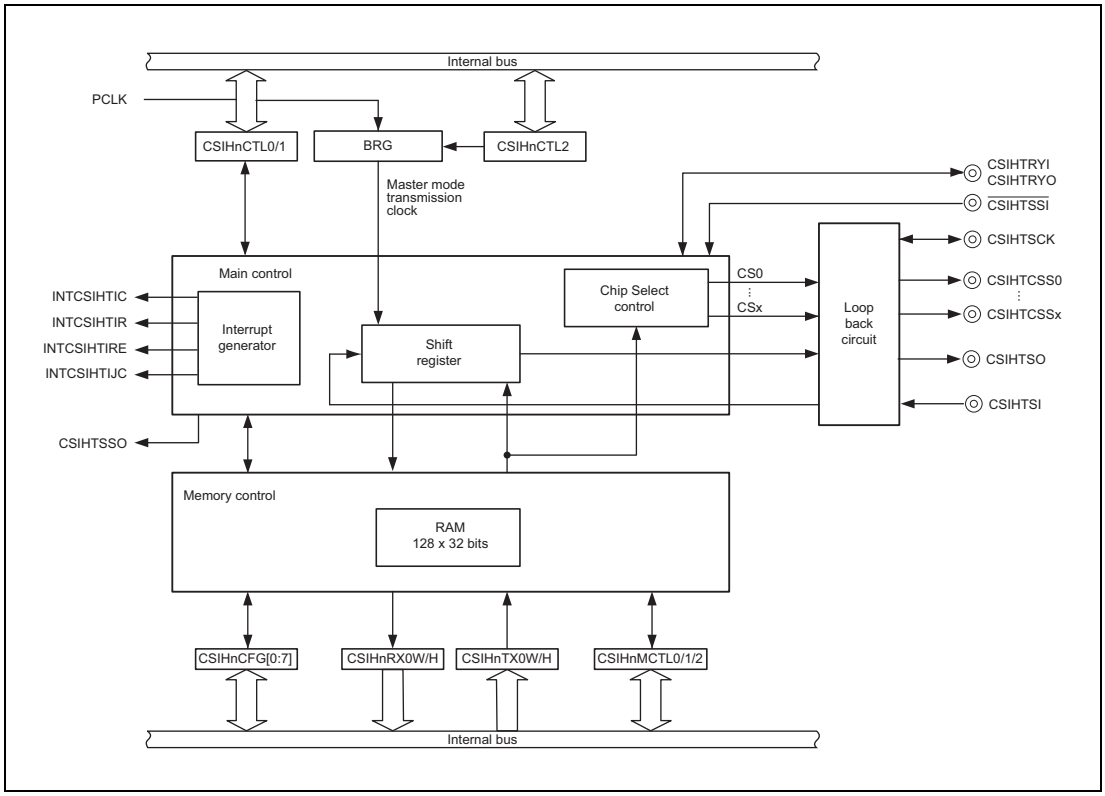


Figure 15.1 CSIH Block Diagram

In master mode, the transmission clock CSIHTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is provided from 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” separates between master and slave mode. In this context, only a master can control and communicate with several slaves (for details see **Section 15.5.1, Operating Modes (Master/Slave)**).
- The “job mode” is related to the AUTOSAR job concept (for details see **Section 15.5.3.3, Job Concept**).
- The “memory mode” takes the various configurations of the associated buffer memory into account (for details see **Section 15.5.6, CSIH Buffer Memory**).
- The “data transfer mode” specifies the kind of the communication – transmit-only, receive-only, or transmit/receive (for details see **Section 15.5.7, Data Transfer Modes**).

15.3 Registers

15.3.1 List of Registers

CSIH registers are listed in the following table.

For details about CSIHn_base, see **Section 15.1.2, Register Base Address**.

Table 15.11 Registers

Module	Register	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

15.3.2 CSIHnCTL0 — CSIHn Control Register 0

This register controls the operation clock and enables/disables transmission/reception and the memory part for transmission and/or reception. It forces the stop of communication at the end of the current job.

Access: This register can be read/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 15.12 CSIHnCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIHnPWR	Controls the operation clock. 0: Stops operation clock. 1: Provides operation clock. Clearing CSIHnPWR to 0 resets the internal circuits, stops operation, and sets CSIH to standby state. No clock is provided to internal circuits. If CSIHnPWR is cleared (to 0) during communication, ongoing communication is immediately aborted. In this case, communication setting must be started over.
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 to these bits, 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. 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

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.3 CSIHnCTL1 — CSIHn Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It 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/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 15.13 CSIHnCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, 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 15.3.11, 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 memory mode/transmit-only buffer mode). For details, see **Section 15.4.3, INTCSIHTIC (Communication Status Interrupt)**.
15 to 8	CSIHnCSx	Selects the active output level of chip select signal x (CSIHTCSSx). 0: Chip select is active low. 1: Chip select is active high. For details, see **Section 15.5.3, 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 15.5.8.2, Data Length Greater than 16 Bits**.

Table 15.13 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 15.5.3.3, 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 CSIHnPHE = 1 and this bit to 1.
5	CSIHnDCS	Enables/disables data consistency check. 0: Disables data consistency check. 1: Enables data consistency check. For details, see **Section 15.5.12.1, Data Consistency Check**.
4	CSIHnCSRI	Defines chip select signal behavior after last data transfer. 0: Chip select signal holds 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 15.5.13, 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 15.4.2, Interrupt Delay**.
1	CSIHnHSE	Enables/disables handshake mode. 0: Disables the handshake function. 1: Enables the handshake function. For details see **Section 15.5.11, Handshake Function**.
0	CSIHnSSE	Enables/disables the slave select function. 0: Input signal $\overline{\text{CSIHTSSI}}$ is ignored. 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 15.5.2, Master/Slave Connections**).

Details about CSIHnCTL1.CSIHnSSE are shown in the following tables.

Table 15.14 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 15.15 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

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.4 CSIHnCTL2 — CSIHn Control Register 2

This register selects operating mode and the basic clock value, and specifies the transfer clock frequency.

For details see **Section 15.5.5, Transmission Clock Selection**.

Access: This register can be read/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 15.16 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 to these bits, 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 according to the 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.CSIH0BRSS[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)

CAUTION

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.5 CSIHnSTR0 — CSIHn Status Register 0

This register indicates the status of CSIH.

Access: This register can only 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 15.17 CSIHnSTR0 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 24	CSIHnSRP[7:0]	Indicates the number of received data packets 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 memory mode, or transmit-only buffer memory 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 packets is managed by CSIHnMCTL2.CSIHnND[7:0].
23 to 16	CSIHnSPF[7:0]	Indicates the number of unsent 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 unsent data packets (0 to 128) … 80_H Other than the above — Undefined These bits are cleared by CSIHnSTCR0.CSIHnPCT. In direct access mode, dual buffer memory mode, or transmit-only buffer memory 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 packets is managed by CSIHnMCTL2.CSIHnND[7:0].

Table 15.17 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 is detected.
1: A time out error is detected.
For details, see Section 15.5.12.3, 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, clearing to 0 takes precedence over setting to 1.
This bit is also initialized when CSIHnCTL0.CSIHnPWR is changed to 0 to 1 or 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 is detected.
1: An overflow error is detected.
For details, see Section 15.5.12.4, 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, clearing to 0 takes precedence over setting to 1.
This bit is also initialized when CSIHnCTL0.CSIHnPWR is changed to 0 to 1 or 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: Transmission is in progress or being prepared.
Conditions for setting and clearing this bit are shown in the following tables.
<table>
<tr><th rowspan="2">Master Mode</th><th colspan="2">Set by</th><th rowspan="2">Cleared by</th></tr>
<tr><th>Direct Access Mode, FIFO Mode</th><th>Double Buffer Mode, Transmit-Only 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">Bit CSIHnMCTL2.CSIHnBTST 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">Set By</th><th rowspan="2">Cleared by</th></tr>
<tr><th>Direct Access Mode, FIFO Mode</th><th>Double Buffer Mode, Transmit-Only 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">Bit CSIHnMCTL2.CSIHnBTST 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><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 unsent data or received data.</td></tr>
</table>

Table 15.17 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 unsent 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, clearing to 0 takes precedence over setting to 1. This bit is initialized when CSIHnCTL0.CSIHnPWR is changed from 0 to 1 or 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, clearing to 0 takes precedence over setting to 1. 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. Writing to this bit is enabled when CSIHnCTL0.CSIHnPWR = 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 15.18 Behavior in Memory Mode

Bit Name	Bit Position	Direct Access Mode	FIFO Mode	Transmit-Only Mode	Dual Buffer Mode
CSIHnSRP[7:0]	31 to 24	Fixed to 0	Number of received words	Fixed to 0	Fixed to 0
CSIHnSPF[7:0]	23 to 16	Fixed to 0	Number of unsent 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: Transmission 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

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.6 CSIHnSTCR0 — CSIHn Status Clear Register 0

This register clears the status flags of the CSIHnSTR0 status register.

Access: This register can be read/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 15.19 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 to these bits, 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 FIFO buffer pointers below (in FIFO mode, dual buffer mode, and transmit-only buffer mode) and the 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 to these bits, 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 to this bit, 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

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.7 CSIHnMCTL0 — CSIHn Memory Control Register 0

This register selects the memory mode and the time-out setting.

Access: This register can be read/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 15.20 CSIHnMCTL0 Register Contents

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, 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 15.5.12.3, Time-Out Error**.

CAUTION

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.8 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/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 15.21 CSIHnMCTL1 Register Contents

Bit Position	Bit name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing to these bits, 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 unsent 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 to these bits, 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

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.9 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 in buffer mode.

Access: This register can be read/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 15.22 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 to these bits, 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 to these bits, write the value after reset.

Table 15.22 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

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.10 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/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 15.23 CSIHnMRWP0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits.
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 to these bits, write the value after reset.

Table 15.23 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

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.11 CSIHnCFGx — CSIHn Configuration Register x

These up to eight registers specify for each chip select signal CSIHTCSSx prescaler, parity, data length, recessive configuration for broadcasting, serial data direction, clock and data phase, idle enforcement configuration, idle time, hold time, inter-data time, and setup time.

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 and data phase

In slave mode, set 0 to all the bits other than above in the CSIHnCFG0 register, and the CSIHnCFG1 to CSIHnCFG7 registers.

Access: This register can be read/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 15.24 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, combining 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 sending 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>No parity is transmitted</td><td>No parity is waited for.</td></tr>
<tr><td>0</td><td>1</td><td>Adds parity bit fixed to 0</td><td>Parity bit is waited for but not judged.</td></tr>
<tr><td>1</td><td>0</td><td>Adds odd parity only</td><td>Odd parity bit is waited for.</td></tr>
<tr><td>1</td><td>1</td><td>Adds even parity</td><td>Even parity bit is waited for.</td></tr>
</table></td></tr>
<tr><td>27 to 24</td><td>CSIHnDLSx [3:0]</td><td>Selects the data length for chip select sinal x.
<table>
<tr><th>CSHBAnDLSx[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
Data length between 1 bit and 6 bits requires that the EDL function is used (see also Section 15.5.8.2, Data Length Greater than 16 Bits). 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. Only when the previous transmit data is 16 bits while CSIHnEDL = 1, writing 1 to this bit is enabled.</td></tr>
<tr><td>23 to 20</td><td>Reserved</td><td>When read, the value after reset is returned. When writing to these bits, write the value after reset.</td></tr>
<tr><td>19</td><td>CSIHnRCBx</td><td>Selects the recessive configuration for broadcasting for chip select x.
0: Dominant (higher priority)
1: Recessive (lower priority)
For details, see Section 15.5.3.1, 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 15.5.9, Serial Data Direction Selection.</td></tr>
</table>

Table 15.24 CSIHnCFGx Register Contents (2/4)

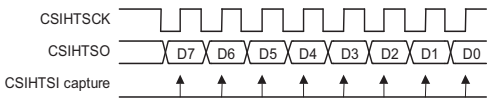
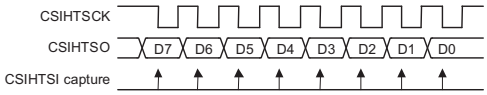
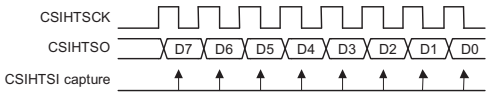
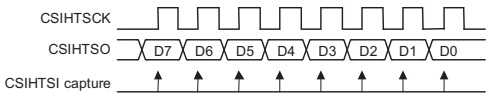
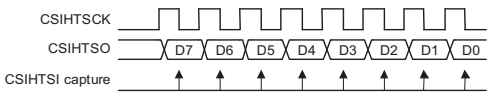
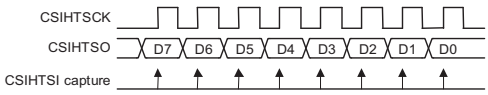
<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>17</td><td>CSIHnCKPx</td><td rowspan="2">CSIHnCKPx: Clock phase selection bit
CSIHnDAPx: Data phase selection bit
• CSIHnCTL1.CSIHnCKR = 0
<table><tr><th>CSIHn CKPx</th><th>CSIHn DAPx</th><th>Clock and Data Phase Selection</th></tr><tr><td>0</td><td>0</td><td>  </td></tr><tr><td>0</td><td>1</td><td>  </td></tr><tr><td>1</td><td>0</td><td>  </td></tr><tr><td>1</td><td>1</td><td>  </td></tr></table>• CSIHnCTL1.CSIHnCKR = 1
<table><tr><th>CSIHn CKPx</th><th>CSIHn DAPx</th><th>Clock and Data Phase Selection</th></tr><tr><td>0</td><td>0</td><td>  </td></tr><tr><td>0</td><td>1</td><td>  </td></tr><tr><td>1</td><td>—</td><td>Setting prohibited</td></tr></table></td></tr>
<tr><td>16</td><td>CSIHnDAPx</td></tr>
<tr><td>15</td><td>CSIHnIDLx</td><td>Selects the idle enforcement configuration for chip select x.
0: If the CSIHnTX0W.CSIHnCSx setting of two consecutive transfers is different, an idle state is inserted between two transfers. If the CSIHnTX0W.CSIHnCSx setting of two consecutive transfers is the same, an idle state is not inserted between two transfers.
1: Regardless of the CSIHnTX0W.CSIHnCSx setting of two consecutive transfers, an idle state is inserted between two transfers.
This bit is only available in master mode.
For details about the forced idle state, see Section 15.5.15, Enforced Chip Select Idle Setting.</td></tr>
</table>

Table 15.24 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.

CSIHn IDx[2:0]	Idle time when CSIHnCTL1.CSIHnSIT is 0	Idle time when CSIHnCTL1.CSIHnSIT is 1
000_B	0.5 transmission clock cycle	1.0 transmission clock cycle
001_B	1.0 transmission clock cycle	1.5 transmission clock cycle
010_B	1.5 transmission clock cycle	2.0 transmission clock cycle
011_B	2.5 transmission clock cycle	3.0 transmission clock cycle
100_B	3.5 transmission clock cycle	4.0 transmission clock cycle
101_B	4.5 transmission clock cycle	5.0 transmission clock cycle
110_B	6.5 transmission clock cycle	7.0 transmission clock cycle
111_B	8.5 transmission clock cycle	9.0 transmission clock cycle

These bits are only available in master mode.

11 to 8	CSIHnHDx[3:0]	Specifies the hold time for chip select signal x in transmission clock cycles.

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 cycle
0010_B	1.5 transmission clock cycle	2.0 transmission clock cycle
0011_B	2.5 transmission clock cycle	3.0 transmission clock cycle
0100_B	3.5 transmission clock cycle	4.0 transmission clock cycle
0101_B	4.5 transmission clock cycle	5.0 transmission clock cycle
0110_B	6.5 transmission clock cycle	7.0 transmission clock cycle
0111_B	8.5 transmission clock cycle	9.0 transmission clock cycle
1000_B	9.5 transmission clock cycle	10.0 transmission clock cycle
1001_B	10.5 transmission clock cycle	11.0 transmission clock cycle
1010_B	11.5 transmission clock cycle	12.0 transmission clock cycle
1011_B	12.5 transmission clock cycle	13.0 transmission clock cycle
1100_B	14.5 transmission clock cycle	15.0 transmission clock cycle
1101_B	16.5 transmission clock cycle	17.0 transmission clock cycle
1110_B	18.5 transmission clock cycle	19.0 transmission clock cycle
1111_B	20.5 transmission clock cycle	21.0 transmission clock cycle

These bits are only available in master mode.

Table 15.24 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 cycle
0011_B	2.0 transmission clock cycle	2.5 transmission clock cycle
0100_B	3.0 transmission clock cycle	3.5 transmission clock cycle
0101_B	4.0 transmission clock cycle	4.5 transmission clock cycle
0110_B	6.0 transmission clock cycle	6.5 transmission clock cycle
0111_B	8.0 transmission clock cycle	8.5 transmission clock cycle
1000_B	9.0 transmission clock cycle	9.5 transmission clock cycle
1001_B	10.0 transmission clock cycle	10.5 transmission clock cycle
1010_B	11.0 transmission clock cycle	11.5 transmission clock cycle
1011_B	12.0 transmission clock cycle	12.5 transmission clock cycle
1100_B	14.0 transmission clock cycle	14.5 transmission clock cycle
1101_B	16.0 transmission clock cycle	16.5 transmission clock cycle
1110_B	18.0 transmission clock cycle	18.5 transmission clock cycle
1111_B	20.0 transmission clock cycle	20.5 transmission clock cycle

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
0001_B	1.0 transmission clock cycle
0010_B	1.5 transmission clock cycle
0011_B	2.5 transmission clock cycle
0100_B	3.5 transmission clock cycle
0101_B	4.5 transmission clock cycle
0110_B	6.5 transmission clock cycle
0111_B	8.5 transmission clock cycle
1000_B	9.5 transmission clock cycle
1001_B	10.5 transmission clock cycle
1010_B	11.5 transmission clock cycle
1011_B	12.5 transmission clock cycle
1100_B	14.5 transmission clock cycle
1101_B	16.5 transmission clock cycle
1110_B	18.5 transmission clock cycle
1111_B	20.5 transmission clock cycle

These bits are only available in master mode.

CAUTION

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.12 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/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 15.25 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 INTCSIHTIJC request in FIFO mode. 0: No interrupt is requested. 1: An interrupt is requested. Generates interrupt INTCSIHTIC or INT_CSIHTIJC after transmission. For details, see **Section 15.4.3, INTCSIHTIC (Communication Status Interrupt)** and **Section 15.4.6, 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 the job does not end. 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 to these bits, write the value after reset.

Table 15.25 CSIHnTX0W Register Contents (2/2)

Bit Position	Bit name	Function
23 to 16	CSIHnCS[7:0]	Activates one or several 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

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.13 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 upper 16 bits of CSIHnTX0W are applied for transfer. 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/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 15.26 CSIHnTX0H Register Contents

Bit Position	Bit name	Function
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

CAUTION

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.14 CSIHnRX0W — CSIHn Receive Data Register 0 for Word Access

This register stores the received data.

Access: This register can only 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 15.27 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 x was activated for the associated reception. 1: Chip select x was deactivated for the associated reception.
15 to 0	CSIHnRX[15:0]	Stores the received data.

NOTE

To read this register, do so one serial clock cycle before an interrupt is generated.

CAUTION

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.15 CSIHnRX0Hn — 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 can only 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 15.28 CSIHnRX0H Register Contents

Bit Position	Bit name	Function
15 to 0	CSIHnRX[15:0]	Stores the received data.

CAUTION

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.16 CSIHnEMU — CSIHn Emulation Register

This register controls operation of SVSTOP.

Access: This register can be read/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 15.29 CSIHnEMU Register Contents

Bit Position	Bit name	Function
7	CSIHn SVSDIS	Selects 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 to these bits, write the value after reset.

CAUTION

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.17 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 15.5.5, Transmission Clock Selection**.

Access: This register can be read/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
	—	—	—	—	CSIH0BRS[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 15.30 CSIHnBRSy Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing to these bits, 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

For the setting of this register, see **Table 15.31, Notes on Setting Registers**.

15.3.18 List of Caution

Table 15.31 Notes on Setting Registers (1/2)

Register	Bit	Content
CSIHnCTL0	CSIHnPWR	If this bit is cleared during communication, ongoing communication is suspended. After cancelling the suspension, 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 with the CSIHnCTL0.CSIHnPWR bit.) Do not modify these bits while CSIHnSTR0.CSIHnTSF = 1, because the specified operation is not guaranteed if ongoing communication is suspended.
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 modify at the same time with the CSIHnCTL0.CSIHnPWR bit.) Modification of this bit is only allowed 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 function 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 allowed 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 of this bit 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 max transfer clock frequency is as follows. • Master mode: 10.0 MHz (however, it must be up to PCLK/4) • Slave mode: 5.0 MHz (however, it must be up to PCLK/6)
CSIHnSTR0	CSIHnSRP[7:0] CSIHnSPF[7:0] CSIHnFLF CSIHnEMF CSIHnTSF	Writing these bits is prohibited, and only reading is permitted.
CSIHnSTR0	CSIHnTMOE CSIHnOFE CSIHnDCE CSIHnPE CSIHnOVE	Writing 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 to1 during communication, ongoing communication is suspended.
CSIHnMCTL0	CSIHnMMS[1:0]	Modification of these bits is only allowed while CSIHnCTL0.CSIHnPWR = 0 and CSIHnCTL0.CSIHnMBS = 0.

Table 15.31 Notes on Setting Registers (2/2)

Register	Bit	Content
CSIHnMCTL0	CSIHnTO[4:0]	Modification of these bits is only allowed 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]	Write to these bits while communication is permitted.
CSIHnMCTL2	CSIHnBTST CSIHnND[7:0] CSIHnSOP[6:0]	Writing these bits is prohibited when CSIHnCTL0.CSIHnPWR = 0. Writing these bits is prohibited when CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0. Writing these bits is prohibited when CSIHnSTR0.CSIHnTSF = 1. Writing these bits is prohibited in direct access or FIFO mode.
CSIHnMRWP0	CSIHnRRA[6:0]	Write to these bits while communication is permitted. Writing 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]	Write to these bits while communication is permitted. Writing 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 allowed 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 allowed 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 this bit is only allowed while CSIHnCTL0.CSIHnPWR = 0. As CSIHnCTL1.CSIHnCKR must be used, set this bit to 0 in slave mode. If CS is not used, use this bit instead of the CSIHnCTL1.CSIHnCKR bit, and clear this bit to 0.
CSIHnTX0W	CSIHnEOJ CSIHnCIRE	This bit is only valid when CSIHnCTL1.CSIHnJE = 1. While CSIHnCTL1.CSIHnJE = 0, the value of this bit is ignored even if 1 is read. Set these bits to 0 in slave mode.
CSIHnTX0W	CSIHnEDL	This bit is 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 to these bits while communication is prohibited in FIFO mode. While CSIHnCTL0.CSIHnPWR = 0 and FIFO mode is on, reading and writing these bits are prohibited. While CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0, writing to these bits are prohibited in direct access mode.
CSIHnRX0W CSIHnRX0H		Writing is prohibited and only reading is permitted while CSIHnCTL0.CSIHnPWR=1. Writing is permitted while CSIHnCTL0.CSIHnPWR=0. These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1→ 0. While CSIHnCTL0.CSIHnPWR = 0, read/write access to these bits is prohibited in FIFO mode.
CSIHnEMU	CSIHnSVSDIS	Modification of this bit is only permitted while SVSTOP = 0.
CSIHnBRSy y = 0 to 3		Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0.

15.4 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)

15.4.1 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 15.32 Interrupt Generation

		Cause of Interrupt	
Memory Mode	**Interrupt**	**Job Mode Disabled CSIHnCTL1.CSIHnJE = 0**	**Job Mode Enabled CSIHnCTL1.CSIHnJE = 1**
FIFO	INTCSIHTIC	Tx data empty*1	Tx data empty*1 except job abortion*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 abortion*4
Transmit-only buffer, dual 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 abortion*4
Direct access	INTCSIHTIC	One single 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 abortion*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 abortion 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.

15.4.2 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by one half period of the transmission clock, CSIHTSCK. This is not possible in slave mode.

The delay is specified by setting bit CSIHnCTL1.CSIHnSIT = 1.

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 and data phase), and CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B (data length 8 bits).

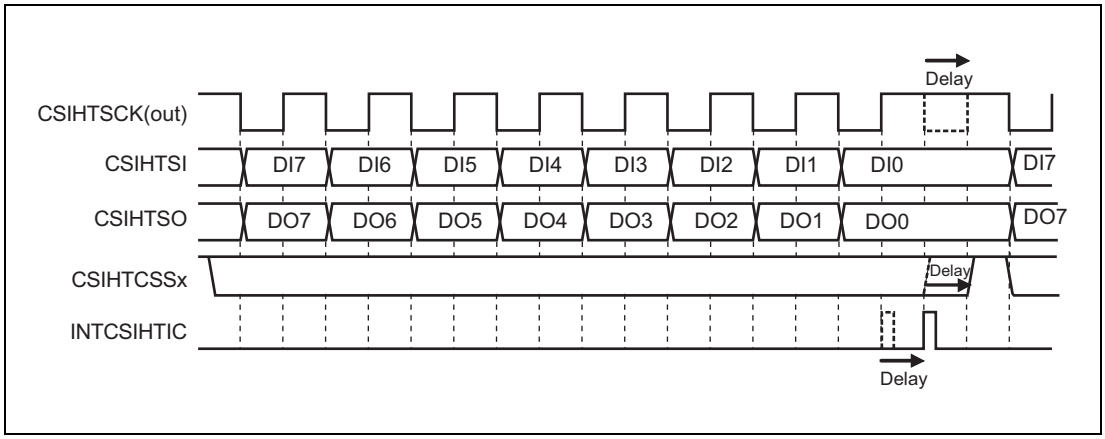


Figure 15.2 Interrupt Delay Function (CSIHnCTL1.CSIHnSIT = 1)

Setting CSIHnCTL1.CSIHnSIT = 1 adds half period delay to the transmission clock. This delays also the end of the present chip select signal (CSIHTCSSx).

15.4.3 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 15.33 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 is about to be missing in the FIFO, indicating to the application that new data should be added. INTCSIHTIC is generated, if the number of data to be sent remaining in the FIFO (CSIHnSTR0.CSIHnSPF[7:0]) equals CSIHnMTCL1.CSIHnFES[6:0].	Similar to “when JE is 0”, an interrupt is generated when the number of transmit data remained in the FIFO CSIHnSTR0.CSIHnSPF[7:0] is the same number as CSIHnMCTL1.CSIHnFES[6:0]. At the time of job abortion, no interrupt is generated.
Transmit-only buffer, dual buffer	Generated at the End of communication. (Specified by the CSIHnMTLC2.CSIHnND[7:0] bit)	Generated when data with CSIHnTX0W.CSIHnCIRE = 1 is sent. Note that if data with CSIHnTX0W.CSIHnCIRE = 1 and job abortion*1 are sent, 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 abortion 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.

15.4.3.1 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)

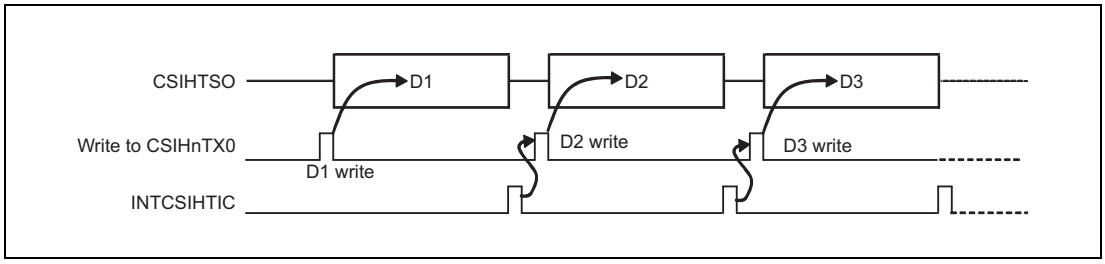


Figure 15.3 Generation of INTCSIHTIC after Transfer (CSIHnCTL1.CSIHnSLIT = 0)

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 is free for the next data. This is specified by setting CSIHnCTL1.CSIHnSLIT = 1.

The effect is illustrated in the figure below.

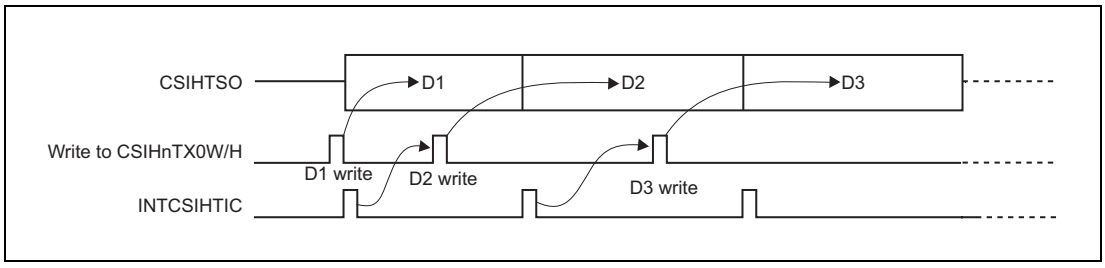


Figure 15.4 Immediate Generation of INTCSIHTIC (CSIHnCTL1.CSIHnSLIT = 1)

Thus, the new data can be written in advance.

NOTE

During high priority communication in transmit-only buffer mode, the operation is in the same way as direct access mode.

15.4.3.2 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 and data phase
  (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

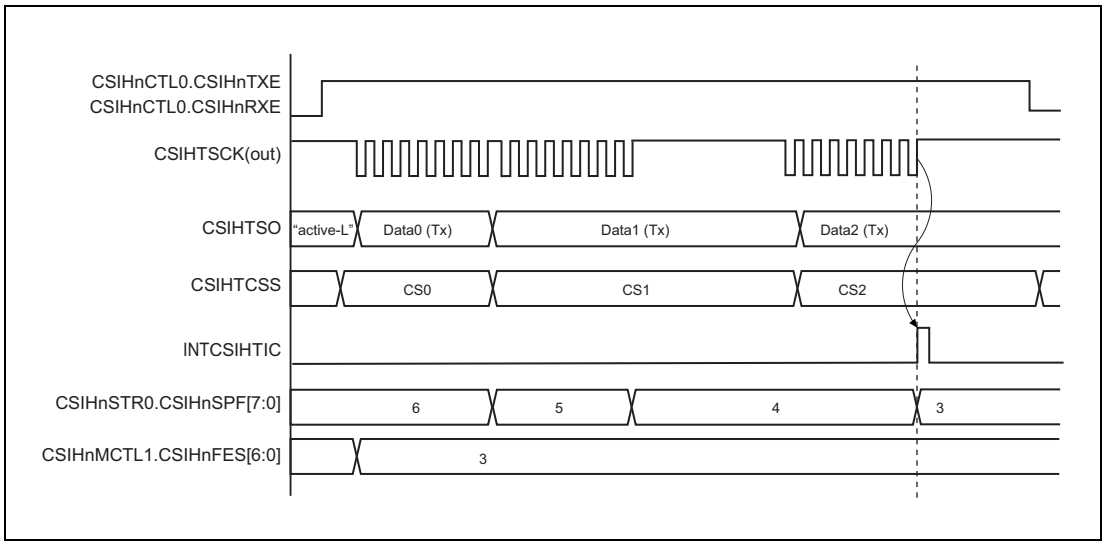


Figure 15.5 Generation of INTCSIHTIC in FIFO Memory Mode

The condition for “FIFO empty” is specified in CSIHnMCTL1.CSIHnFES[6:0]. In the example of the diagram above, the number of unsent data in FIFO is set to 3.

CSIHnSTR0.CSIHnSPF[7:0] indicates the number of unsent data. When both match, the interrupt INTCSIHTIC occurs.

15.4.3.3 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 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)

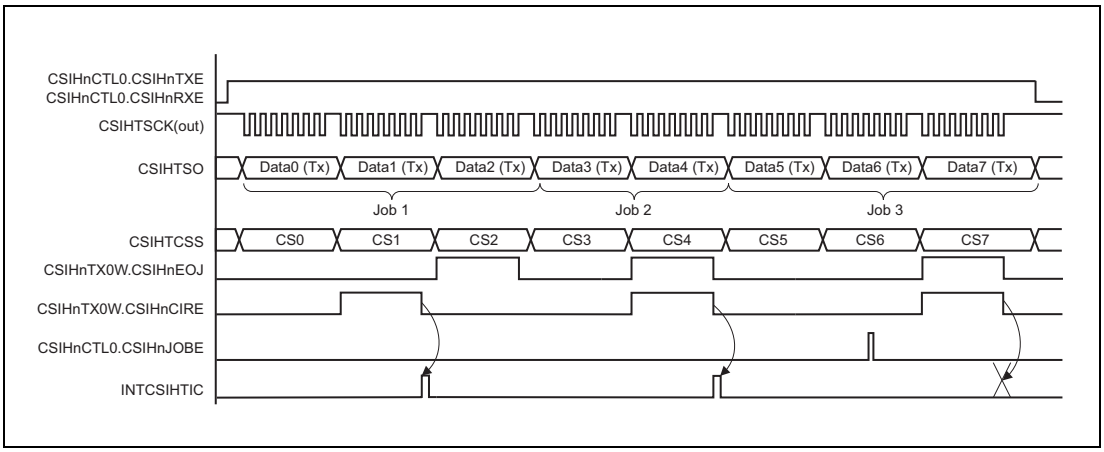


Figure 15.6 Generation of INTCSIHTIC in Job Mode

The rules for generating INTCSIHTIC in job mode are shown in the following table.

Table 15.34 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

15.4.4 INTCSIHTIR (Reception Status Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the conditions below.

Table 15.35 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 is 1 and the FIFO buffer is almost full with received data, indicating to 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	Generated when the communication has finished (as specified by the CSIHnMCTL2.CSIHnND[7:0] bit) and CSIHnCTL0.CSIHnRXE = 1.	Generated after every data transfer.
Transmit-only buffer, Direct access	Generated after every data transfer.	

15.4.4.1 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 and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

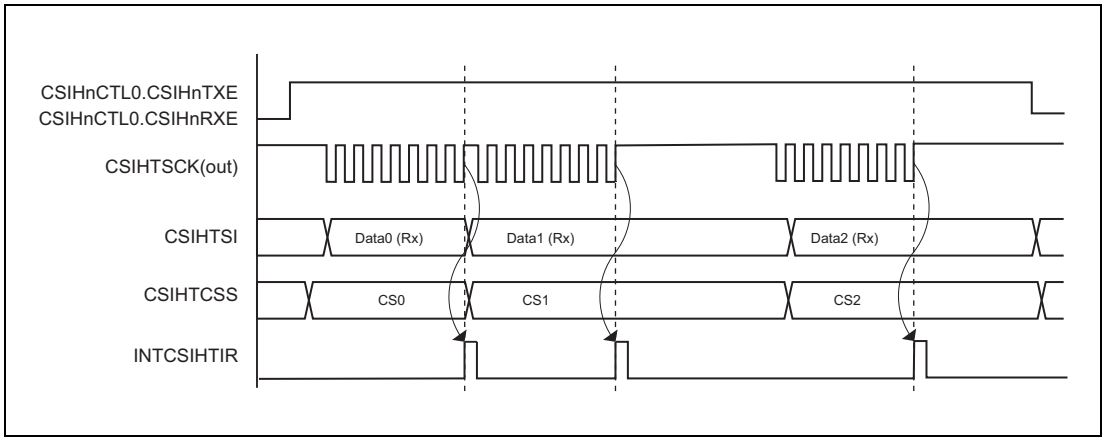


Figure 15.7 Generation of INTCSIHTIR in Direct Access Memory Mode

15.4.4.2 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 and data phase
  (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

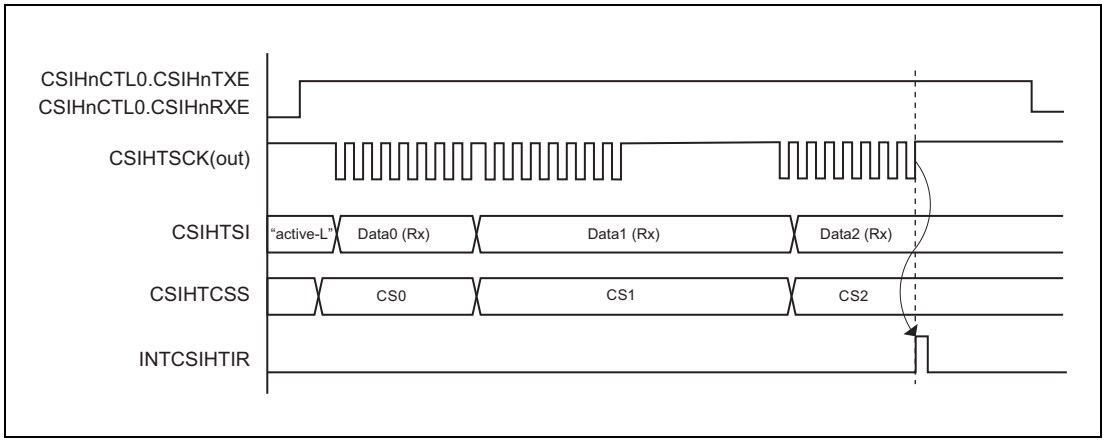


Figure 15.8 Generation of INTCSIHTIR in Dual Buffer Mode

15.4.5 INTCSIHTIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

For details about generation interruption timing, see **Section 15.5.12, Error Detection**.

Table 15.36 Data Error Types (1/2)

Error Type	Communication Status after Error Interrupt	Comment
FIFO overflow error	Interrupt is generated and communication continues.	The data are not written to the FIFO buffer and the overflow of data is lost, but communications started before the error is continued.,
Parity error	Interrupt is generated and communication continues.	—
Data consistency error	Interrupt is generated and communication continues.	—

Table 15.36 Data Error Types (2/2)

Error Type	Communication Status after Error Interrupt	Comment
Time-out error	Interrupt is generated and communication continues.	—
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 remained 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 15.5.12, Error Detection**.

15.4.6 INTCSIHTIJC (Job Completion Interrupt)

This interrupt supports the handling of jobs, see **Section 15.5.3.3, 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 15.37 INTCSIHTIJC Interrupt Generation

Memory Mode	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	Not applicable	Indicates that the communication has stopped at the end of a job after a job abortion*1 was triggered If FIFO empty is not detected and when CSIHnCIRE is 1, INTCSIHTIJC is generated.
Transmit-only buffer		Indicates that the communication has stopped at the end of a job after a job abortion*1 was triggered.
Dual buffer		
Direct access		

Note 1. Job abortion condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

15.5 Operation

15.5.1 Operating Modes (Master/Slave)

For a particular CSIH module, the master or slave mode determines the source of the serial clock.

15.5.1.1 Master Mode

In master mode, the serial transmission clock is generated by the internal baud rate generator (BRG) and provided to the slave(s) by signal CSIHTSCK.

Master mode is enabled by setting CSIHnCTL2.CSIHnPRS[2:0] to anything but 111_B. In master mode, the BRG frequency can be set by combining the CSIHnCTL2.CSIHnPRS[2:0] bit and the CSIHnBRSy.CSIHnBRS[11:0] bit.

(1) Chip select signals

In master mode, one or several chip select signals can be used. If several slaves are connected to the master, the chip select signals can be used to address one or several of the slaves. Only a 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 15.5.3, Chip Selection (CS) Features**.

(2) Clock defaults

The default level of CSIHTSCK depends on the clock phase 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 data bits, CSIHnCTL1.CSIHnCKR = 0, CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0, and MSB first.

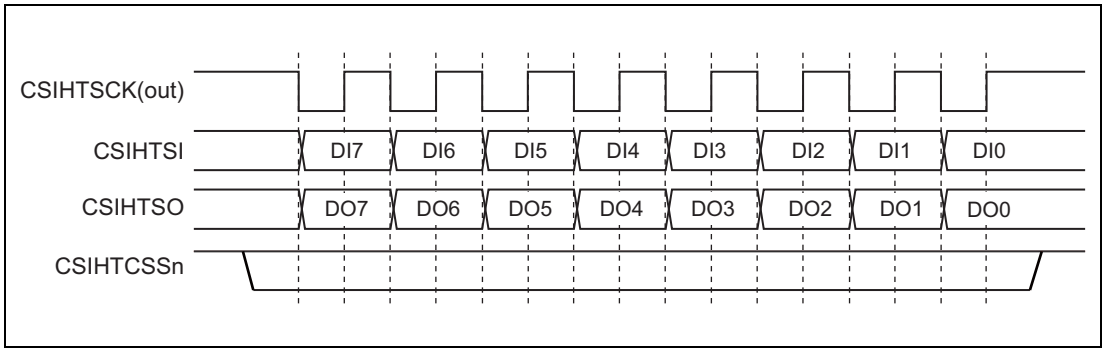


Figure 15.9 Transmit/Receive in Master Mode

15.5.1.2 Slave Mode

In slave mode, another device is the communication master and provides the transmission clock. Transmit/receive operation normally is started as soon as a clock signal is detected.

Slave mode is selected by setting the CSIHnCTL2.CSIHnPRS[2:0] bit to 111_B.

In slave mode, the transmission protocol setting of the CSIHnCFG0 register is enabled (setting of the CSIHnCFG1-CSIHnCFG7 register is 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 bit CSIHnBRSy.CSIHnBRS[11:0] to 000_H.

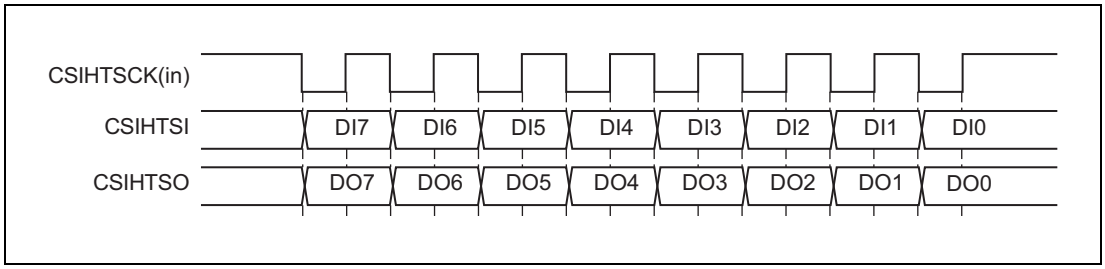


Figure 15.10 Transmit/Receive in Slave Mode

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.

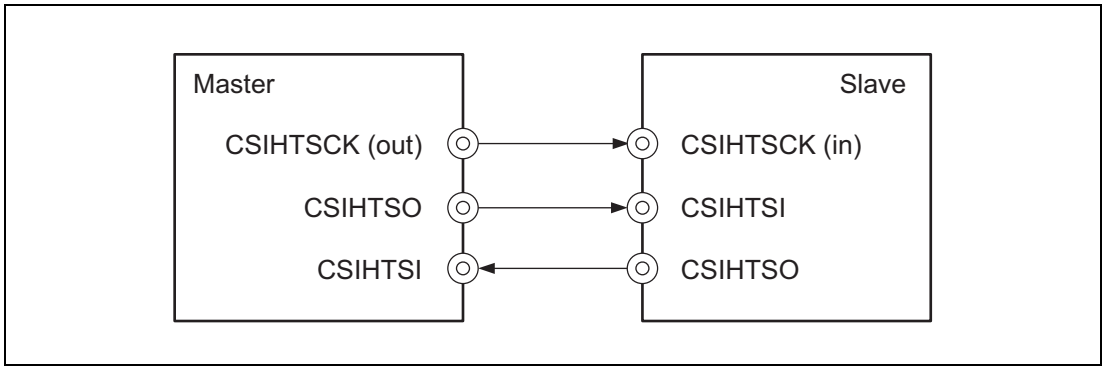


Figure 15.11 Direct Master/Slave Connection

15.5.2.2 One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this example, the master provides 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 bit CSIHnCTL1.CSIHnSSE.

u

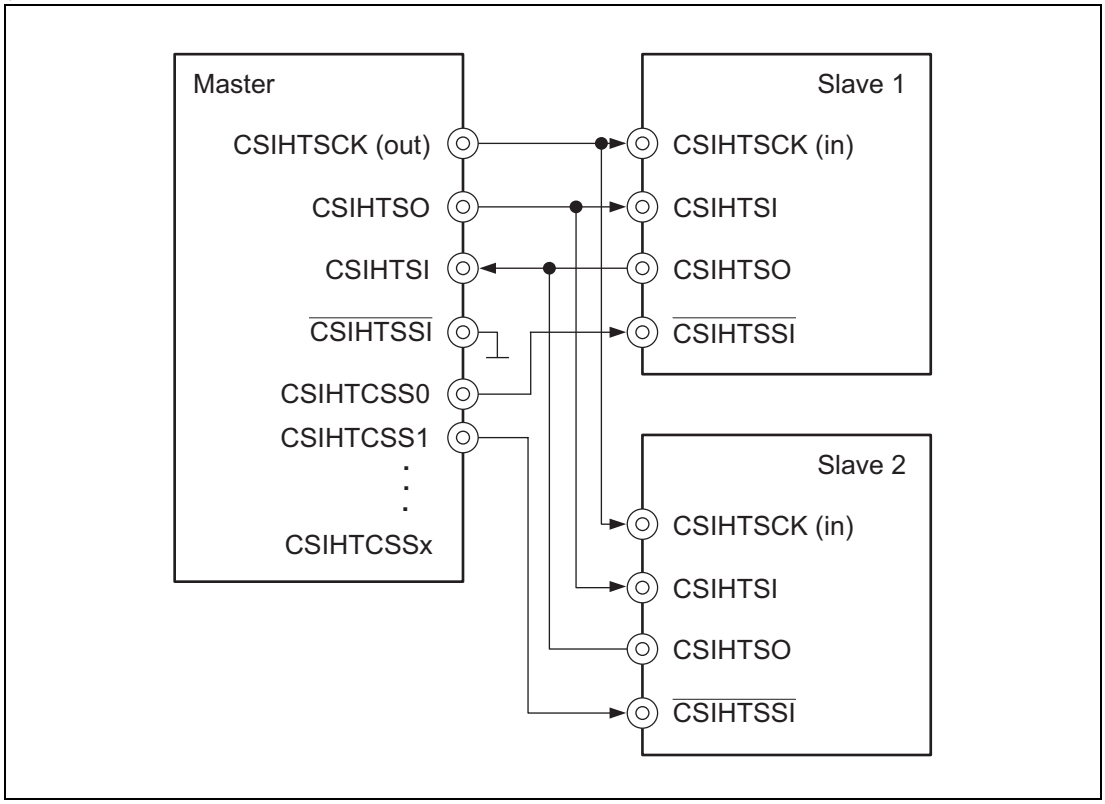


Figure 15.12 Connections between One Master and Multiple Slaves

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 has 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, its output CSIHTSO of a slave that is not selected is set to input mode in order to avoid interference with the output of another slave that was selected.

15.5.3 Chip Selection (CS) Features

The chip select signal, CSIHTCSSx can be used by the master to select one or several slaves for communication.

15.5.3.1 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 is only available in master mode are:

- Prescaler selection of the baud rate generator separately for each chip select signal
  (CSIHnCFGx.CSIHnBRSSx[1:0])
- Chip select priority: Separates between “dominant” and “recessive” chip select signals. The priority applies if two or more chip selects signals with different configurations are simultaneously activated for message broadcasting. In this case, the configuration that is set as dominant is used. (CSIHnCFGx.CSIHnRCBx)

The principle is also called “Recessive Configuration for Broadcasting” (RCB).

CAUTION

It is forbidden to specify several chip select signals as dominant with different configurations unless all dominant chip selects have the same configuration.

- Chip select timing:
  - Setup time T_{setup}: The time from setting the CS signal active to starting data output.
    (CSIHnCFGx.CSIHnSPx[3:0])
  - Inter-data time T_{inter}: The time between one data and the next following data while the same CS signal is active.
    (CSIHnCFGx.CSIHnINx[3:0])
  - Hold time T_{hold}: Hold time of CS active level before changing the CS.
    (CSIHnCFGx.CSIHnINx[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 15.13 provides an example when the default CSIHTCSS1 and CSIHTCSS2 signals are active low (CSIHnCTL1.CSIHnCSL1 bit = 0, CSIHnCTL1.CSIHnCSL2 bit = 0). The active level can be specified individually for each CS.

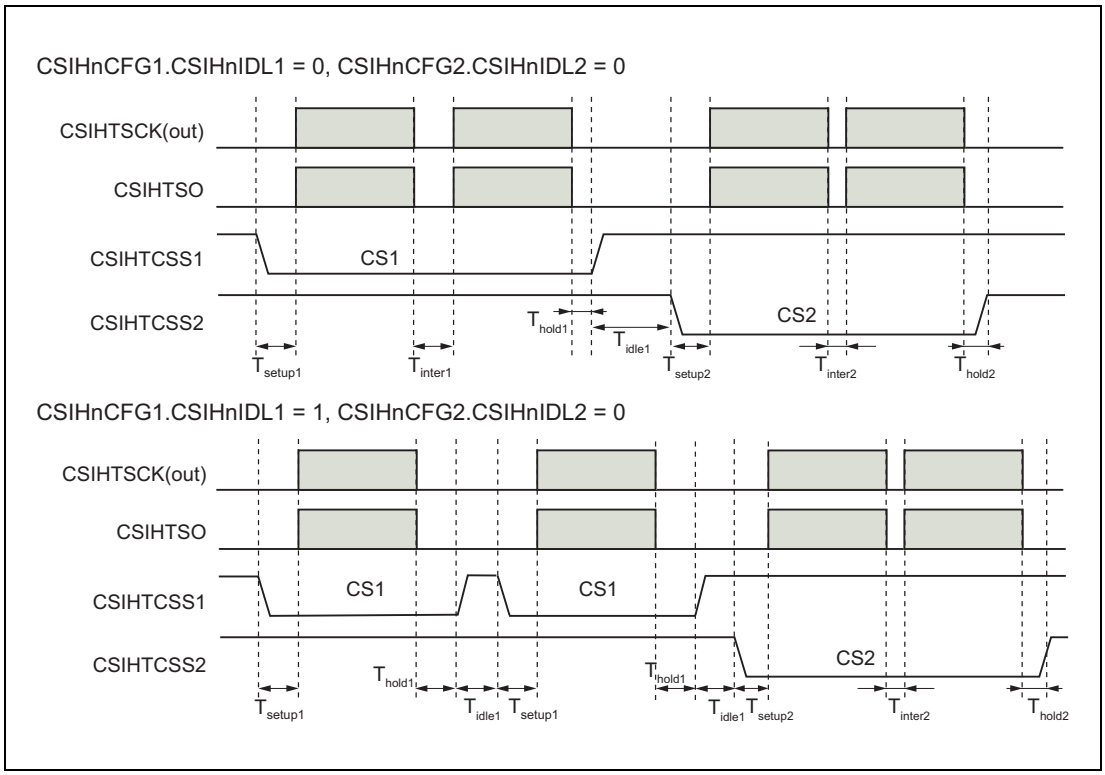


Figure 15.13 Chip Select Timings

Note that each CS signal can have a different value for setup, 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.

15.5.3.2 CS Example

The following figure shows an example of two consecutive data transmissions.

The first communication uses CS0 to communicate with one single slave. The second enables CS0 and CS1 to broadcast a message to two slaves. The priority of CS0 is set to “recessive: low priority”, the priority of CS1 to “dominant: high priority”. Consequently, the second is conducted by using CS1 settings which are set in dominant.

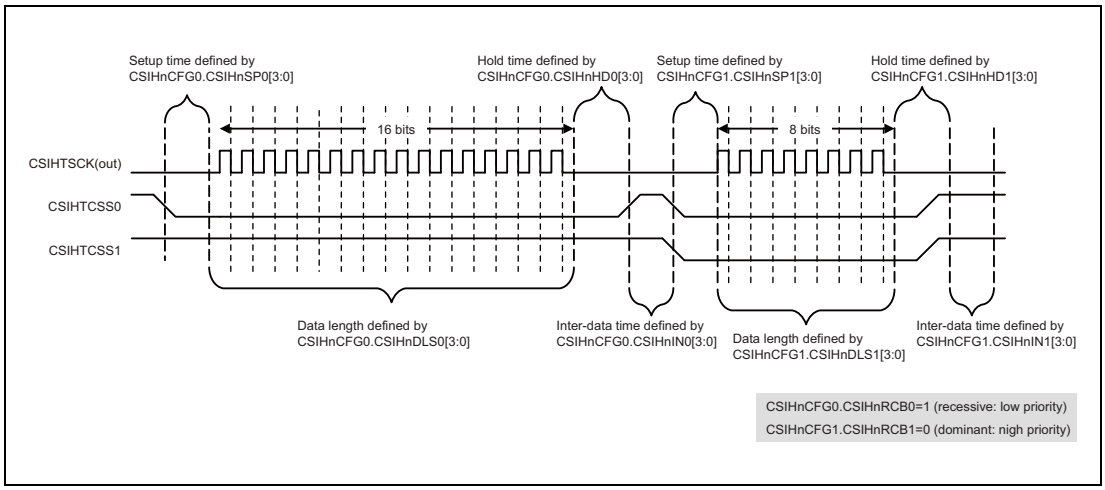


Figure 15.14 Chip Select and RCB Example

15.5.3.3 Job Concept

In terms of CSIH, a job consists of a 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.

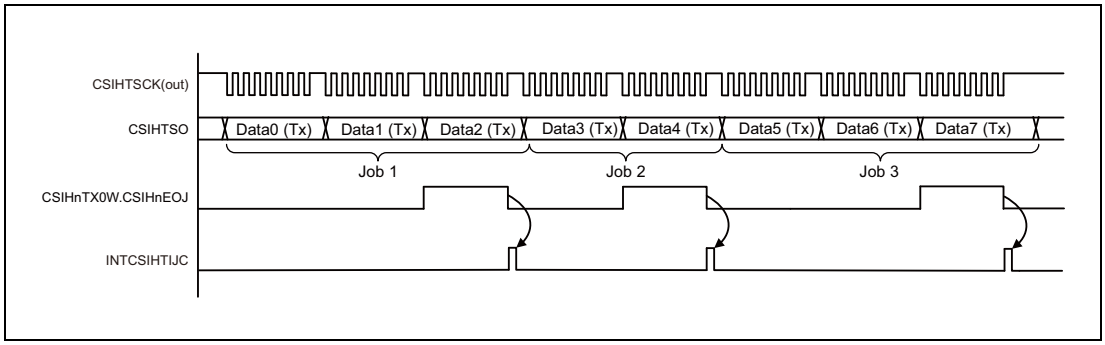


Figure 15.15 Job Examples

A job ends by transmitting data with CSIHnTX0W.CSIHnEOJ = 1.

A communication stop can be specified to occur after a job has finished. This is done by setting CSIHnCTL0.CSIHnJOBE. When CSIHnJOBE is set, the communication continues until data is sent, for which the CSIHnEOJ bit was set. After this data is sent, the communication is stopped and the interrupt INTCSIHTIJC is generated.

15.5.4 Details of Chip Select Timing

15.5.4.1 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 period of transmission clock (CSIHTSCK(out)).

If the idle time is set to 0.5 transmission clock periods (in CSIHnCFGx.CSIHnIDx[2:0]) and two consecutive data is sent with different CSIHnCFGx.CSIHnCKPx configuration, the idle time is automatically extended to one period of CSIHTSCK(out).

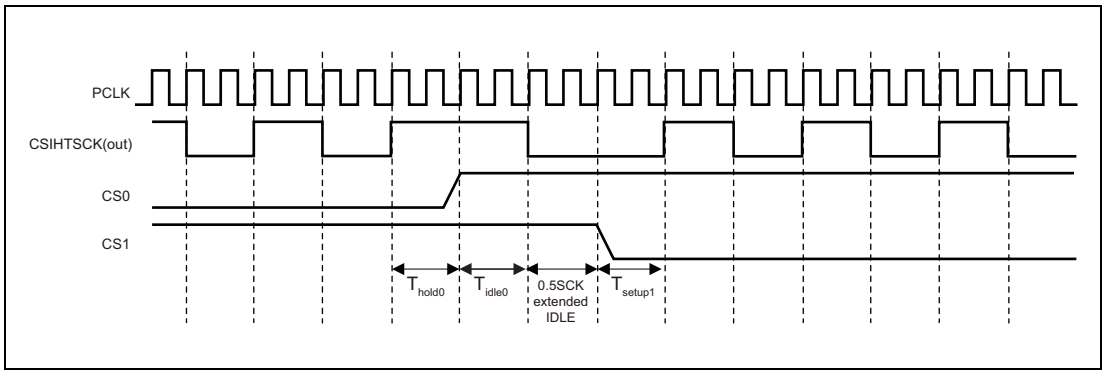


Figure 15.16 Clock Phase Timing with PCLK/4, $T_{hold}0 = T_{setup1}$ = 0.5CSIHTSCK, T_{idle0} = 0.5CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG1.CSIHnCKP1 = 1 (CSIHTCSS1)

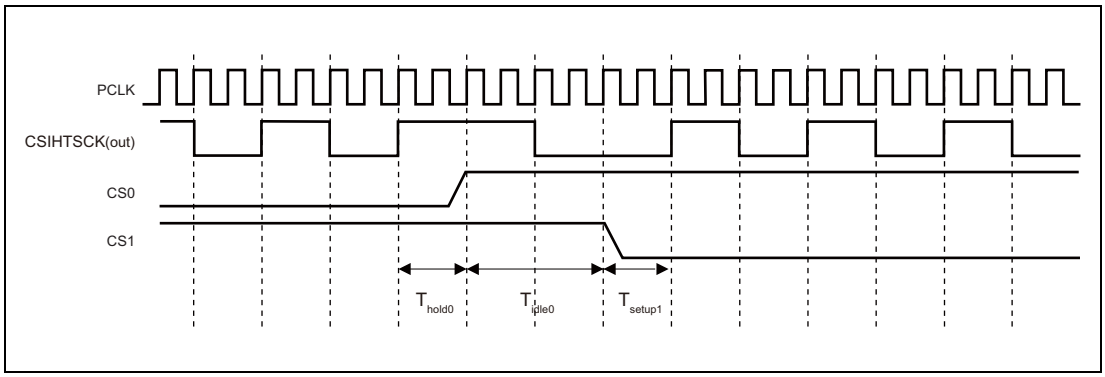


Figure 15.17 Clock Phase Timing with PCLK/4, $T_{hold}0 = T_{setup1}$ = 0.5CSIHTSCK, T_{idle0} = 1CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG1.CSIHnCKP1 = 1 (CSIHTCSS1)

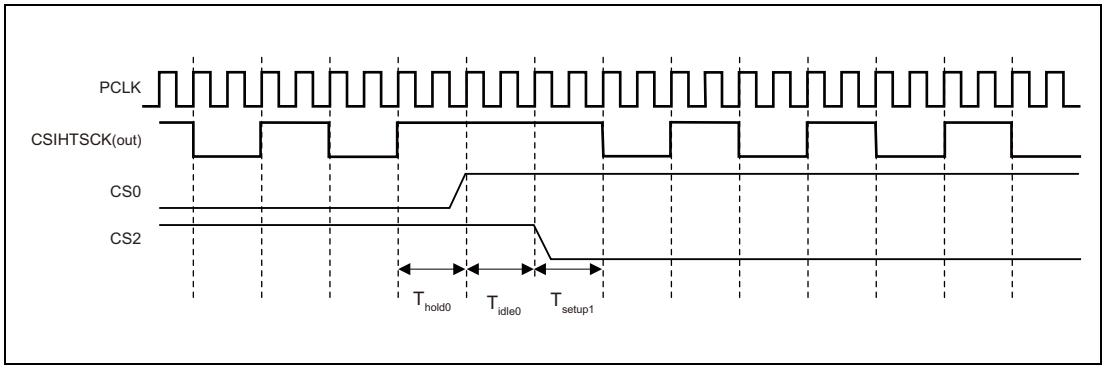


Figure 15.18 Clock Phase Timing with PCLK/4, $T_{hold}0 = T_{setup1}$ = 0.5CSIHTSCK, T_{idle0} = 0.5CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG2.CSIHnCKP2 = 0 (CSIHTCSS2)

15.5.4.2 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 described below.

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 the signals on CSIHTCSS[7:0] changing to the active level until output (on CSIHTSO) of the data to be transmitted.

Accordingly, there is a gap of 0.5 cycles of CSIHTSCK until output of an edge of the serial clock signal (CSIHTSCK).

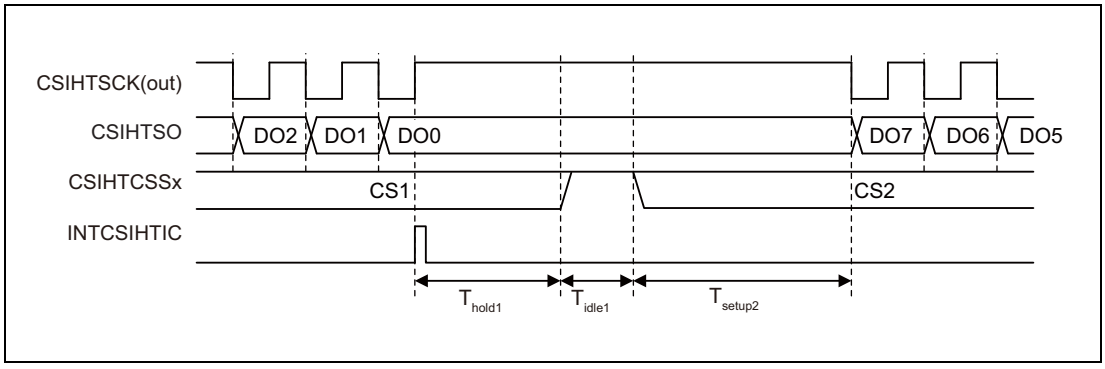


Figure 15.19 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 0

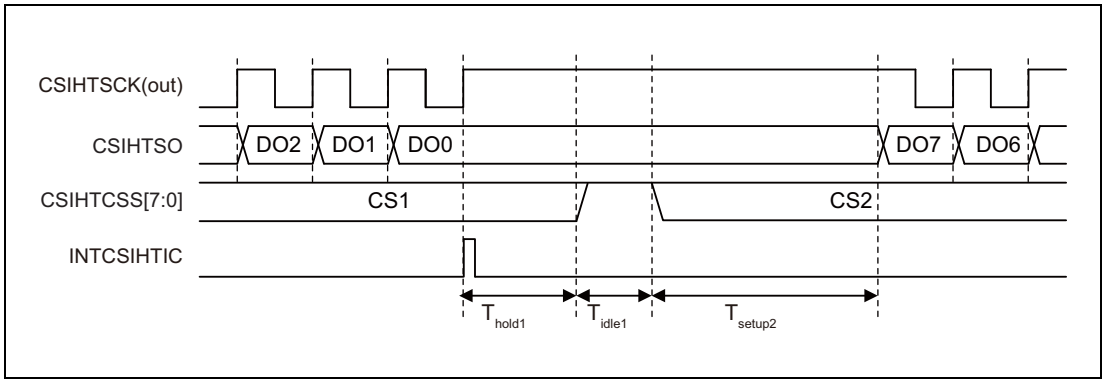


Figure 15.20 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 1, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 1

15.5.5 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 with the CSIHnCFGx.CSIHnBRSSx[1:0] bits.

The following figure shows a block diagram of the baud rate generator.

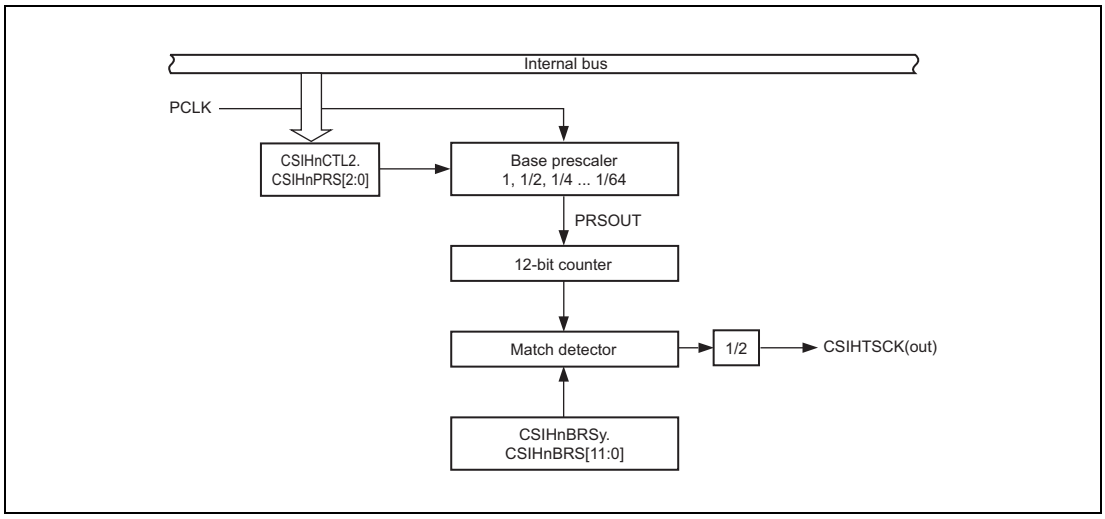


Figure 15.21 Baud Rate Generator Block Diagram

By setting CSIHnBRSy.CSIHnBRS[11:0] to 000_H, CSIHnBRSy.CSIHnBRS[11:0] disables the baud rate generator, and thus 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 / (2α × k × 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 followings.

- 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 up to PCLK/4)
  - In slave mode: 5.0 MHz (however, it must be up to PCLK/6)

15.5.6 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 bits data plus 7 bits 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

15.5.6.1 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 to the FIFO memory. Transmission and reception occur simultaneously – one data is sent, one data is received. 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 15.38 FIFO Mode

Pointer Description	Control Bit*1	Range
Number of unsent words	CCSIHnSTR0.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 also two FIFO status flags:

- CSIHnSTR0.CSIHnFLF: FIFO full
- CSIHnSTR0.CSIHnEMF: FIFO empty

When this mode is started, bit CSIHnSTCR0.CSIHnPCT must be set. By doing this, only CSIHnSTR0.CSIHnEMF is set, but not reset.

All FIFO pointers and FIFO flags excluding CSIHnSTR0.CSIHnEMF are reset and CSIHnSTR0.CSIHnEMF is set.

15.5.6.2 Dual Buffer Mode

In this mode, the memory is divided into two parts of equal size – this means 64 words for transmit data and 64 words for received data. In dual buffer mode, the respective buffer pointers indicate the values shown in the following table.

Table 15.39 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 remained in the transmit buffer	CSIHnMCTL2.CSIHnND[6:0]	0 to 64
Address to which data is sent to	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $00FC_H$

Note 1. Both pointers are automatically incremented after each read/write.

15.5.6.3 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 15.40 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 $00FC_H$
The number of transmit data remained in the transmit buffer	CSIHnMCTL2.CSIHnND[6:0]	0 to 128
Address to which data is sent to	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $007F_H$

Note 1. Pointers are automatically incremented after each read/write.

15.5.6.4 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.

15.5.7 Data Transfer Modes

15.5.7.1 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 bit CSIHnMCTL2.CSIHnBTST is set.

15.5.7.2 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 in the CSIHnTX0W or CSIHnTX0H register.

In slave mode, reception starts as soon as the CSIHTSCK transmission clock from the master is received. 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 bit CSIHnMCTL2.CSIHnBTST is set.

15.5.7.3 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 bit CSIHnMCTL2.CSIHnBTST is set.

15.5.7.4 Summary

The following table summarizes this section. It shows how data transfer is started in the various memory, operating, and transfer modes.

Table 15.41 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 on the CSIHnTX0H register
	Slave	Incoming clock from master	Incoming clock from the master
Transmit-only buffer, dual buffer	Master	CSIHnMCTL2.CSIHnBTST = 1	CSIHnMCTL2.CSIHnBTST = 1
	Slave	Incoming clock from master	Incoming clock from master

15.5.8 Data Length Selection

15.5.8.1 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):

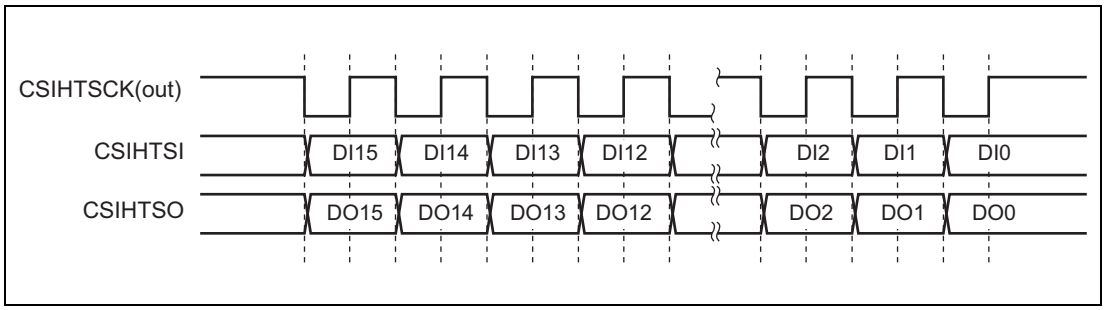


Figure 15.22 16 Bit Data Length, MSB First

Data length = 14 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1110_B):

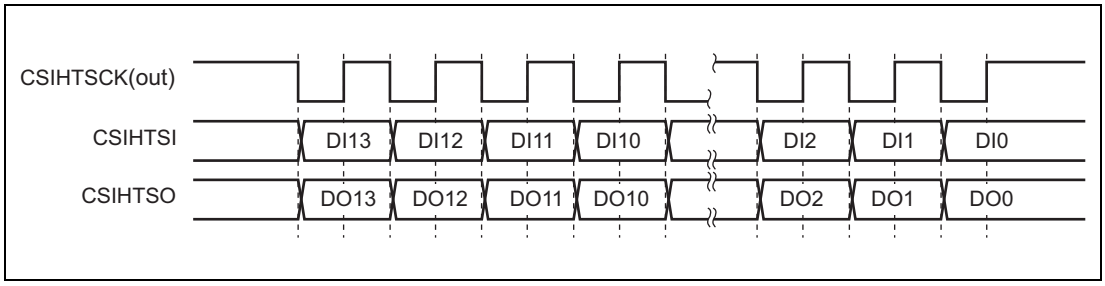


Figure 15.23 14 Bit Data Length, MSB First

15.5.8.2 Data Length Greater than 16 Bits

If the data to be sent/received exceeds 16 bits, the extended data length (EDL) feature can be used.

EDL function is enabled by setting bit CSIHnCTL1.CSIHnEDLE to 1.

EDL function works as follows:

- 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 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 for sending 40-bit data ($123456789A_H$) to CS0:

40 bits are split into two blocks of 16 bits plus 8 bits.

- Initialize CSIHnCFG0.CSIHnDLS0[3:0] = 8.
- To send $123456789A_H$ 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.

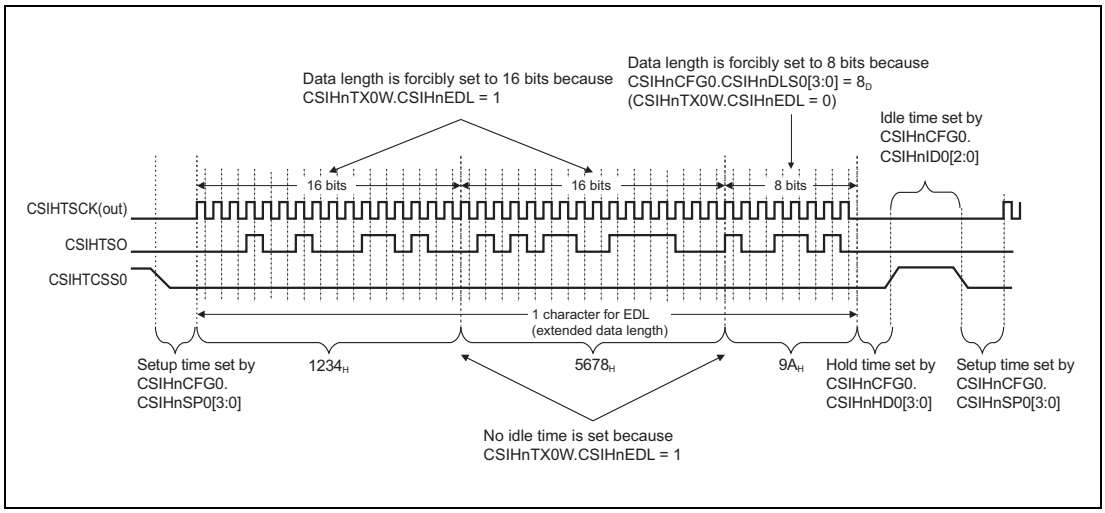


Figure 15.24 EDL Timing Diagram

NOTES

1. Data lengths settings lower than 7 bits are only permitted in combination with EDL mode.
2. It is not possible to send 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. When data is sent using extended data length (EDL) function, use the same chip select signal.
5. To consider the data direction, pay attention to the following example:
   - 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)
6. Operation is not guaranteed if CSIHnTX0W.CSIHnEOJ and CSIHnTX0W.CSIHnEDL are simultaneously set to “1” while CSIHnCTL1.CSIHnJE = 1 and CSIHnCTL1.CSIHnEDLE = 1.
7. EDL mode cannot be used in receive-only mode of slave mode. (CSIHnCTL2.CSIHnPRS[2:0] = 111_B, CSIHnCTL0.CSIHnTXE = 0, CSIHnCTL0.CSIHnRXE = 1)

15.5.9 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).

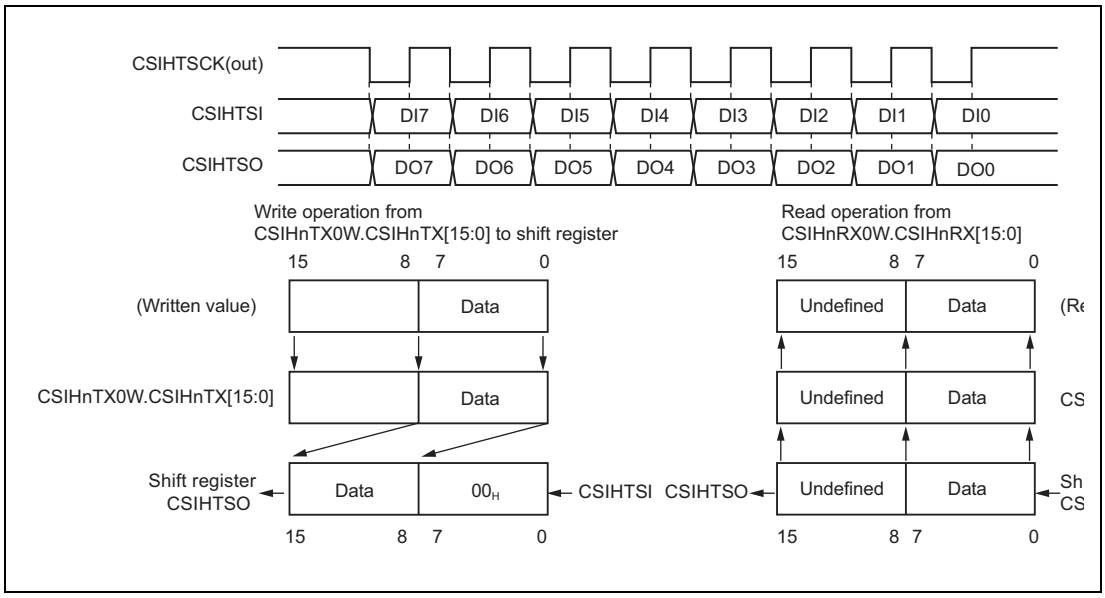


Figure 15.25 Serial Data Direction Select Function - MSB First (CSIHnDIRx = 0)

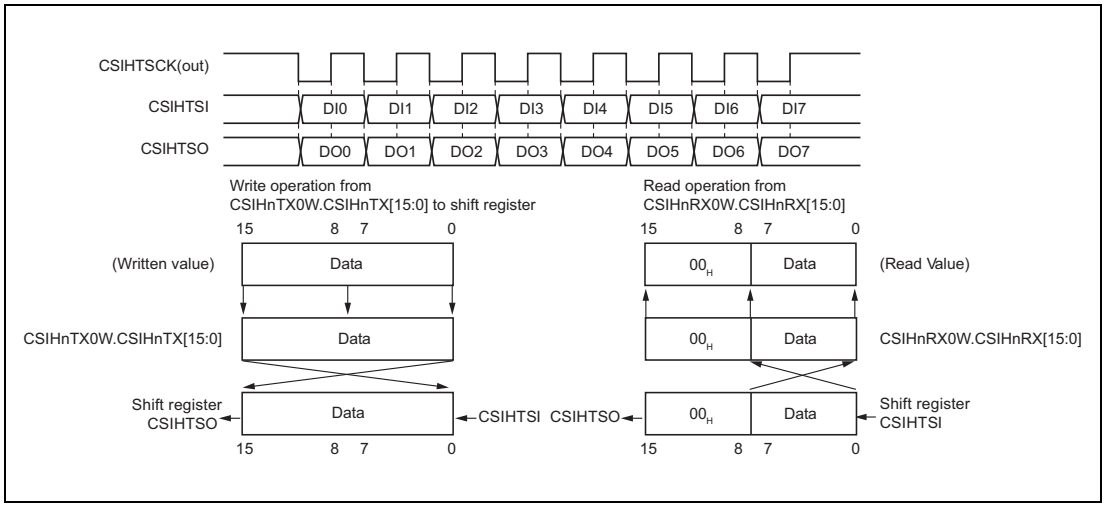


Figure 15.26 Serial Data Direction Select Function - LSB First (CSIHnDIRx = 1)

15.5.10 Slave Select (SS) Function

The Slave Select (SS) function realizes communication between one master and multiple slaves (SPI communications).

In master mode, the master device outputs the salve select signal (CSIHTCSSx) to select a single 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 15.5.2, Master/Slave Connections**, for an example of a connection using the SS function.

15.5.10.1 Communication Timing Using SS Function

The following figure illustrates the communication signal using the SS function and timings.

In slave mode, the data transfer configuration is determined by the CSIHnCFG0 register.

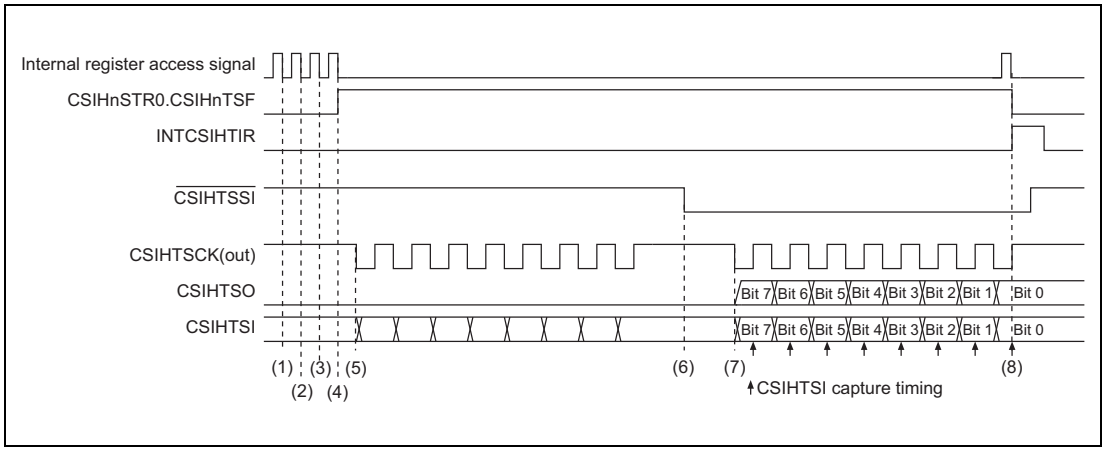


Figure 15.27 Tx/Rx Timing of Communication Using SS Function

(1) CSIH is put into 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) CSIH is set to transmit/receive mode (CSIHnCTL0.CSIHnTXE = 1, CSIHnCTL0.CSIHnRXE = 1, and CSIHnCTL0.CSIHnPWR = 1). Communication start is permitted.

(4) The “transmission in progress” 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 applied. Input at CSIHTSI is ignored.

(6) As soon as $\overline{\text{CSIHTSSI}}$ falls to low level, indicating that CSIHTSO is enabled and ready for transmission.

(7) Now, 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 when the reception is complete. The CSIHnRX0W/H register can be read.

15.5.10.2 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 in case of 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).

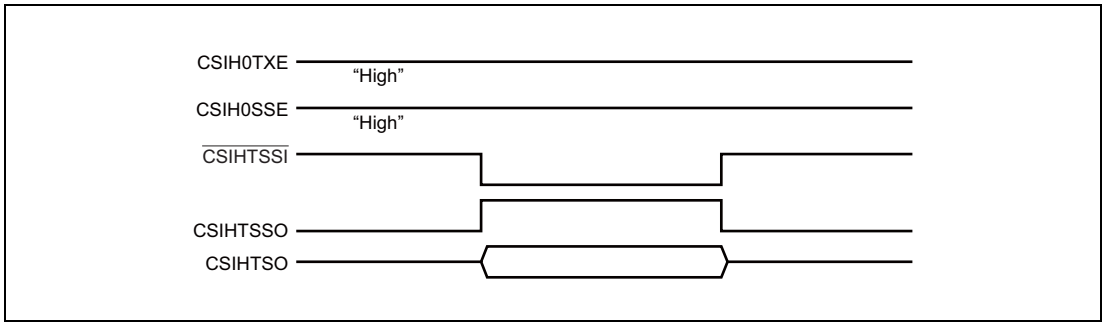


Figure 15.28 Operation of CSIHTSSO

CAUTION

If $\overline{\text{CSIHTSSI}}$ pin is changed during communication (CSIHnSTR0.CSIHnTSF = 1), current communication is not assured.

15.5.11 Handshake Function

CSIH features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by bit CSIHnCTL1.CSIHnHSE. For handshake, the signals CSIHTRYI and CSIHTRYO are used.

The timing depends on the data phase selection bit, CSIHnCFGx.CSIHnDAPx setting.

15.5.11.1 Slave Mode

When CSIHnCTL1.CSIHnHSE = 1 and the slave is busy, the CSIHTRYO signal outputs low level. 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 output indicates the busy state (is at the low level).

Table 15.42 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	

The example below is on the assumption of an eight-bit data length.

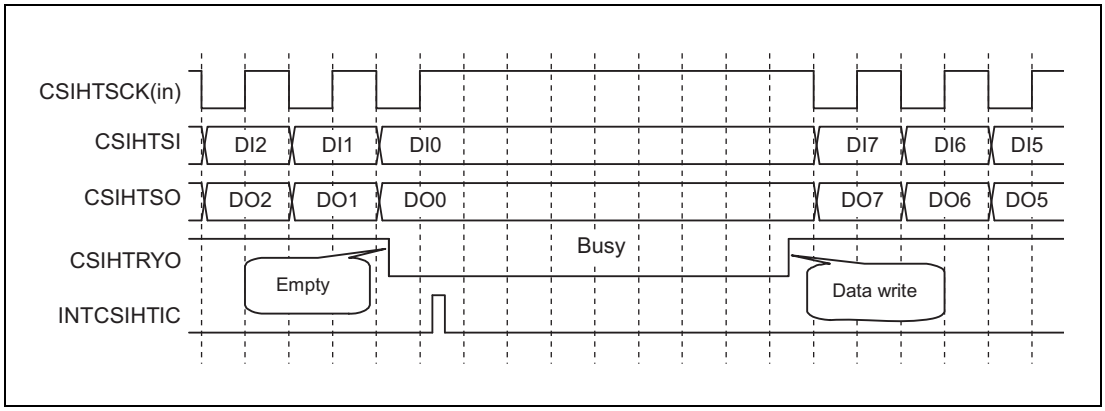


Figure 15.29 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 0)

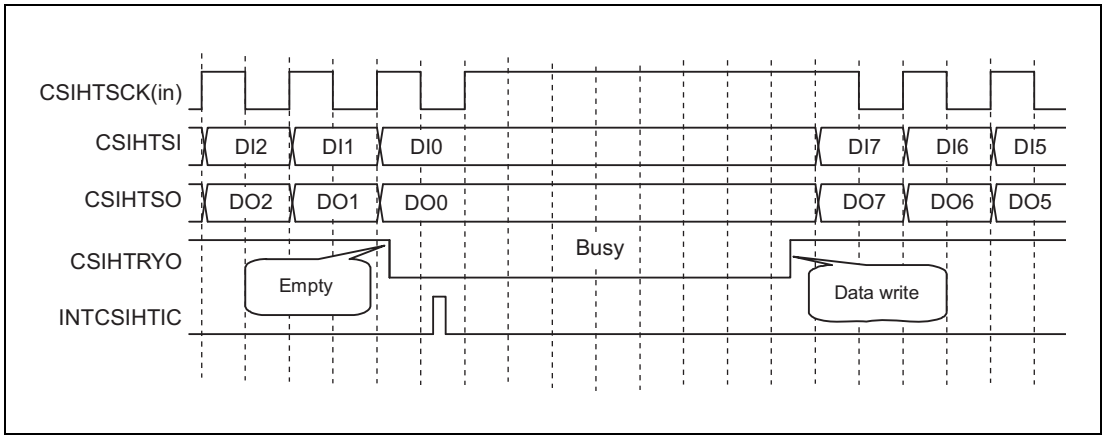


Figure 15.30 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 1)

2. When transmit 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 (CSIHnRX0W/H is full) because the previously received data is still in the CSIHnRX0W/H register.
   When CSIHnCTL0.CSIHnRXE is 1 and is in the following states, CSIHTRYO outputs busy state (low level).

Table 15.43 Memory Mode and Slave Reception State

Memory Mode	Slave Reception State
Direct access mode	When CSIHnRX0W or CSIHnRX0H is full
FIFO mode	When receive data is remained in buffer (CSIHnSTR0.CSIHnFLF = 1)
Dual buffer mode	No applicable case
Transmit-only buffer mode	When CSIHnRX0W or CSIHnRX0H is full

The example below is on the assumption of an eight-bit data length.

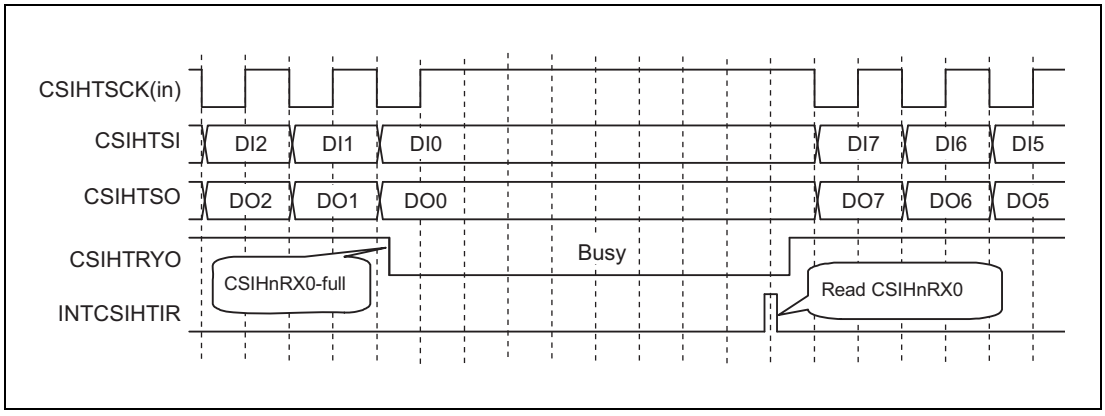


Figure 15.31 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 0)

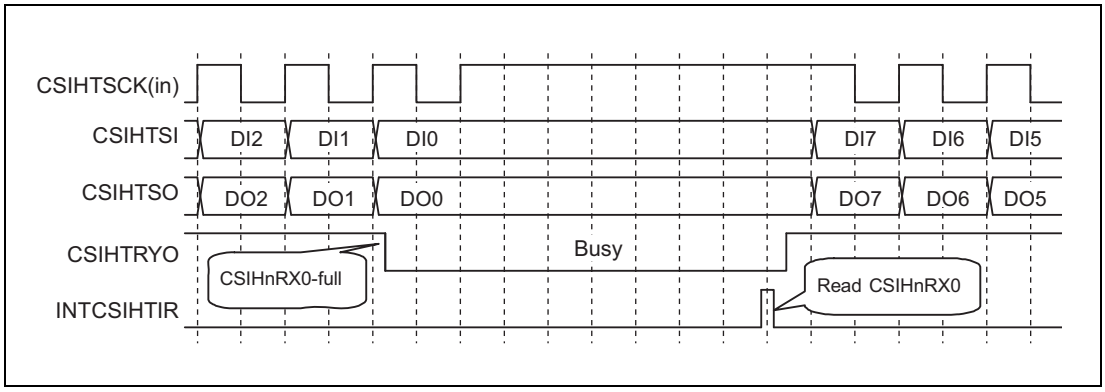


Figure 15.32 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 1)

15.5.11.2 Master Mode

When the master detects CSIHTRYI = 0, the following transfer is put on hold, and the master goes into wait status. It suspends the CSIHTSCK clock.

The CSIHTRYI level is checked at each half clock cycle of CSIHTSCK.

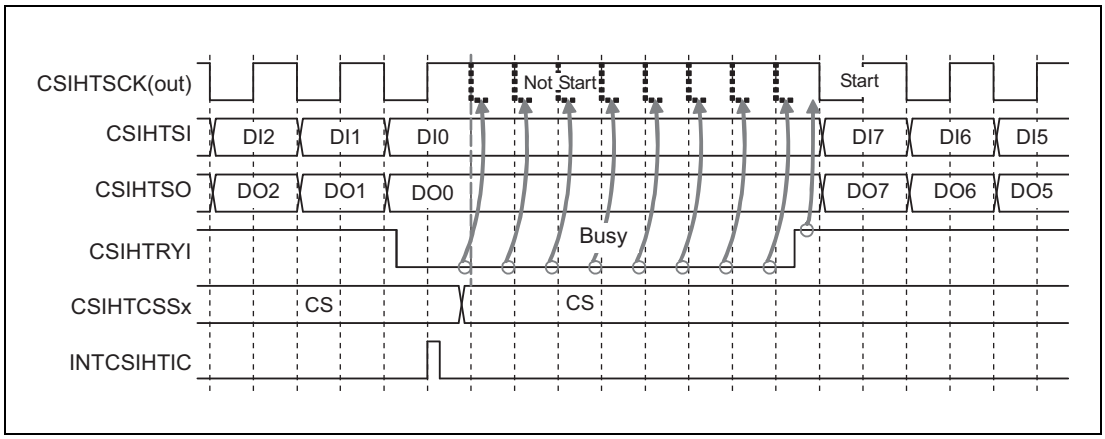


Figure 15.33 Master's Reaction on CSIHTRYI (CSIHnCFGx.CSIHnDAPx = 0)

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").

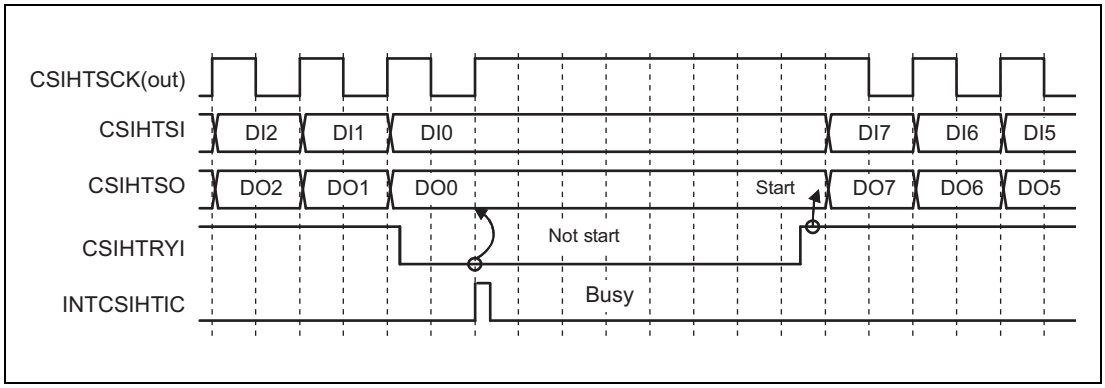


Figure 15.34 Master's Reaction on CSIHTRYI (CSIHnCFGx.CSIHnDAPx = 1)

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 made to wait but continues until the data transfer is completed.**

15.5.12 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 one of these errors is detected, the interrupt request, INTCSIHTIRE is generated and the corresponding flags are set.

15.5.12.1 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 bit CSIHnCTL1.CSIHnDCS (when checking data consistency, make sure that PIPCn.PIPCn_m = 1). It is not active 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 at CSIHTSO are read back via the CSIHTDCS signal into an own 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:

- Interrupt INTCSIHTIRE is generated.
- Bit CSIHnSTR0.CSIHnDCE 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.

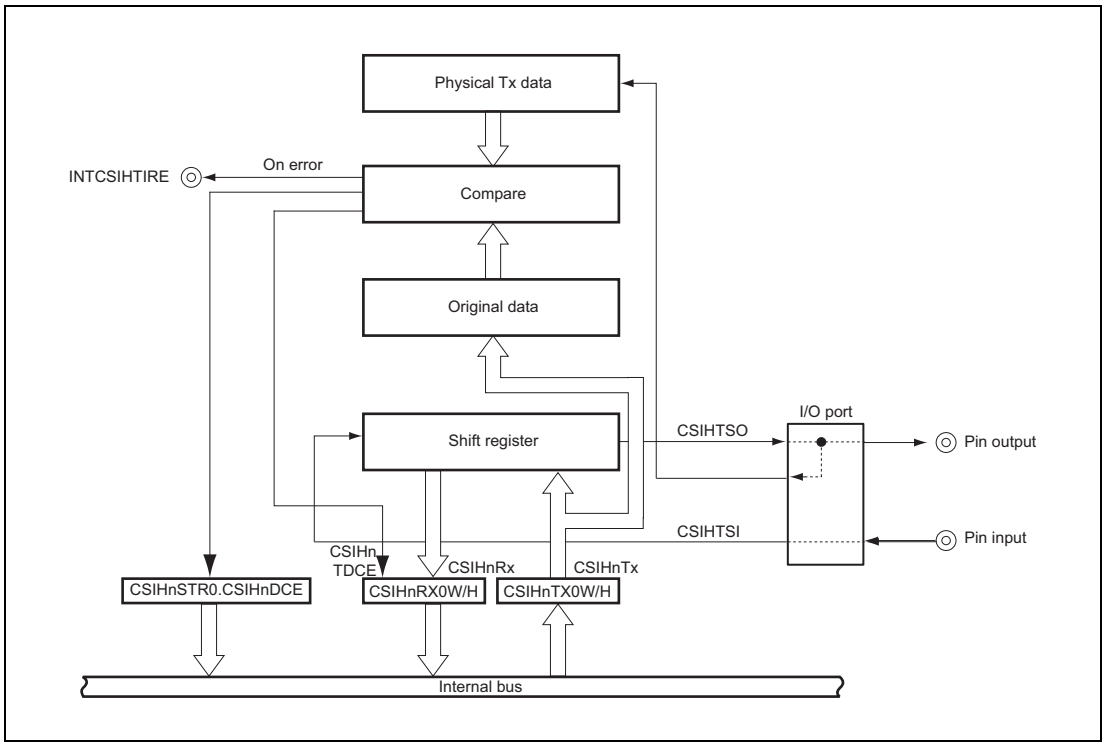


Figure 15.35 Data Consistency Check Functional Block Diagram

15.5.12.2 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. In case of parity error:

- Interrupt INTCSIHTIRE is generated.
- Bit CSIHnSTR0.CSIHnPE 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.

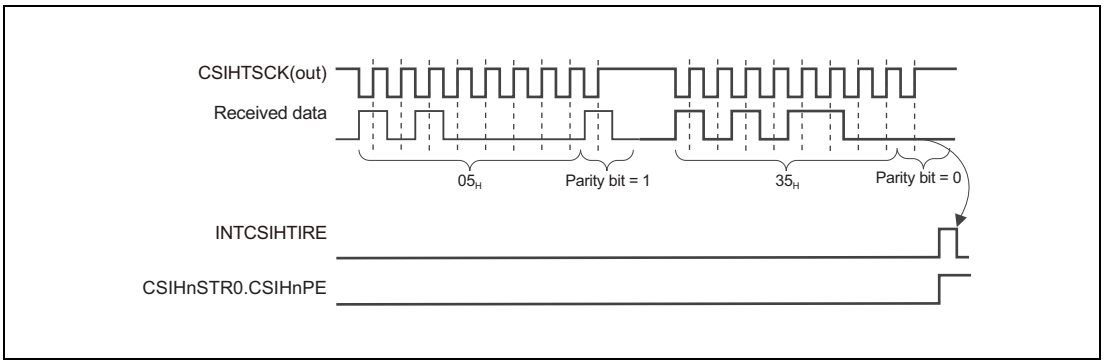


Figure 15.36 Parity Check Example

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.

15.5.12.3 Time-Out Error

Time-out errors can be checked only in slave FIFO mode.

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 is defined in CSIHnMCTL0.CSIHnTO[4:0] in multiples of 8 times the 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).

A dedicated time-out counter measures the time between the last and the next read operation.

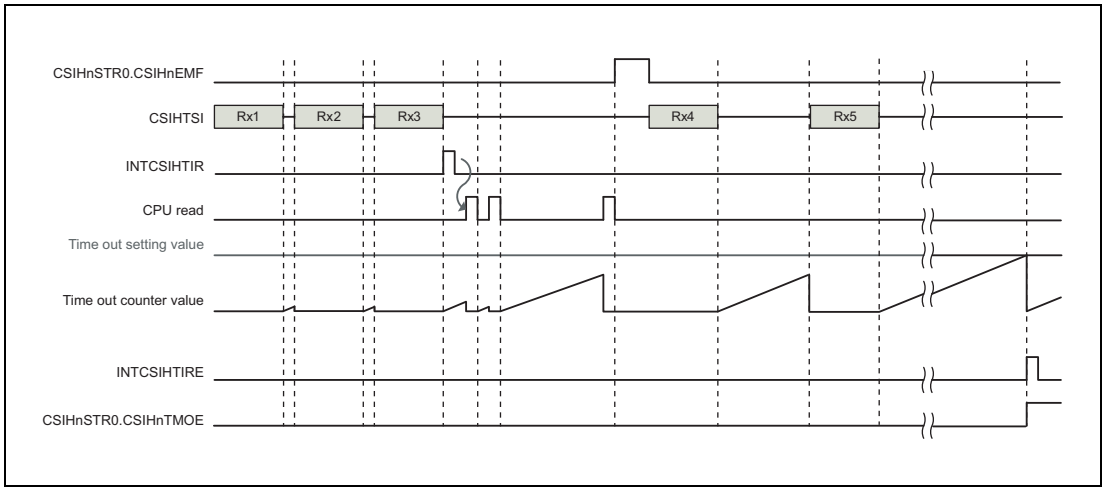


Figure 15.37 Time-Out Check Functional Timing Diagram

The start timing of the time-out counter is as follows:

- When reception is completed,
- When data read from the CPU completes,
  (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 bit 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 the latter.

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 occurs:

- Interrupt INTCSIHTIRE is generated.
- Bit CSIHnSTR0.CSIHnTMOE is set.

15.5.12.4 Overflow Error

An overflow error can happen 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, 90 are still in the FIFO.

In this case, only 38 cells are available for new transmission data packets. When the CPU tries to write the 39th data, an overflow error happens.

This is illustrated in the following figure.

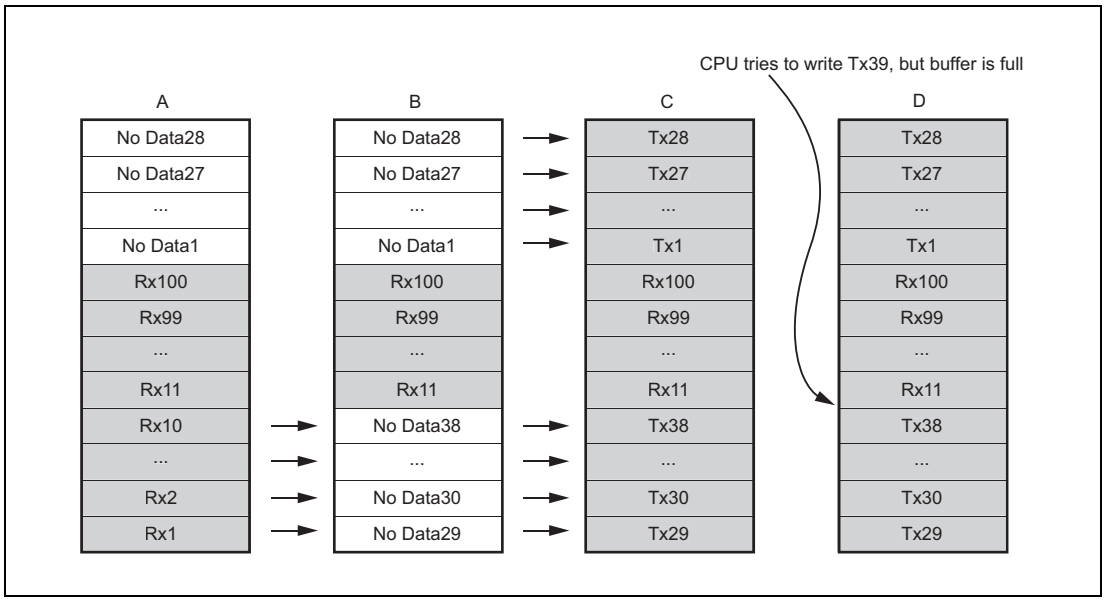


Figure 15.38 FIFO Overview

The 39th and subsequent data are discarded. The figure below shows the overflow timing.

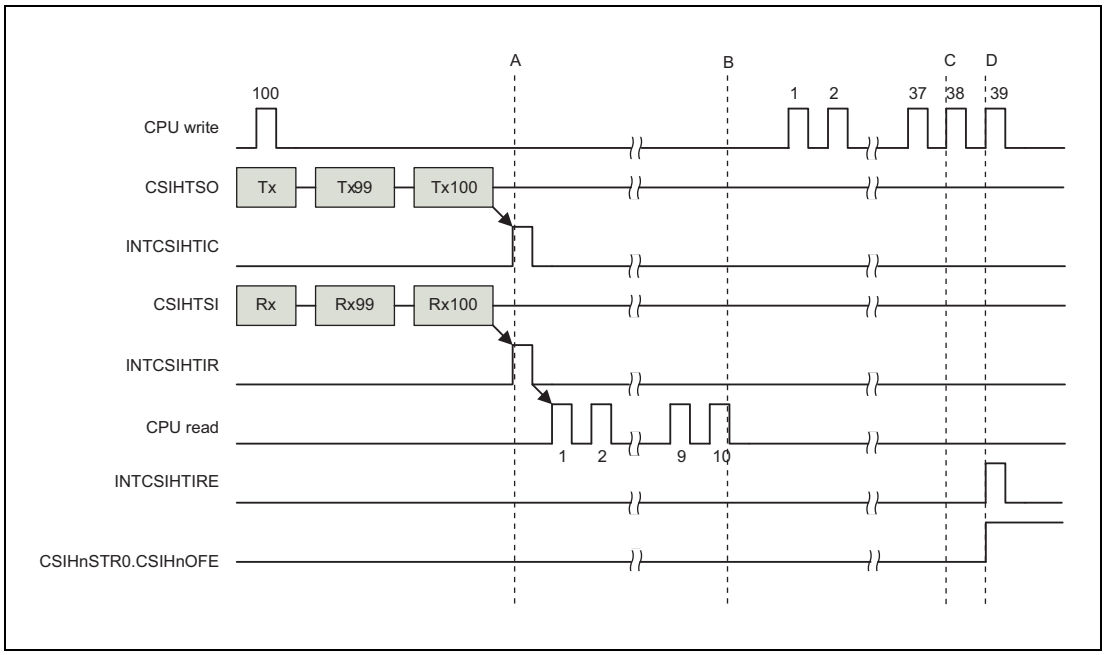


Figure 15.39 FIFO Overflow Timing

In case of overflow error:

- Interrupt INTCSIHTIRE is generated.
- Bit CSIHnSTR0.CSIHnOFE is set.

15.5.12.5 Overrun Error

An overrun error can happen in direct access, transmit-only buffer, and FIFO modes. It cannot happen in dual buffer mode. The overrun error is not generated if data reception is disabled (CSIHnCTL0.CSIHnRXE = 0).

There are two conditions for overrun errors.

Condition for errors 1

- In FIFO mode, while the number of received data is 0 and CPU reads the CSIHnRX0W/H register

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 previous received data is remained in the CSIHnRX0W/H register.
  - In FIFO mode, when FIFO buffer completes receiving data in the full state, an interrupt is generated, and communication continues.

(1) Direct access/transmit-only buffer

In direct access and transmit-only buffer mode, 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 previous reception data.

The following figure illustrates the function.

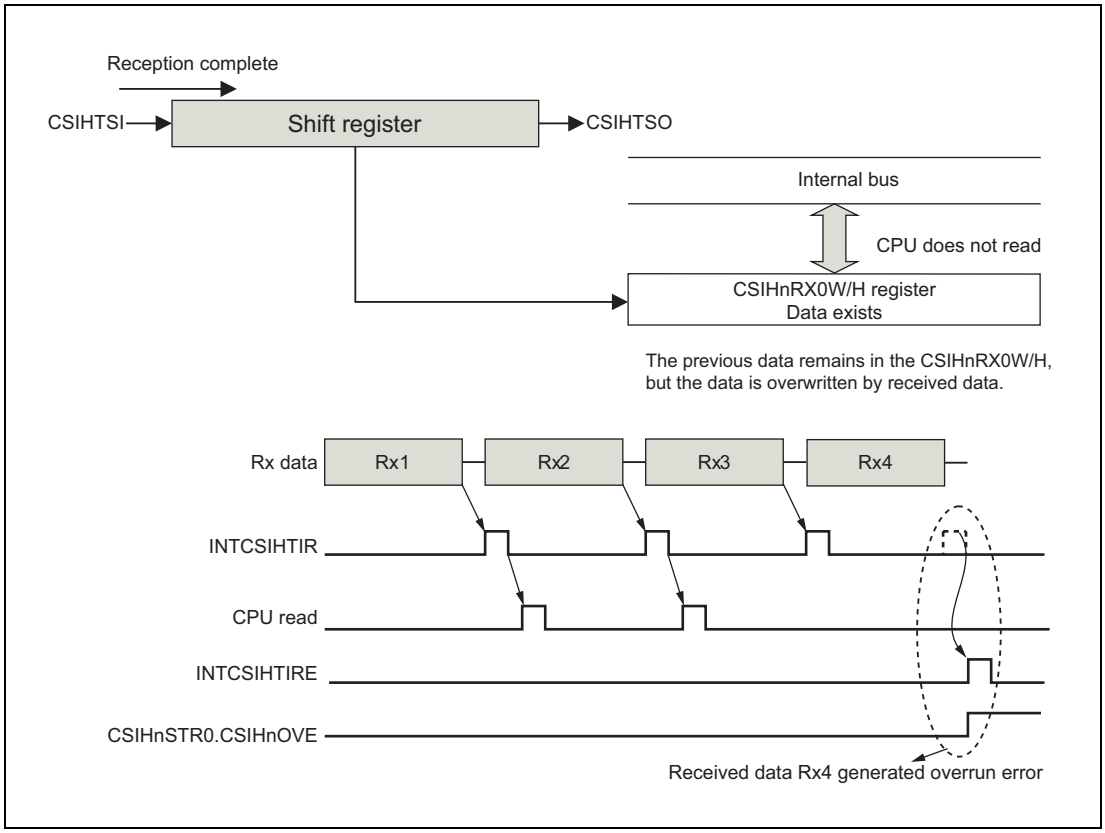


Figure 15.40 Overrun Error Detection in Direct Access and Transmit-Only Buffer Mode

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 has read its reception register and is 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.

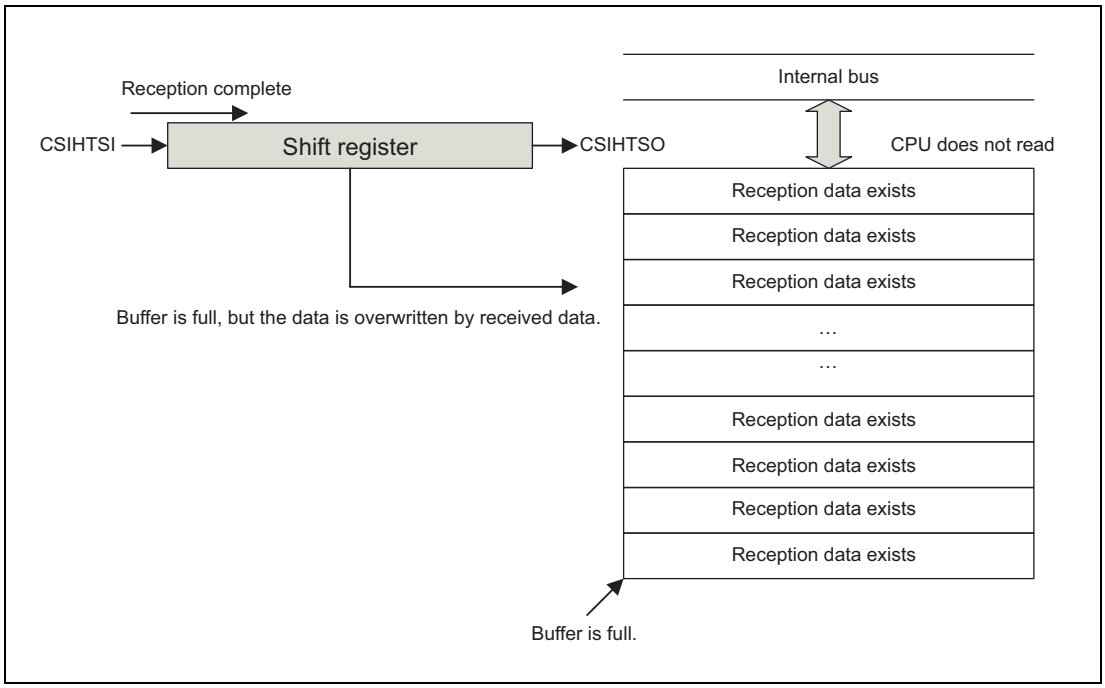


Figure 15.41 Overrun Error Detection in FIFO Mode (FIFO Full)

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 has read its reception register and is ready again.

2. The CPU attempts to read non existing reception data.

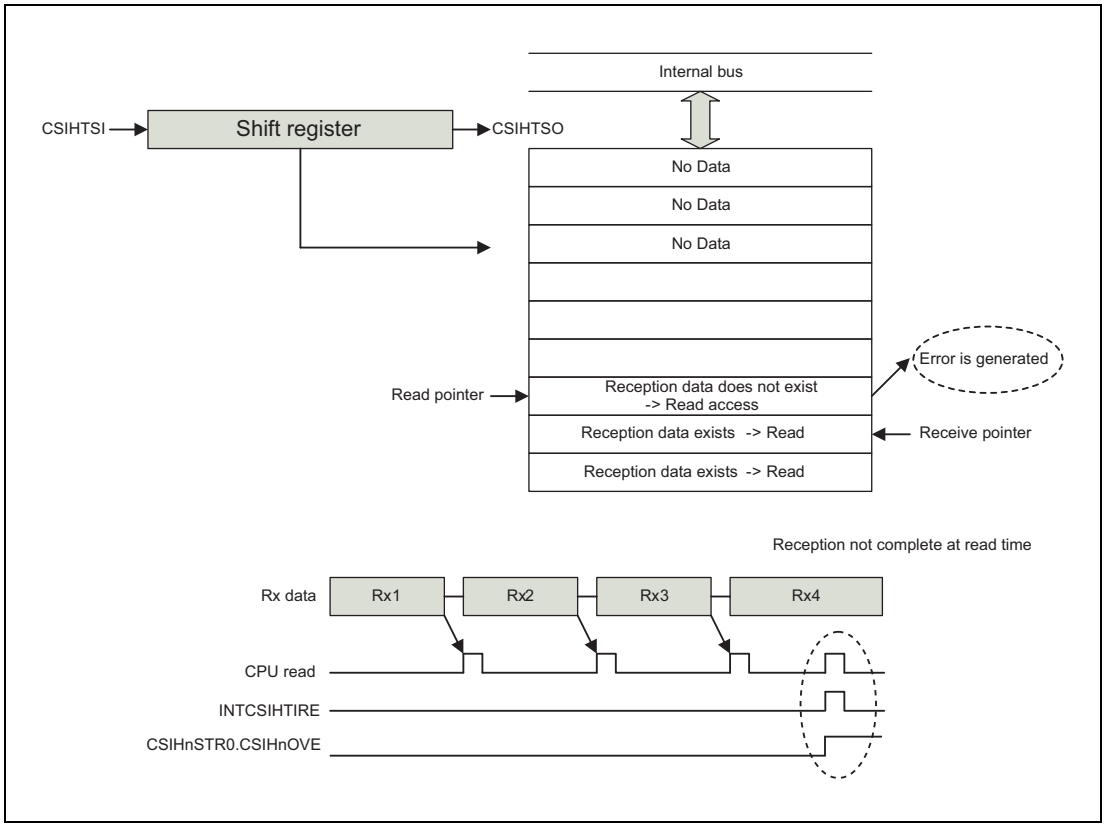


Figure 15.42 Overrun Error Detection in FIFO Mode (No Data)

In case of overrun error:

- Interrupt INTCSIHTIRE is generated.
- Bit CSIHnSTR0.CSIHnOVE is set.
- Received data is overwritten and the communication continues.
  (When the CPU tries to read non-existent data, the CPU starts reading again after a wait until reception is completed.)

For details see **Section 15.5.11, Handshake Function**.

15.5.13 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), 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 15.44 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)

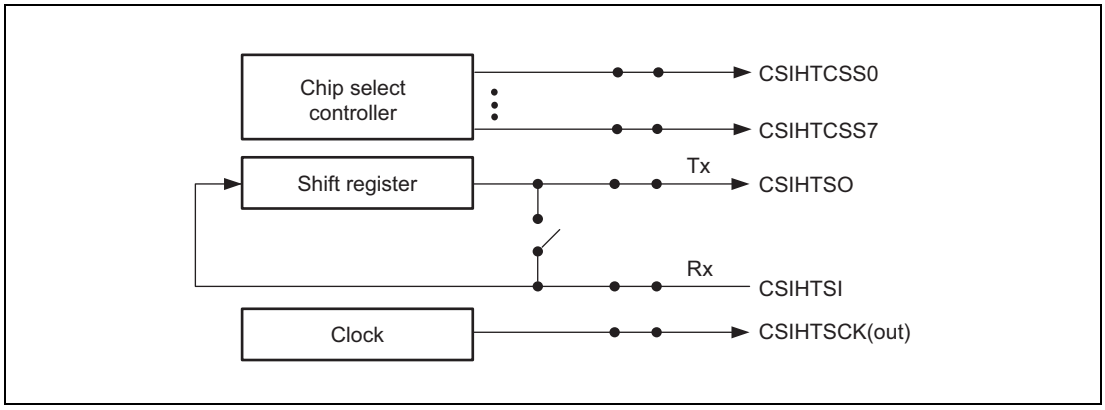


Figure 15.43 Normal Operation

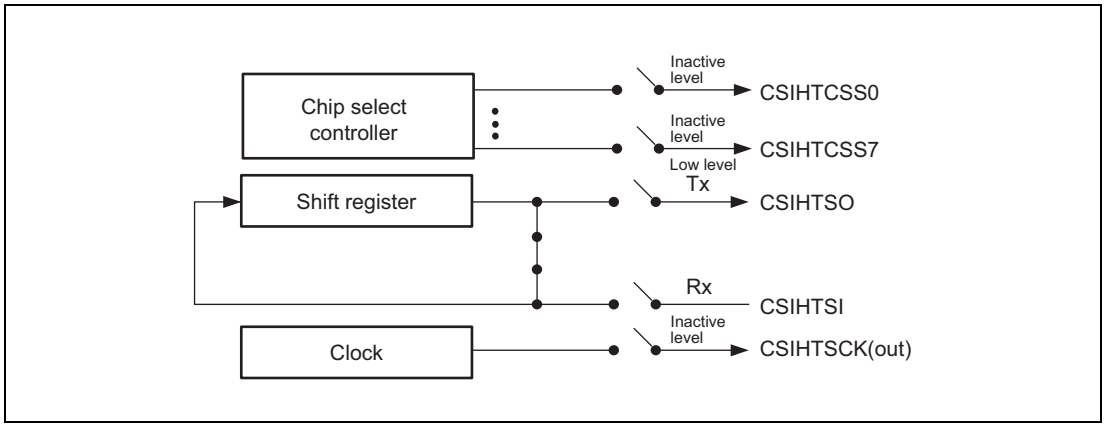


Figure 15.44 Loop-Back Mode Operation

15.5.14 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 direct access mode as high-priority communication only. To enable this function, CSIHnCTL1.CSIHnPHE and CSIHnCTL1.CSIHnJE must be set to 1.

The following figure illustrates CPU-controlled high-priority communication.

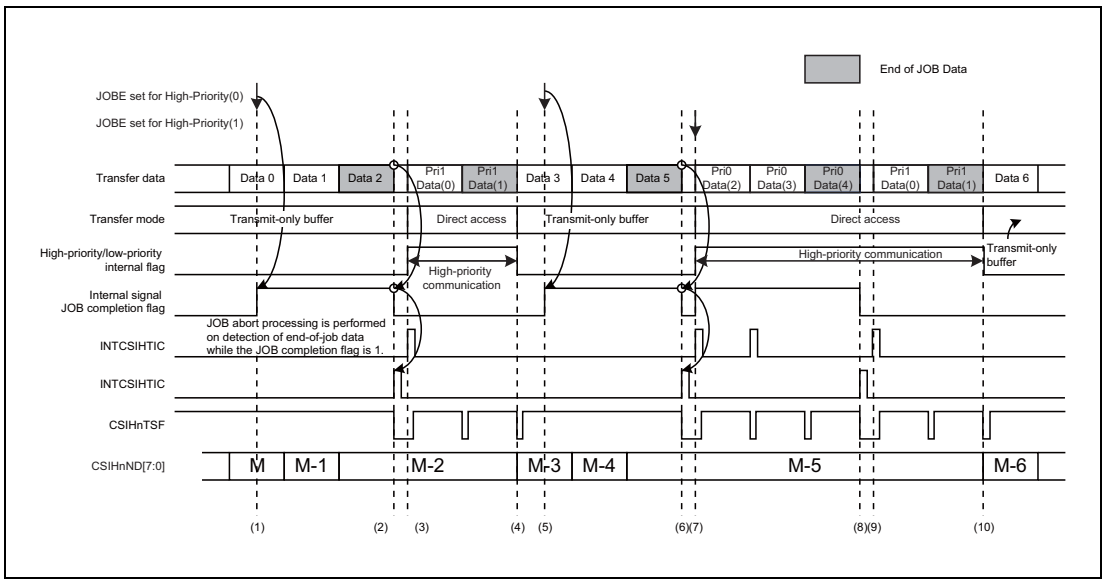


Figure 15.45 Example of CPU-Controlled High-Priority Communication

(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. An internal signal, the JOB completion flag is cleared due to the abortion of communication, 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 switches memory mode to transmit-only buffer mode automatically, 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 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 (switch from transmit-only buffer mode to direct access mode) and from high priority to low priority (switch from direct access mode to transmit-only buffer mode).

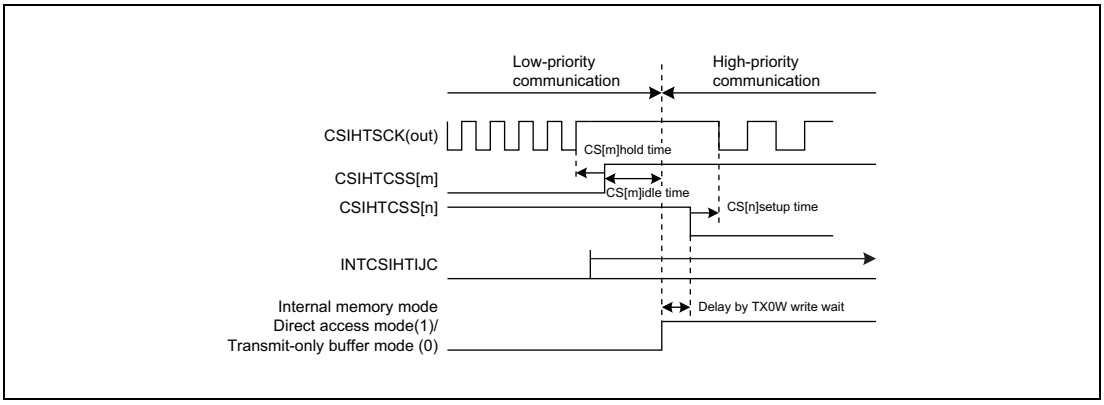


Figure 15.46 Transition from Low-Priority Mode to High-Priority Mode

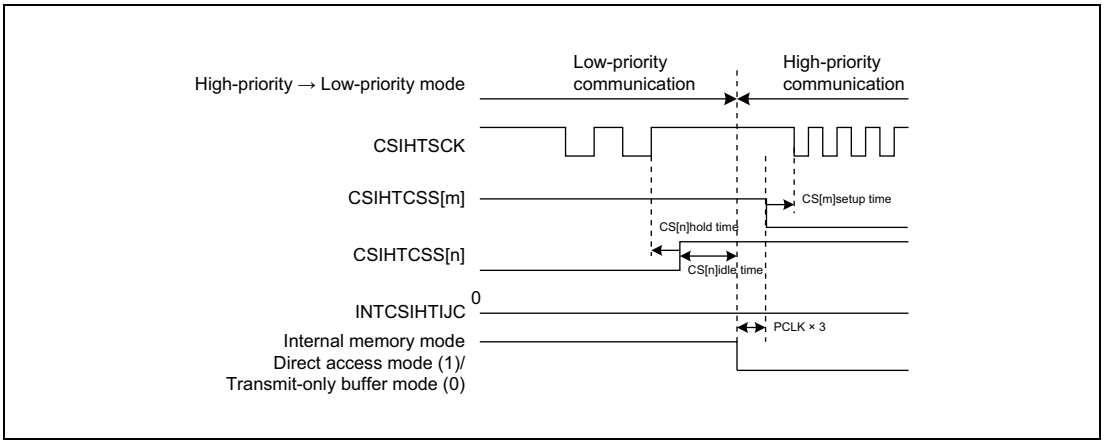


Figure 15.47 Transition from High-Priority Mode to Low-Priority Mode

Do not conduct write operation of communication data or CSIHnCTL0.CSIHnJOBE bit operation during setting prohibit 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 CSIHTIJC interrupt is detected.

During high communication mode period, there is no setting prohibit 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 internal detection timing of CSIHnJOBE bit setting. When CSIHnJOBE bit setting is detected before the last bit communication, 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.

15.5.15 Enforced Chip Select Idle Setting

This macro is able to insert an idle state between the two consecutive transfer data by the setting of CSIHnCFGx.CSIHnIDLx. Detail is as follows.

1. When CSIHnCFGx.CSIHnIDLx = 0
   If a next CSIHTCSSx is the same as the previous one, an idle state is not inserted and an inter-data time is inserted.
   If a 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 a next CSIHTCSSx is not different from the previous one.

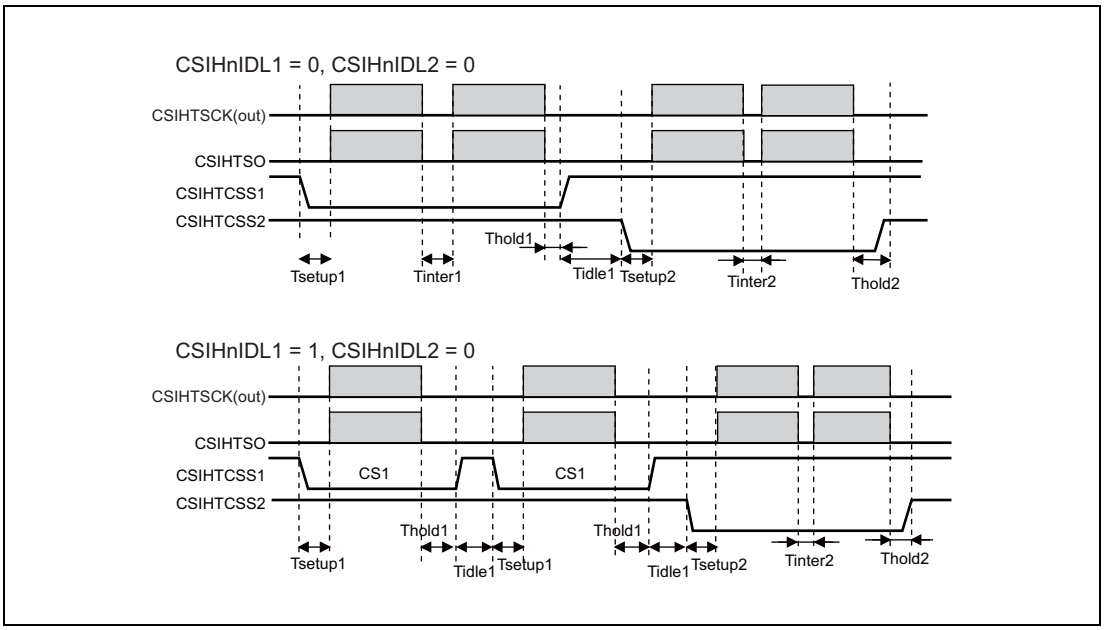


Figure 15.48 Enforced Chip Select Idle Example

CAUTION

If high priority communication function controlled by CPU is validated (CSIHnCTL1.CSIHnPHE = 1), when switch from low priority communication mode to high priority communication mode or switch from high priority communication mode to low priority communication mode, IDLE state is inserted regardless of the setting of CSIHnCFGx.CSIHnIDLx bit.

15.6 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

15.6.1 Procedures in Direct Access Mode

Two examples for a master are provided, one with job mode disabled, and the other one with job mode enabled.

15.6.1.1 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 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)

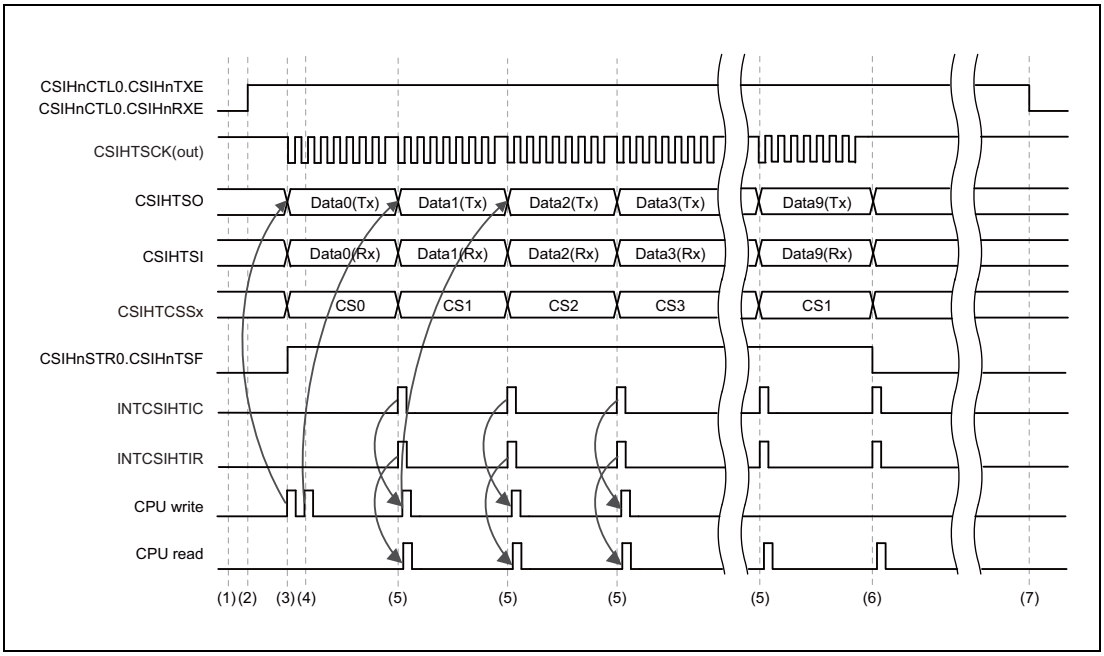


Figure 15.49 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 0

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 bits CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits reception), CSIHnMBS = 1 (Direct access mode selection).
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 transmission of a data 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.

15.6.1.2 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 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 of them sends three data.

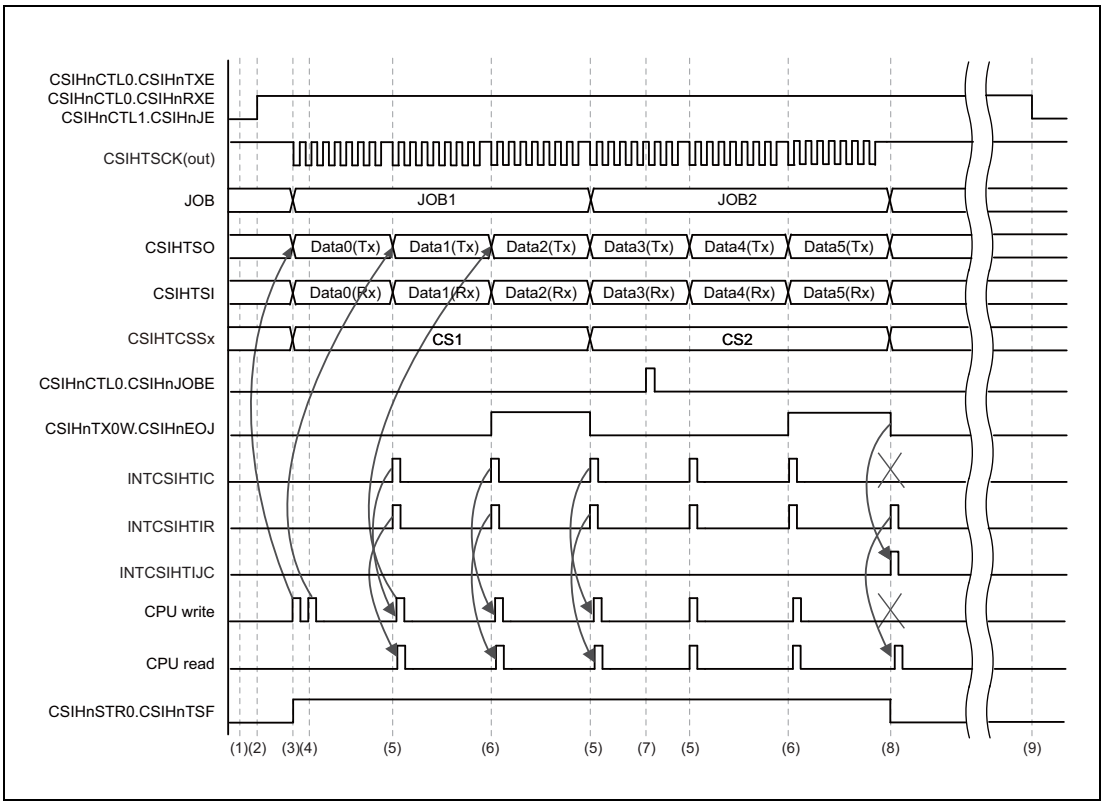


Figure 15.50 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS1 to CS2.
   Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set the bits 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 a forced stop of communication 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.

15.6.2 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.

15.6.2.1 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)

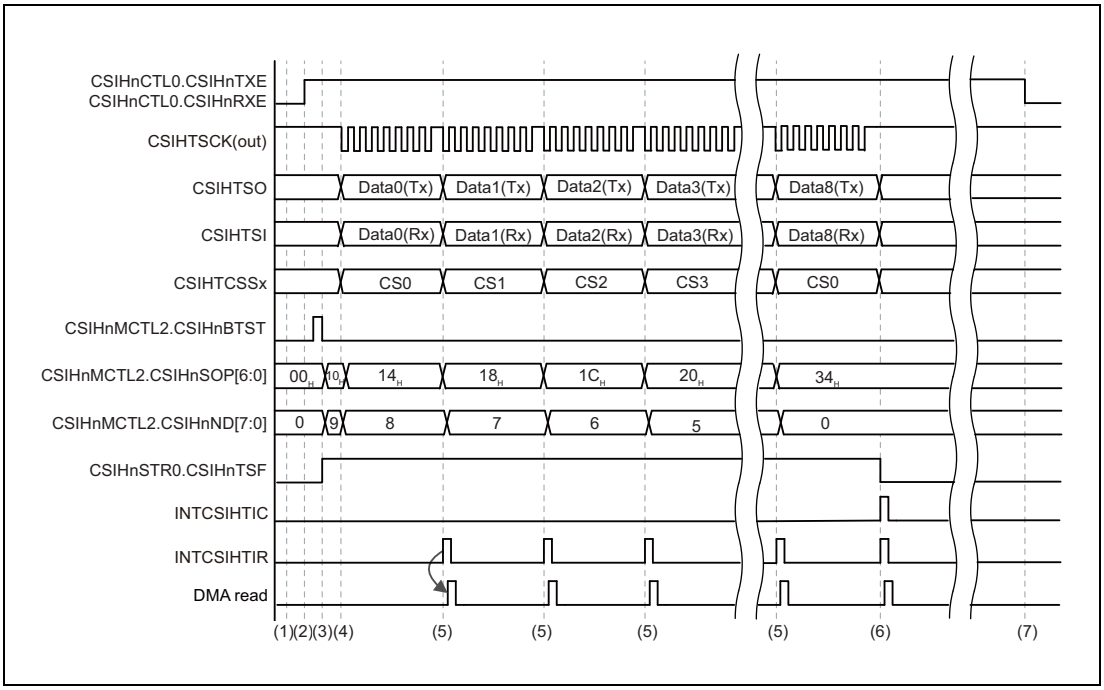


Figure 15.51 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The procedure 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 CS0 to CS3.
   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 the bits 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 bits CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission/reception is started. Bit CSIHnMCTL2.CSIHnSOP[6:0] is automatically incremented and bit CSIHnMCTL2.CSIHnND[7:0] is decremented after each data transmission.
5. After every data reception, the interrupt request, INTCSIHTIR is generated. INTCSIHTIR indicates that the reception 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.

15.6.2.2 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)

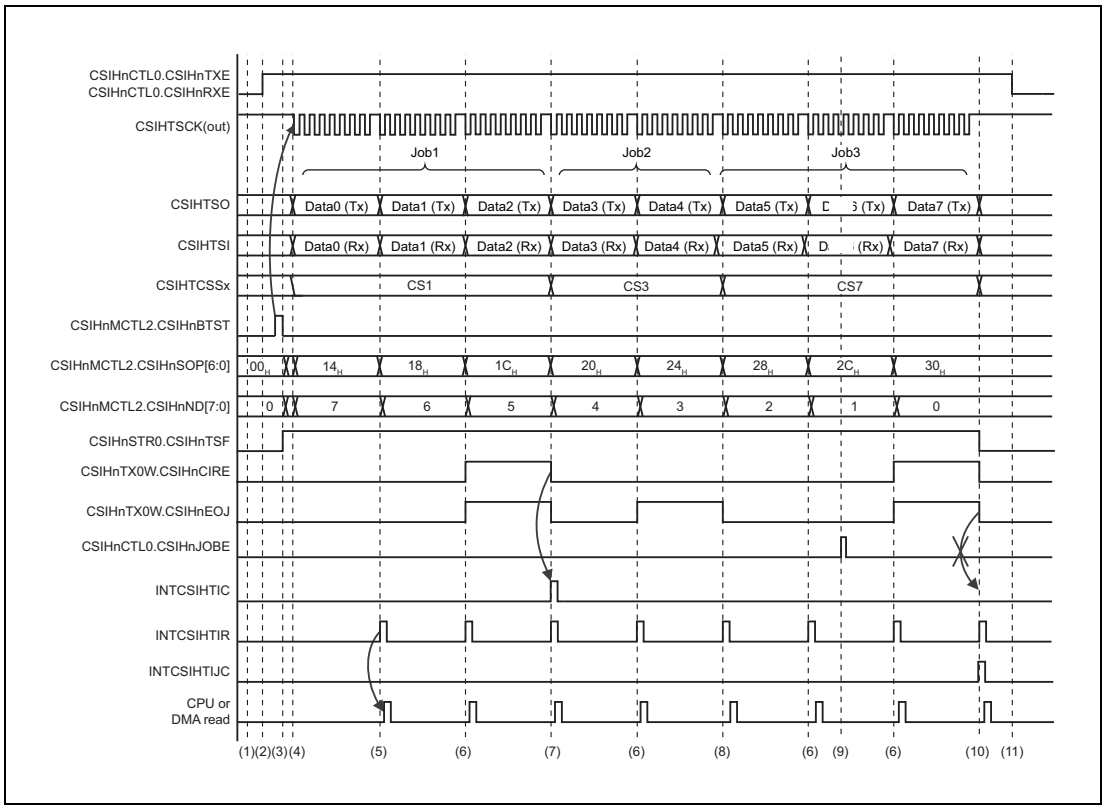


Figure 15.52 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 1

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 bits 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 bits CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. Bit CSIHnMCTL2.CSIHnSOP[6:0] is automatically incremented and bit CSIHnMCTL2.CSIHnND[7:0] is decremented after each data transmission.
5. After every data reception, the interrupt request, INTCSIHTIR is generated. INTCSIHTIR indicates that the reception 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 a forced stop of communication 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.

15.6.3 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.

15.6.3.1 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).
- Default 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)

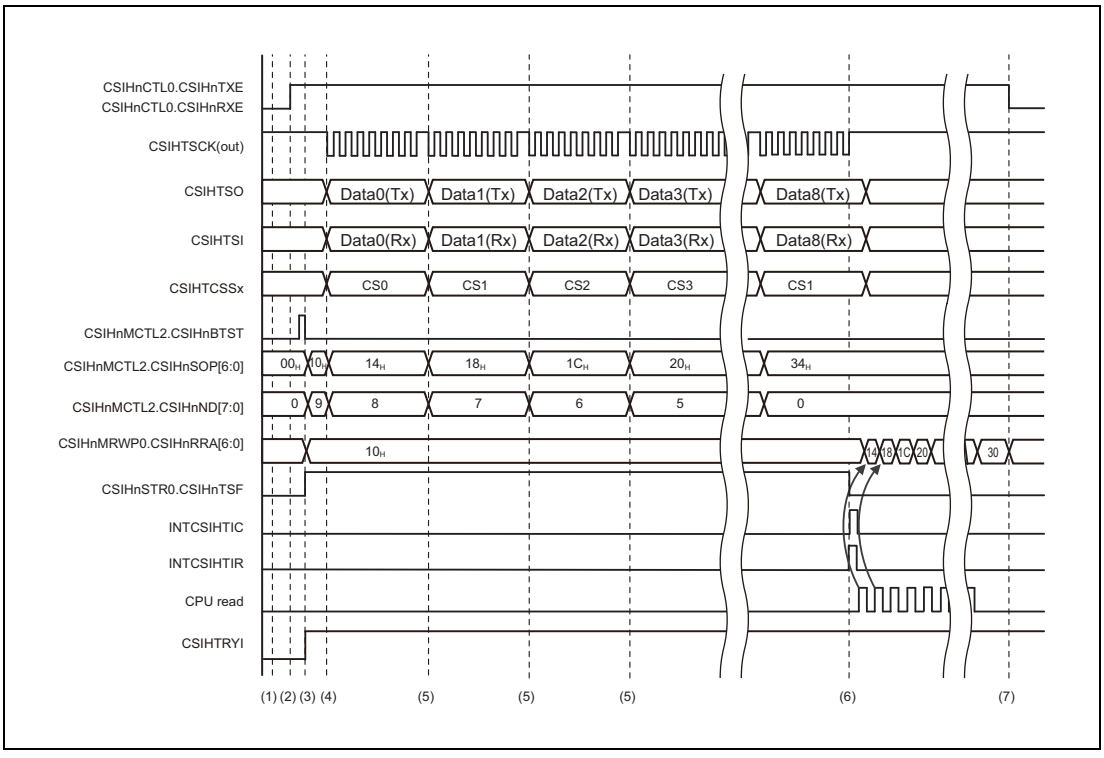


Figure 15.53 Master in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. The example uses chip select signals CS0 to CS3.
   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 bits 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 CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]. Permit buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. Bit CSIHnMCTL2.CSIHnSOP[6:0] is automatically incremented and bit CSIHnMCTL2.CSIHnND[7:0] is 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 adress of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0]. These bits are incremented everytime a 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.

15.6.3.2 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)

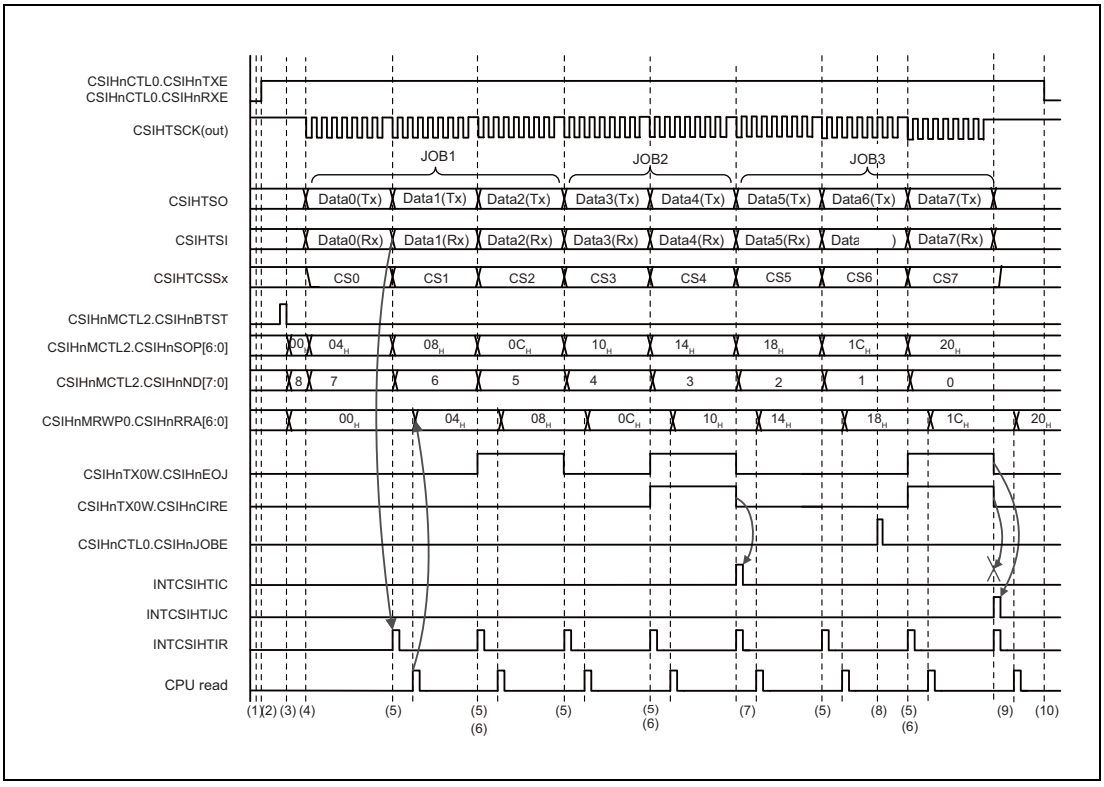


Figure 15.54 Master in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 1

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 bits 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 bits CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started.
   Bit CSIHnMCTL2.CSIHnSOP[6:0] is automatically incremented, and bits CSIHnMCTL2.CSIHnND[7:0] is decremented after each data transmission. This is repeated until the last data is transmitted/received.
5. The INTCSIHTIR interrupt request is generated everytime a 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 a forced stop of communication 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.

15.6.3.3 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 enable (CSIHnCTL1.CSIHnHSE = 1)

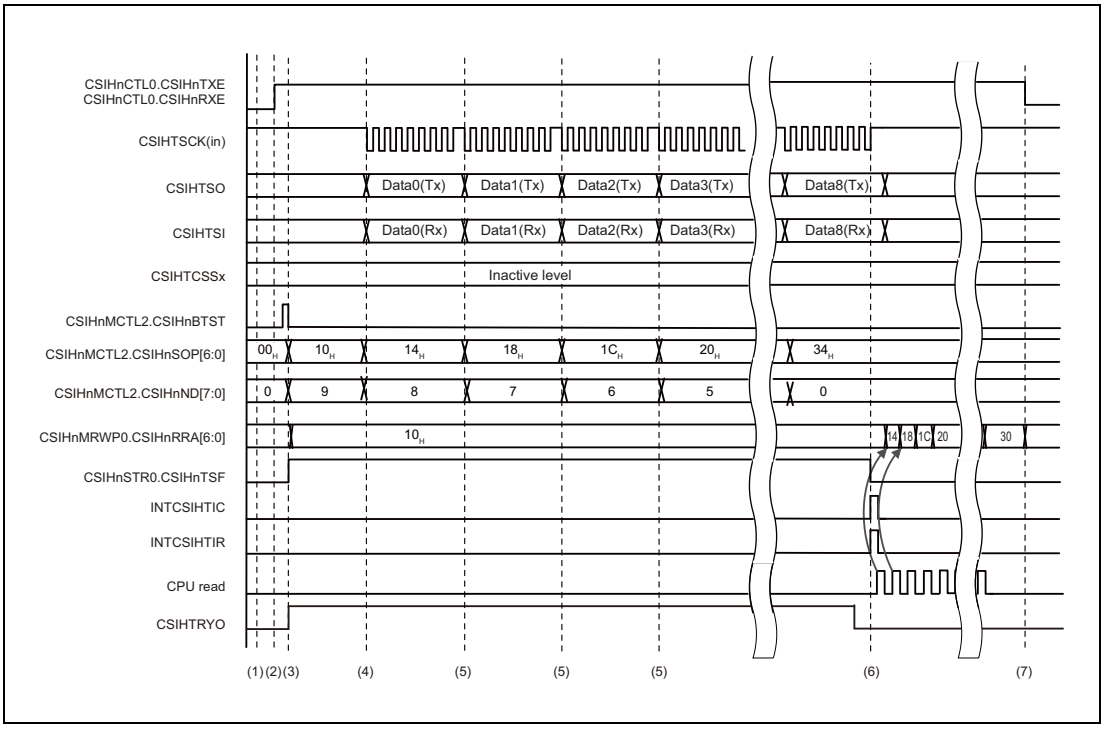


Figure 15.55 Slave in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 0

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 bits 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 CSIHnMCTL2.CSIHnSOP[6:0] and the number of data by setting CSIHnMCTL2.CSIHnND[7:0]. Permit the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started when the input clock from the master is received. Bit CSIHnMCTL2.CSIHnSOP[6:0] is automatically incremented and bit CSIHnMCTL2.CSIHnND[7:0] is 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 adress of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0]. These bits are incremented everytime a 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.

15.6.4 Procedures in FIFO Mode

Two examples for a master is provided, one with job mode disabled, the other one with job mode enabled.

15.6.4.1 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)

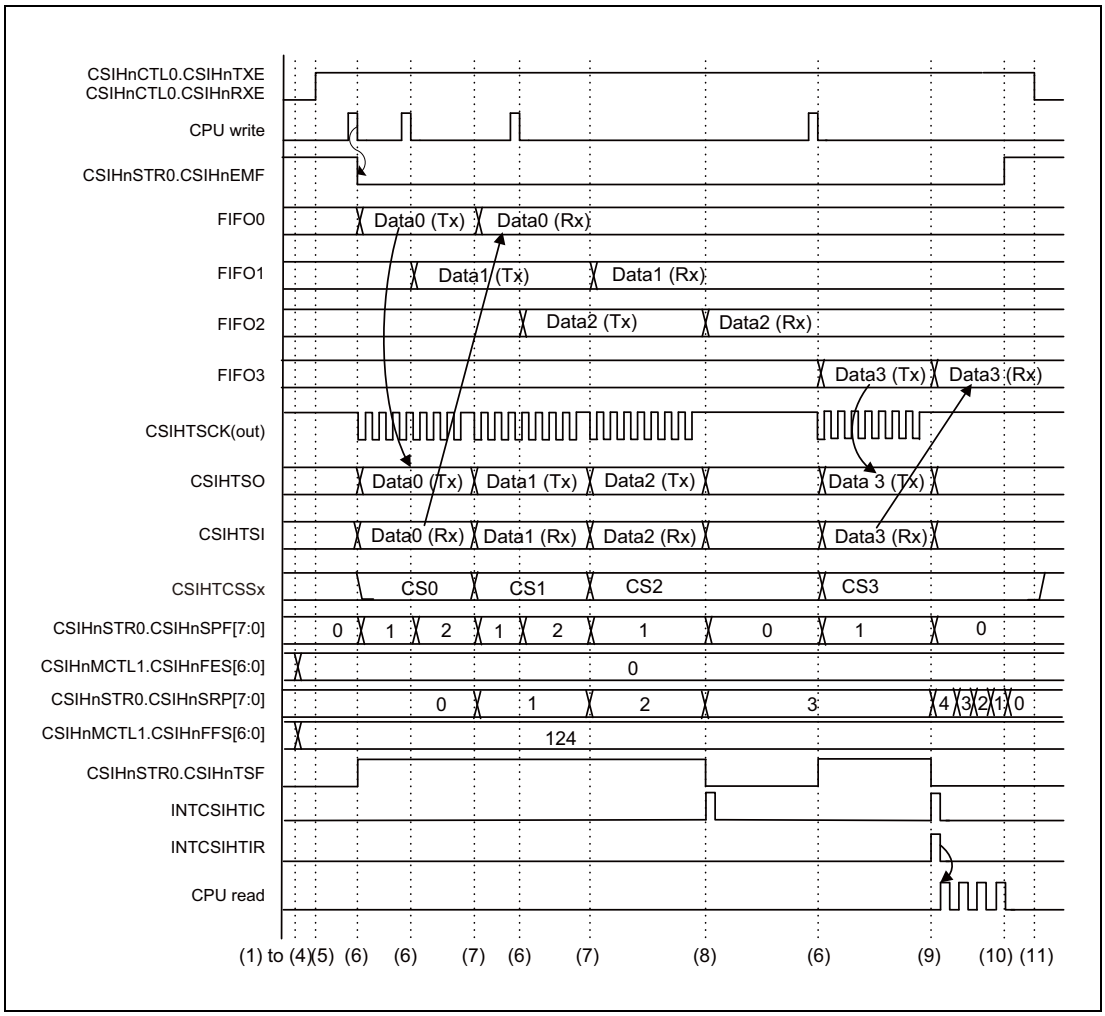


Figure 15.56 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 0

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 CSIHnMCTL0.CSIHnMMS[1:0] = 00B. This example uses chip select signals CS0 to CS3.
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. The CSIHnMCTL1.CSIHnFES [6:0] bits specify the condition for the INTCSIHTIC interrupt output.
   The CSIHnFFS[6:0] bits in the same register specify the condition for the INTCSIHTIR interrupt.
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] is 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 will be 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 minimum the power consumption of CSIHn.

15.6.4.2 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)

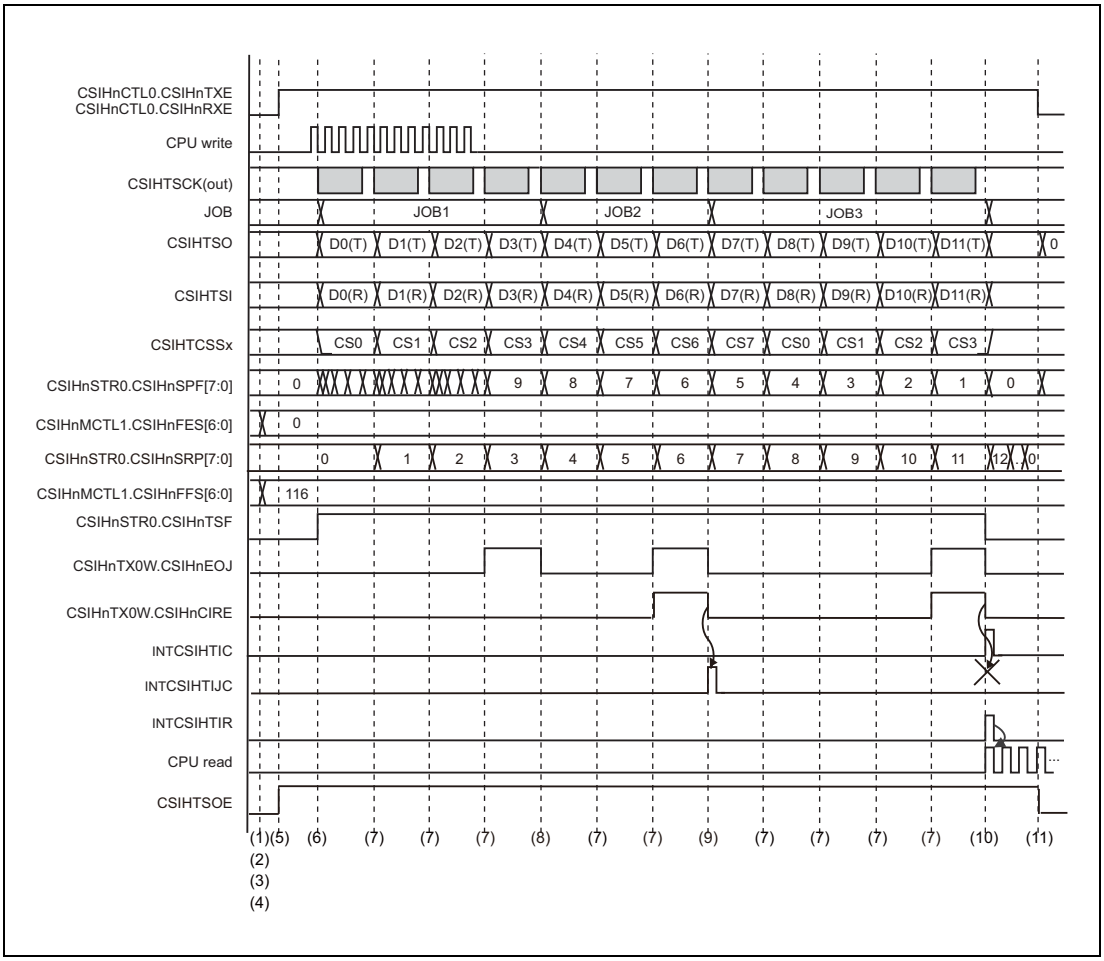


Figure 15.57 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. Set job mode disenable and master mode in the bits corresponding to 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 bit CSIHnSTCR0.CSIHnPCT to 1 to clear all buffer pointers.
3. Make sure 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 bits 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 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] is CSIHnSPF[7:0]. INTCSIHTIC is generated so that INTCSIHTIJC is not generated.
    When CSIHnFFS[6:0] is 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.

15.7 Detection and Correction of Errors in CSIHn RAM

15.7.1 ECC for the CSIHn RAM

Table 15.45 gives an outline of the ECC functions for the CSIH RAM.

Table 15.45 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. With the initial settings, error detection/correction is enabled.
Error notification	When an ECC 2-bti error is generated, the error is notified to the error control module. • 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.

15.7.2 Interrupt Request

Table 15.46 lists the ECC interrupt request of CSIHn RAM.

Table 15.46 CSIHn ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit interrupt signal	Description	Interrupt Number	DMA Trigger Number
INTECCDCSIH0	CSIHn ECC2 Bit error interrupt	7	—
INTECCDCSIH1		8	
INTECCDCSIH2		9	
INTECCDCSIH3		10	

15.7.3 ECCCSIHnCTL — CSIHn ECC Control Register

The ECCCSIHnCTL register controls the mode of the ECC and the status for 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/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 15.47 ECCCSIHnCTL 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 to these bits, 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 to this bit, 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 to this bit, 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 15.47 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 to this bit, 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 (INTECCDCSIHn) 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.

15.7.4 ECCCSIHnTMC — CSIHn ECC Test Mode Control Register

The ECCCSIHnTMC register is used to switch to the test mode, and this register is for test mode control.

This register can be used when CSIHn is not accessed to RAM.

When writing to bit 7, ETMA1 and ETMA0 need to be 10_B.

Access: This register can be read/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 15.48 ECCCSIHnTMC 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 to these bits, 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: ECCCSIHnTED, ECCCSIHnTRC, ECCCSIHnSYND, ECCCSIHnHORD, ECCCSIHnECRD, ECCCSIHnERDB Register test control bits: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. When writing to these bits, 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 which is output to the ECC to be stored in RAM, the ECC data generated for write data or the value of the ECCCSIHnERDB 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 ECCCSIHnERDB Register is stored in RAM.

Table 15.48 ECCCSIHnTMC Register Contents (2/2)

Bit Position	Bit Name	Function
2	ECENS	ECC Encoder Input Selection This bit specifies 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 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 ECCCSIHnTED. 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 ECCCSInTED 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 ECCCSIHnERDB . 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.

15.7.5 ECCCSIHnTED — CSIHn 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 (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 accessed to RAM.

Access: This register can be read/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 15.49 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 for the value of this register.

15.7.6 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 accessed to RAM.

Access: This register can be read/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 15.7.7**)								ECCCSIHnHORD (see **Section 15.7.8**)							
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 15.7.9**)								ECCCSIHnERDB (see **Section 15.7.10**)							
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

15.7.7 ECCCSIHnSYND — CSIHn ECC Decode Syndrome Data Register

This register is a read-only register for storing generated syndrom data in ECC test mode.

Writing to this register is ignored.

This register is read-only 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 can only 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 15.50 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.

15.7.8 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 be accessed only 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 can only 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 15.51 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. When ECCCSIHnTMC.ECTRRS =1 and if ECCCSIHnTED register is read, ECC code is stored.

15.7.9 ECCCSIHnECRD — CSIHnECC 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 (ECCCSIHnTMC.ECTMCE = 1). When ECC test mode is disabled (ECCCSIHnTMC.ECTMCE = 0), 00_H is read.

Access: This register can only 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 15.52 ECCCSIHnECRD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, 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 ECCCSIHnTED register when ECCCSIHnTMC.ECENS =1.

15.7.10 ECCCSIHnERDB — CSIHn 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 (ECCCSIHnTMC.ECTMCE = 1). When ECCCSIHnTMC.ECTMCE = 0, writing to this register is ignored and 00_H is read.

Access: This register can be read/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 15.53 ECCCSIHnERDB Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6 to 0	ERDB[6:0]	These bits can store this register value as ECC data when ECCCSIHnTMC.ECREOS = 1. When the register is read while ECCCSIHnTEC.ECREIS =1, the value read from these bits is ECC data read from the RAM. When ECCCSIHnTMC.ECTRRS = 1, ECC data to be stored in RAM will be read for this register value instated of written data.

15.7.11 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.

Setting 1 to the bit corresponding to each function will enable writing to the read-only bit.

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

Address: FFBC 0600_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCANE7A0	—	—	—	RTCSIHE7A3	RTCSIHE7A2	RTCSIHE7A1	RTCSIHE7A0
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 15.54 SELB_READTEST Register Contents

Bit Position	Bit Name	Function
7	RTCANE7A0	RS-CAN 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-CANn ECC read-only bit can be written).
6 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	RTCSIHE7A3	CSIH3 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH3 ECC read-only bit can be written).
2	RTCSIHE7A2	CSIH2 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH2 ECC read-only bit can be written).
1	RTCSIHE7A1	CSIH1 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH1 ECC read-only bit can be written).
0	RTCSIHE7A0	CSIH0 ECC Register Write Access for Testing Purpose Enable/Disable 0: Write access for testing purpose is disabled. 1: Write access for testing purpose is enabled (CSIH0 ECC read-only bit can be written).

Note 1. For details, see **Section 19.11, Detection and Correction of Errors in RS-CAN RAM**.

Section 16 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/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RLIN2.

16.1 Features of RH850/F1L RLIN2

16.1.1 Number of Units and Channels

This microcontroller has the following number of RLIN2 units and channels.

Table 16.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	1	1	2	4
Name	RLIN24n (n = 0)	RLIN24n (n = 0)	RLIN24n (n = 0)	RLIN24n (n = 0)	RLIN24n (n = 0, 1)	RLIN24n (n = 0, 1) RLIN21n (n = 0, 1)

Table 16.2 Configurations of RLIN2 Units and Correspondence between the Number of Channels per Unit and the Channel Numbers of the Units

Unit Name RLIN24n/ RLIN21n	Channels per Unit	Unit Channel Number (i)	Channel Name RLIN2m	RH850/F1L 48 pins (2 ch)	RH850/F1L 64 pins (2 ch)	RH850/F1L 80 pins (2 ch)	RH850/F1L 100 pins (3 ch)	RH850/F1L 144 pins (6 ch)	RH850/F1L 176 pins (10 ch)
RLIN240	4	0	RLIN20	√	√	√	√	√	√
		1	RLIN21	√	√	√	√	√	√
		2	RLIN22				√	√	√
		3	RLIN23					√	√
RLIN241	4	0	RLIN24					√	√
		1	RLIN25					√	√
		2	RLIN26						√
		3	RLIN27						√
RLIN210	1	0	RLIN28						√
RLIN211	1	0	RLIN29						√

Table 16.3 Index

Index	Meaning
n	Throughout this section, the individual RLIN2 units are identified by the index “n” (n = 0, 1).
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 configure 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, RLIN24nGLWBR / RLIN21nGLWBR are the LIN wake-up baud rate select registers, which are the global registers of RLIN2. RLIN24nmLiMD / RLIN21nmLiMD are the LIN mode registers, which are the channel registers.

The following lists the index corresponding to each product.

Table 16.4 Index Correspondence of Each Product

Index Correspondence of Each Product			
48 pins, 64 pins, 80 pins	**100 pins**	**144 pins**	**176 pins**
i = 0, 1 (RLIN240)	i = 0 to 2 (RLIN240)	i = 0 to 3 (RLIN240) i = 0, 1 (RLIN241)	i = 0 to 3 (RLIN240, 1) i = 0 (RLIN210, 1)
b = 1 to 8	b = 1 to 8	b = 1 to 8	b = 1 to 8

16.1.2 Register Base Address

RLIN2 base addresses are listed in the following table.

RLIN2 register addresses are given as offsets from the base addresses in general.

Table 16.5 Register Base Address

Base Address Name	Base Address
<RLIN240_base>	FFCE 0000$_H$
<RLIN241_base>	FFCE 0080$_H$
<RLIN210_base>	FFCE 0100$_H$
<RLIN211_base>	FFCE 0120$_H$

16.1.3 Clock Supply

The RLIN2 clock supply is shown in the following table.

Table 16.6 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
RLIN24n	LIN communication clock sources	CKSCLK_ILIN
RLIN21n	LIN communication clock sources	CKSCLK_ILIN

16.1.4 Interrupt Request

RLIN2 interrupt requests are listed in the following table.

Table 16.7 Interrupt Request (1/2)

Unit Interrupt Signal	Outline	Interrupt Number
RLIN240		
INTRLIN0	RLIN20 interrupt	50
INTRLIN1	RLIN21 interrupt	51
INTRLIN2	RLIN22 interrupt	154
INTRLIN3	RLIN23 interrupt	155

Table 16.7 Interrupt Request (2/2)

Unit Interrupt Signal	Outline	Interrupt Number
RLIN241		
INTRLIN4	RLIN24 interrupt	218
INTRLIN5	RLIN25 interrupt	219
INTRLIN6	RLIN26 interrupt	267
INTRLIN7	RLIN27 interrupt	268
RLIN210		
INTRLIN8	RLIN28 interrupt	277
RLIN211		
INTRLIN9	RLIN29 interrupt	278

16.1.5 Reset Sources

RLIN2 reset sources are listed in the following table. RLIN2 is initialized by these reset sources.

Table 16.8 Reset Sources

Unit Name	Reset Source
RLIN24n	All reset sources (ISORES)
RLIN21n	All reset sources (ISORES)

16.1.6 External Input/Output Signals

External input/output signals of RLIN2 are listed in below.

Table 16.9 External Input/Output Signals

Unit Signal Name	Outline	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)
RLIN210		
RLIN2mRX (m = 8)	RLIN210 receive data input	RLIN2mRX (m = 8)
RLIN2mTX (m = 8)	RLIN210 transmit data output	RLIN2mTX (m = 8)
RLIN211		
RLIN2mRX (m = 9)	RLIN211 receive data input	RLIN2mRX (m = 9)
RLIN2mTX (m = 9)	RLIN211 transmit data output	RLIN2mTX (m = 9)

16.2 Overview

16.2.1 Functional Overview

The LIN Master Interface is a hardware LIN communication controller that complies to 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 16.10 gives the LIN Master Interface specifications.

Table 16.10 LIN Master Interface Specifications

Item		Specifications
Channel count		10 channels (In this product, 1-channel version and 4-channel version of RLINS 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	• Break (low) transmission width: 13 to 28 Tbits • Break delimiter transmission 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 (for each frame)
	Response field data byte count	Variable from 0 to 8 bytes
	Frame communication modes	• Mode in which header transmission and response transmission/reception is 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	LIN wake-up mode provided • Wake-up transmission (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 (INTRLINm) for each channel.

Note 1. Since the same register is used for setting, the inter-byte space (header) = response space.
Note 2. For wake-up reception, the low level width of the input signal is indicated.

16.2.2 Block Diagram

Figure 16.1 shows a block diagram of the LIN master interface.

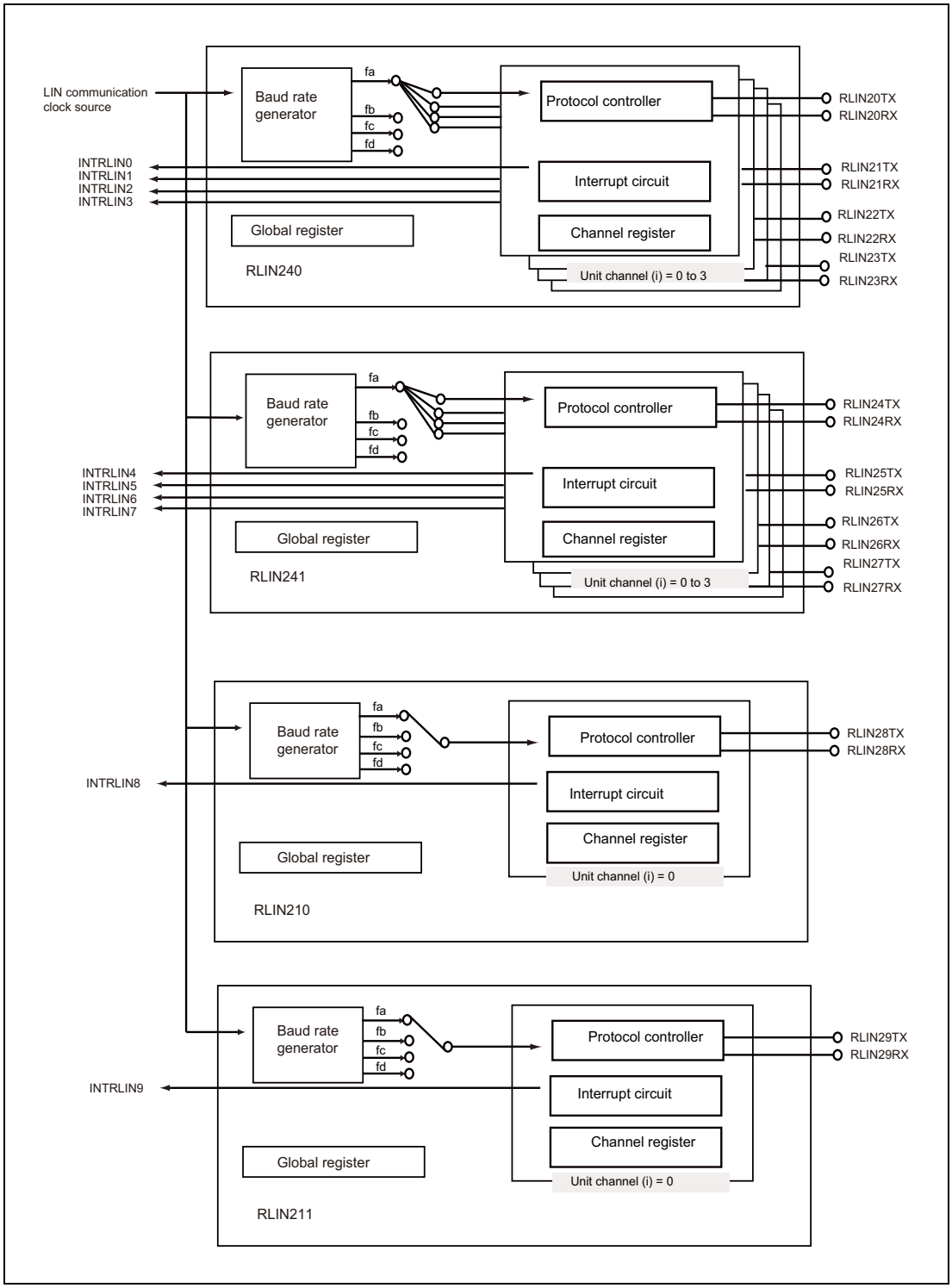


Figure 16.1 LIN Master Interface Block Diagram (176 Pins, RLIN2 10 Channels Embedded)

16.3 Registers

The registers of the LIN master interface are configured with global registers and channel registers. As the global registers are allocated for each unit, they can be set respectively in units. As the channel registers are allocated for each channel, they can be controlled respectively in channels.

16.3.1 List of Registers

RLIN2 registers are listed in the following table.

For details about <RLIN24n_base> and <RLIN21n_base>, see **Section 16.1.2, Register Base Address**.

Table 16.11 Registers (1/2)

Module	Register	Symbol	Address
Global registers			
RLN24n	LIN wake-up baud rate select register	RLN24nGLWBR	<RLIN24n_base> + 01_H
RLN21n		RLN21nGLWBR	<RLIN21n_base> + 01_H
RLN24n	LIN baud rate prescaler 0 register	RLN24nGLBRP0	<RLIN24n_base> + 02_H
RLN21n		RLN21nGLBRP0	<RLIN21n_base> + 02_H
RLN24n	LIN baud rate prescaler 1 register	RLN24nGLBRP1	<RLIN24n_base> + 03_H
RLN21n		RLN21nGLBRP1	<RLIN21n_base> + 03_H
RLN24n	LIN self test control register	RLN24nGLSTC	<RLIN24n_base> + 04_H
RLN21n		RLN21nGLSTC	<RLIN21n_base> + 04_H
Channel registers			
RLN24nm	LIN mode register	RLN24nmLiMD	<RLIN24n_base> + 08_H + i × 20_H
RLN21nm		RLN21nmLiMD	<RLIN21n_base> + 08_H+ i × 20_H
RLN24nm	LIN break field configuration register	RLN24nmLiBFC	<RLIN24n_base> + 09_H+ i × 20_H
RLN21nm		RLN21nmLiBFC	<RLIN21n_base> + 09_H+ I × 20_H
RLN24nm	LIN space configuration register	RLN24nmLiSC	<RLIN24n_base> + $0A_H$ + i × 20_H
RLN21nm		RLN21nmLiSC	<RLIN21n_base> + $0A_H$ + i × 20_H
RLN24nm	LIN wake-up configuration register	RLN24nmLiWUP	<RLIN24n_base> + $0B_H$ + i × 20_H
RLN21nm		RLN21nmLiWUP	<RLIN21n_base> + $0B_H$ + i × 20_H
RLN24nm	LIN interrupt enable register	RLN24nmLiIE	<RLIN24n_base> + $0C_H$ + i × 20_H
RLN21nm		RLN21nmLiIE	<RLIN21n_base> + $0C_H$ + i × 20_H
RLN24nm	LIN error detection enable register	RLN24nmLiEDE	<RLIN24n_base> + $0D_H$ + i × 20_H
RLN21nm		RLN21nmLiEDE	<RLIN21n_base> + $0D_H$ + i × 20_H
RLN24nm	LIN control register	RLN24nmLiCUC	<RLIN24n_base> + $0E_H$ + i × 20_H
RLN21nm		RLN21nmLiCUC	<RLIN21n_base> + $0E_H$ + i × 20_H
RLN24nm	LIN transmission control register	RLN24nmLiTRC	<RLIN24n_base> + 10_H + i × 20_H
RLN21nm		RLN21nmLiTRC	<RLIN21n_base> + 10_H + i × 20_H
RLN24nm	LIN mode status register	RLN24nmLiMST	<RLIN24n_base> + 11_H + i × 20_H
RLN21nm		RLN21nmLiMST	<RLIN21n_base> + 11_H + i × 20_H
RLN24nm	LIN status register	RLN24nmLiST	<RLIN24n_base> + 12_H + i × 20_H
RLN21nm		RLN21nmLiST	<RLIN21n_base> + 12_H + i × 20_H
RLN24nm	LIN error status register	RLN24nmLiEST	<RLIN24n_base> + 13_H + i × 20_H
RLN21nm		RLN21nmLiEST	<RLIN21n_base> + 13_H + i × 20_H

Table 16.11 Registers (2/2)

Module	Register	Symbol	Address
RLN24nm	LIN data field configuration register	RLN24nmLiDFC	<RLIN24n_base> + 14_H + i × 20_H
RLN21nm		RLN21nmLiDFC	<RLIN21n_base> + 14_H + i × 20_H
RLN24nm	LIN ID buffer register	RLN24nmLiIDB	<RLIN24n_base> + 15_H + i × 20_H
RLN21nm		RLN21nmLiIDB	<RLIN21n_base> + 15_H + i × 20_H
RLN24nm	LIN check sum buffer register	RLN24nmLiCBR	<RLIN24n_base> + 16_H + i × 20_H
RLN21nm		RLN21nmLiCBR	<RLIN21n_base> + 16_H + i × 20_H
RLN24nm	LIN data buffer 1 register	RLN24nmLiDBR1	<RLIN24n_base> + 18_H + i × 20_H
RLN21nm		RLN21nmLiDBR1	<RLIN21n_base> + 18_H + i × 20_H
RLN24nm	LIN data buffer 2 register	RLN24nmLiDBR2	<RLIN24n_base> + 19_H + i × 20_H
RLN21nm		RLN21nmLiDBR2	<RLIN21n_base> + 19_H + i × 20_H
RLN24nm	LIN data buffer 3 register	RLN24nmLiDBR3	<RLIN24n_base> + $1A_H$ + i × 20_H
RLN21nm		RLN21nmLiDBR3	<RLIN21n_base> + $1A_H$ + i × 20_H
RLN24nm	LIN data buffer 4 register	RLN24nmLiDBR4	<RLIN24n_base> + $1B_H$ + i × 20_H
RLN21nm		RLN21nmLiDBR4	<RLIN21n_base> + $1B_H$ + i × 20_H
RLN24nm	LIN data buffer 5 register	RLN24nmLiDBR5	<RLIN24n_base> + $1C_H$ + i × 20_H
RLN21nm		RLN21nmLiDBR5	<RLIN21n_base> + $1C_H$ + i × 20_H
RLN24nm	LIN data buffer 6 register	RLN24nmLiDBR6	<RLIN24n_base> + $1D_H$ + i × 20_H
RLN21nm		RLN21nmLiDBR6	<RLIN21n_base> + $1D_H$ + i × 20_H
RLN24nm	LIN data buffer 7 register	RLN24nmLiDBR7	<RLIN24n_base> + $1E_H$ + i × 20_H
RLN21nm		RLN21nmLiDBR7	<RLIN21n_base> + $1E_H$ + i × 20_H
RLN24nm	LIN data buffer 8 register	RLN24nmLiDBR8	<RLIN24n_base> + $1F_H$ + i × 20_H
RLN21nm		RLN21nmLiDBR8	<RLIN21n_base> + $1F_H$ + i × 20_H

Note: When writing to a register not used, write the value after reset.

16.3.2 Global Registers

16.3.2.1 RLN24nGLWBR / RLN21nGLWBR — LIN Wake-Up Baud Rate Select Register

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

Address: RLN24nGLWBR: <RLIN24n_base> + 01_H
RLN21nGLWBR:<RLIN21n_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 16.12 RLN24nGLWBR / RLN21nGLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. The write value should always be the value after reset.
0	LWBR0	Wake-up Baud Rate Select 0: In LIN wake-up mode, the clock specified by the LCKS bit setting in the RLN24nmLiMD / RLN21nmLiMD registers is used. (when LIN1.3 is used) 1: In LIN wake-up mode, the clock fa is used regardless of the setting of the LCKS bit in the RLN24nmLiMD / RLN21nmLiMD registers.(When LIN2.x is used)

Set the RLN24nGLWBR / RLN21nGLWBR register when all channels in the same unit are in LIN reset mode (while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0).

LWBR0 Bit (Wake-Up Baud Rate Select)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN24nGLWBR / RLN21nGLWBR 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 / RLN21nmLiMD register (the LCKS bit is not changed) and the 2.5-Tbit or longer low-level width of the input signal to 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 / RLN21nmLiMD register.

16.3.2.2 RLN24nGLBRP0 / RLN21nGLBRP0 — LIN Baud Rate Prescaler 0 Register

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

Address: RLN24nGLBRP0: <RLIN24n_base> + 02_H
RLN21nGLBRP0: <RLIN21n_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 16.13 RLN24nGLBRP0 / RLN21nGLBRP0 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 / RLN21nGLBRP0 register when all channels in the same unit are in LIN reset mode (while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST 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.

16.3.2.3 RLN24nGLBRP1 / RLN21nGLBRP1 — LIN Baud Rate Prescaler 1 Register

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

Address: RLN24nGLBRP1: <RLIN24n_base> + 03_H
RLN21nGLBRP1: <RLIN21n_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 16.14 RLN24nGLBRP1 / RLN21nGLBRP1 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 / RLN21nGLBRP1 register when all channels in the same unit are in LIN reset mode (while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST 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.

16.3.2.4 RLN24nGLSTC / RLN21nGLSTC — LIN Self-Test Control Register

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

Address: RLN24nGLSTC: <RLIN24n_base> + 04_H
RLN21nGLSTC: <RLIN21n_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 16.15 RLN24nGLSTC / RLN21nGLSTC Register Contents

Bit Position	Bit Name	Function
7 to 0	—	Writing $A7_H$, 58_H, and 01_H successively to the RLN24nGLSTC / RLN21nGLSTC 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 RLN24nGLSTC / RLN21nGLSTC register cancels protection of LIN self-test mode.

Set the RLN24nGLSTC / RLN21nGLSTC register when all channels in the same unit are in LIN reset mode (while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0).

Writing $A7_H$, 58_H, and 01_H successively to the RLN24nGLSTC / RLN21nGLSTC 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 16.15, 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)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.
For leaving LIN self-test mode, see **Section 16.15, LIN Self-Test Mode**.
Writing 1 to this bit does not affect the value of the RLN24nGLSTC / RLN21nGLSTC register if it is not a part of successive writing of $A7_H$, 58_H, and 01_H.

16.3.3 Channel Registers

16.3.3.1 RLN24nmLiMD / RLN21nmLiMD — LIN Mode Register

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

Address: RLN24nmLiMD: <RLIN24n_base> + 08_H + i × 20_H
RLN21nmLiMD: <RLIN21n_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 16.16 RLN24nmLiMD / RLN21nmLiMD Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
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	Reserved	When read, the value after reset is returned. The write value should always be the value after reset.

Set the RLN24nmLiMD / RLN21nmLiMD register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0 (in LIN reset mode).

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 the LWBR0 bit in the RLN24nGLWBR / RLN21nGLWBR is 1 (when LIN 2.x is used) and the RLN24nmLiMST / RLN21nmLiMST register is 01h (in LIN wake-up mode), regardless of the setting of the LWBR0 bit, fa is input to the protocol controller (LCKS bit is not changed).

16.3.3.2 RLN24nmLiBFC / RLN21nmLiBFC — LIN Break Field Configuration Register

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

Address: RLN24nmLiBFC: <RLIN24n_base> + 09_H + i × 20_H
RLN21nmLiBFC: <RLIN21n_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 16.17 RLN24nmLiBFC / RLN21nmLiBFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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 RLN24nmLiBFC / RLN21nmLiBFC register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0 (in 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 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.

16.3.3.3 RLN24nmLiSC / RLN21nmLiSC — LIN Space Configuration Register

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

Address: RLN24nmLiSC: <RLIN24n_base> + $0A_H$ + i × 20_H
RLN21nmLiSC: <RLIN21n_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 16.18 RLN24nmLiSC / RLN21nmLiSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to this bit, 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 RLN24nmLiSC / RLN21nmLiSC register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0 (in 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; the setting is disabled during response reception.
The inter-byte space (header) is equal to the response space.

16.3.3.4 RLN24nmLiWUP / RLN21nmLiWUP — LIN Wake-Up Configuration Register

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

Address: RLN24nmLiWUP: <RLIN24n_base> + $0B_H$ + i × 20_H
RLN21nmLiWUP: <RLIN21n_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 16.19 RLN24nmLiWUP / RLN21nmLiWUP 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 to these bits, write the value after reset.

Set the RLN24nmLiWUP / RLN21nmLiWUP register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0 (in 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.
When the LWBR0 bit in the RLN24nGLWBR / RLN21nGLWBR is 1 (when LIN 2.x is used), regardless of the setting of the LCKS bit in the RLN24nmLiMD / RLN21nmLiMD register, fa is selected as the LIN system clock (fLIN) (LCKS bit is not changed).

16.3.3.5 RLN24nmLiIE / RLN21nmLiIE — LIN Interrupt Enable Register

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

Address: RLN24nmLiIE: <RLIN24n_base> + $0C_H$ + i × 20_H
RLN21nmLiIE: <RLIN21n_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 16.20 RLN24nmLiIE / RLN21nmLiIE Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
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 RLN24nmLiIE / RLN21nmLiIE register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0 (in LIN reset mode).

ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables interrupt request upon detection of an error.
With 0 set, the interrupt request is not generated when the ERR flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.
With 1 set, the interrupt request is generated when the ERR flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.
Errors that constitute interrupt sources can be the bit error, physical bus error, frame timeout error, framing error, and checksum error.
Detection of the bit error, physical bus error, frame/response timeout error, and framing error can be enabled or disabled using the RLN24nmLiEDE / RLN21nmLiEDE register.

FRCIE Bit (Successful Frame/Wake-Up Reception Interrupt Request Enable)

The FRCIE bit enables or disables interrupt request upon successful reception of a frame or a wake-up signal (counting of low level width of the input signal).
With 0 set, the interrupt request is not generated when the FRC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.
With 1 set, the interrupt request is generated when the FRC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.

FTCIE Bit (Successful Frame/Wake-Up Transmission Interrupt Request Enable)

The FTCIE bit enables or disables interrupt request upon successful transmission of a frame or a wake-up signal.
With 0 set, the interrupt request is not generated when the FTC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.
With 1 set, the interrupt request is generated when the FTC flag in the RLN24nmLiST / RLN21nmLiST register is set to 1.

16.3.3.6 RLN24nmLiEDE / RLN21nmLiEDE — LIN Error Detection Enable Register

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

Address: RLN24nmLiEDE: <RLIN24n_base> + $0D_H$ + i × 20_H
RLN21nmLiEDE: <RLIN21n_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 16.21 RLN24nmLiEDE / RLN21nmLiEDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	FTERE	Frame Timeout Error Detection Enable 0: Disables frame timeout error detection. 1: Enables frame 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 0: Disables bit error detection. 1: Enables bit error detection.

Set the RLN24nmLiEDE / RLN21nmLiEDE register while the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 0_B (in 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 RLN24nmLiEST / RLN21nmLiEST register.
For details on the framing error, see **Section 16.14, Error Status**.

FTERE Bit (Frame Timeout Error Detection Enable)

The FTERE bit enables or disables detection of the frame timeout error.
With 0 set, the frame timeout error is not detected.
With 1 set, the frame timeout error is detected.
When this bit is set to 1, the detection result is indicated in the FTER flag in the RLN24nmLiEST / RLN21nmLiEST register.
For details on the frame timeout error, see **Section 16.14, 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 RLN24nmLiEST / RLN21nmLiEST register.
For details on the physical bus error, see **Section 16.14, 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 RLN24nmLiEST / RLN21nmLiEST register.
For details on the bit error, see **Section 16.14, Error Status**.

16.3.3.7 RLN24nmLiCUC / RLN21nmLiCUC — LIN Control Register

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

Address: RLN24nmLiCUC: <RLIN24n_base> + $0E_H$ + i × 20_H
RLN21nmLiCUC: <RLIN21n_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 16.22 RLN24nmLiCUC / RLN21nmLiCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode is caused. 1: LIN operation mode is caused.
0	OM0	LIN Reset 0: LIN reset mode is caused. 1: LIN reset mode is canceled.

Set the RLN24nmLiCUC / RLN21nmLiCUC register to 01_H to cause a transition to LIN wake-up mode after canceling LIN reset mode, and set the register to 03_H to cause a transition to LIN operation mode.

In LIN self-test mode, set the RLN24nmLiCUC / RLN21nmLiCUC 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 RLN24nmLiMST / RLN21nmLiMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.
With 0 set, LIN wake-up mode is caused.
With 1 set, LIN operation mode is caused.
This bit is valid only when the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register is 1.
Writing a value to this bit is disabled while the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC 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 is caused.
With 1 set, LIN reset mode is canceled.

16.3.3.8 RLN24nmLiTRC / RLN21nmLiTRC — LIN Transmission Control Register

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

Address: RLN24nmLiTRC: <RLIN24n_base> + 10_H + i × 20_H
RLN21nmLiTRC: <RLIN21n_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 16.23 RLN24nmLiTRC / RLN21nmLiTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	RTS	Response Transmission Start 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 0: Frame Transmission/wake-up transmission/reception is stopped. 1: Frame Transmission/wake-up transmission/reception is started.

RTS Bit (Response Transmission Start)

Set the RTS bit (response transmit start 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 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 RLN24nmLiTRC / RLN21nmLiTRC register using the store instruction.
Writing a value to this bit is disabled when the OMM0 bit is 0 (in LIN reset mode). When the OMM1 bit is “0” (in LIN wake-up mode), do not write “1”.
Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is halted).

FTS Bit (Frame Transmission/Wake-Up Transmission/Reception Start)

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 low level width of the input signal).
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 (in LIN reset mode).
This bit is set to 0 upon completion of frame or wake-up communication and transition to LIN reset mode.

16.3.3.9 RLN24nmLiMST / RLN21nmLiMST — LIN Mode Status Register

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

Address: RLN24nmLiMST: <RLIN24n_base> + 11_H + i × 20_H
RLN21nmLiMST: <RLIN21n_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 16.24 RLN24nmLiMST / RLN21nmLiMST 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: The LIN master interface is in LIN wake-up mode. 1: The LIN master interface is in LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: The LIN master interface is in LIN reset mode. 1: The LIN master interface is 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" (in LIN reset mode), the value of this bit is invalid.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

16.3.3.10 RLN24nmLiST / RLN21nmLiST — LIN Status Register

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

Address: RLN24nmLiST: <RLIN24n_base> + 12_H + i × 20_H
RLN21nmLiST: <RLIN21n_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 16.25 RLN24nmLiST / RLN21nmLiST 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 to these bits, 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 to these bits, 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 RLN24nmLiST / RLN21nmLiST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

Writing to this register is prohibited while the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is 1 (frame transmit or 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 (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.
“1” is set upon completion of header transmission, but an interrupt request is not generated.
To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC 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.
“1” is set upon completion of Data 1 reception, but an interrupt request is not generated.
To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC 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 (when the value of any of the flags of the RLN24nmLiEST / RLN21nmLiEST registers is 1). Here, an interrupt request is generated if the ERRIE bit in the RLN24nmLiIE / RLN21nmLiIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN24nmLiEST / RLN21nmLiEST 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 is generated if the FRCIE bit in the RLN24nmLiIE / RLN21nmLiIE register is 1 (interrupt is enabled).
To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

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 is generated if the FTCIE bit in the RLN24nmLiIE / RLN21nmLiIE register is 1 (interrupt is enabled).
To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

16.3.3.11 RLN24nmLiEST / RLN21nmLiEST — LIN Error Status Register

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

Address: RLN24nmLiEST: <RLIN24n_base> + 13_H + i × 20_H
RLN21nmLiEST: <RLIN21n_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 16.26 RLN24nmLiEST / RLN21nmLiEST Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, 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 / RLN21nmLiEST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC 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 / RLN21nmLiTRC 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.

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 value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC 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.
The FER flag is set to 1 upon framing error detection when the FERE bit of the RLN24nmLiEST / RLN21nmLiEST register is 1 (framing error detection enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC 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 value that has been retained before 1 is written.
The FTER flag is set to 1 upon frame timeout detection when the FTERE bit of the RLN24nmLiEDE / RLN21nmLiEDE register is 1 (frame timeout detection enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC 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.
The PBER flag is set to 1 upon physical bus error detection when the PBERE bit of the RLN24nmLiEDE / RLN21nmLiEDE register is 1 (physical bus error detection enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC 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.
The BER flag is set to 1 upon bit error detection when the BERE bit of the RLN24nmLiEDE / RLN21nmLiEDE register is 1 (bit error detection enabled). To clear the bit to 0 before the next communication (when the value of the FTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

16.3.3.12 RLN24nmLiDFC / RLN21nmLiDFC — LIN Data Field Configuration Register

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

Address: RLN24nmLiDFC: <RLIN24n_base> + 14_H + i × 20_H
RLN21nmLiDFC: <RLIN21n_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 16.27 RLN24nmLiDFC / RLN21nmLiDFC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
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 mode 1: Enhanced checksum mode
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.

Set the RLN24nmLiDFC / RLN21nmLiDFC register when the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

FSM Bit (Frame Separate Mode Select)

The FSM bit sets the response transmission mode.

With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is 1), response is transmitted/received without the RTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register being set.
With 1 set, frame separate mode is selected. When the RTS bit of the RLN24nmLiTRC / RLN21nmLiTRC register is set to 1 during header transmission, response transmission is executed after header transmission has ended.
For response reception (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 16.11.1, Transmission of LIN Frames**.

CSM Bit (Checksum Select)

The CSM bit sets checksum mode.
With 0 set, classic checksum mode is selected.
With 1 set, enhanced checksum mode is selected.
When the frame timeout error is used (the FTERE bit in the RLN24nmLiEDE / RLN21nmLiEDE register is 1), the timeout time depends on the setting of this bit. For details on the bit error, see **Section 16.14, Error Status**.

RFT Bit (Response Field Communication Direction Select)

The RFT bit sets the direction of the response field/wake-up signal communication/wake-up.
With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (low level width of the input signal is counted).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.

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.

16.3.3.13 RLN24nmLiIDB / RLN21nmLiIDB — LIN ID Buffer Register

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

Address: RLN24nmLiIDB: <RLIN24n_base> + 15_H + i × 20_H
RLN21nmLiIDB: <RLIN21n_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 16.28 RLN24nmLiIDB / RLN21nmLiIDB 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 RLN24nmLiIDB / RLN21nmLiIDB register when the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC 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 16.15, 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 for P0 and IDP1 for P1.
Since parity is not automatically calculated, set the calculation value. Note that if the erroneous result is set, it is transmitted as is.

ID[5:0] Bits (ID Setting)

The ID bit sets the 6-bit ID value to be transmitted in the ID field of the LIN frame.

16.3.3.14 RLN24nmLiCBR / RLN21nmLiCBR — LIN Checksum Buffer Register

Access: This register can be read only in 8-bit units. In LIN self-test mode, this register can be read/written in 8-bit units.

Address: RLN24nmLiCBR: <RLIN24n_base> + 16_H + i × 20_H
RLN21nmLiCBR: <RLIN21n_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 16.29 RLN24nmLiCBR / RLN21nmLiCBR 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 / RLN21nmLiDFC 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 RLN24nmLiDFC / RLN21nmLiDFC 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.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the RLN24nmLiDFC / RLN21nmLiDFC 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 / RLN21nmLiDFC 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 16.15, LIN Self-Test Mode**.

Set the RLN24nmLiCBR / RLN21nmLiCBR register when the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

16.3.3.15 RLN24nmLiDBRb / RLN21nmLiDBRb — LIN Data Buffer b Register

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

Address: RLN24nmLiDBR1: <RLIN24n_base> + 18_H + i × 20_H, RLN21nmLiDBR1: <RLIN21n_base> + 18_H + i × 20_H
RLN24nmLiDBR2: <RLIN24n_base> + 19_H + i × 20_H, RLN21nmLiDBR2: <RLIN21n_base> + 19_H + i × 20_H
RLN24nmLiDBR3: <RLIN24n_base> + $1A_H$ + i × 20_H, RLN21nmLiDBR3: <RLIN21n_base> + $1A_H$ + i × 20_H
RLN24nmLiDBR4: <RLIN24n_base> + $1B_H$ + i × 20_H, RLN21nmLiDBR4: <RLIN21n_base> + $1B_H$ + i × 20_H
RLN24nmLiDBR5: <RLIN24n_base> + $1C_H$ + i × 20_H, RLN21nmLiDBR5: <RLIN21n_base> + $1C_H$ + i × 20_H
RLN24nmLiDBR6: <RLIN24n_base> + $1D_H$ + i × 20_H, RLN21nmLiDBR6: <RLIN21n_base> + $1D_H$ + i × 20_H
RLN24nmLiDBR7: <RLIN24n_base> + $1E_H$ + i × 20_H, RLN21nmLiDBR7: <RLIN21n_base> + $1E_H$ + i × 20_H
RLN24nmLiDBR8: <RLIN24n_base> + $1F_H$ + i × 20_H, RLN21nmLiDBR8: <RLIN21n_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 16.30 RLN24nmLiDBRb / RLN21nmLiDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or allows the received data to be read. Setting range: 00_H to FF_H

- For response transmission:

  The LDBRn registers set the data to be transmitted in the response field.
  Use these registers with the following settings.

  – The RFT bit in RLN24nmLiDFC / RLN21nmLiDFC register is 1 (transmission).
  – The FSM bit in RLN24nmLiDFC / RLN21nmLiDFC register is 0 (not frame separate mode).
  – The FTS bit in RLN24nmLiTRC / RLN21nmLiTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

  or

  – The RFT bit in RLN24nmLiDFC / RLN21nmLiDFC register is 1 (transmission).
  – The FSM bit in RLN24nmLiDFC / RLN21nmLiDFC register is 1 (frame separate mode).
  – The RTS bit in RLN24nmLiDFC / RLN21nmLiDFC register is 0 (response transmission is halted).

- For response reception:

  The LDBRn 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)

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 16.15, LIN Self-Test Mode**.

16.4 Interrupt Sources

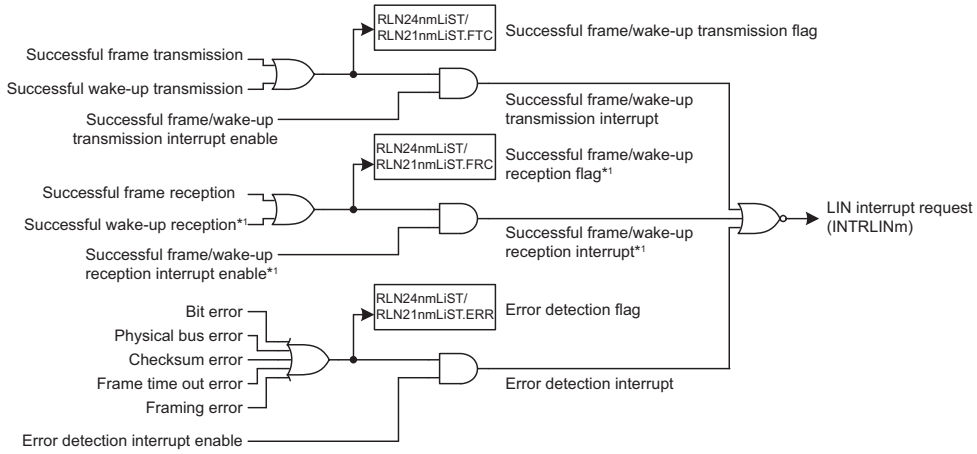
The LIN interrupts are interrupt requests generated by the LIN master interface.

There are three interrupt factors for each channel; frame/wake-up transmit completion, frame/wake-up receive completion, and error detection.

The interrupt request from three interrupt states, frame/wake-up transmit completion, frame/wake-up receive completion, and error detection, is ORed to be one interrupt request “LIN interrupt”.

The respective interrupt request is output when the corresponding flag in the RLN24nmLiST / RLN21nmLiST register is set to 1 while the corresponding bit in the RLN24nmLiIE / RLN21nmLiIE register is 1 (interrupt enabled). However, if an interrupt is requested when the corresponding flag in the RLN24nmLiST / RLN21nmLiST register has been set to 1, it is ignored. Therefore, clear the corresponding flag to 0 to enable the interrupt.

Figure 16.2 shows a block diagram of the LIN interrupt.



Note 1. The wake-up receive signal indicates the count of input signal low width.

Figure 16.2 LIN Interrupt Block Diagram

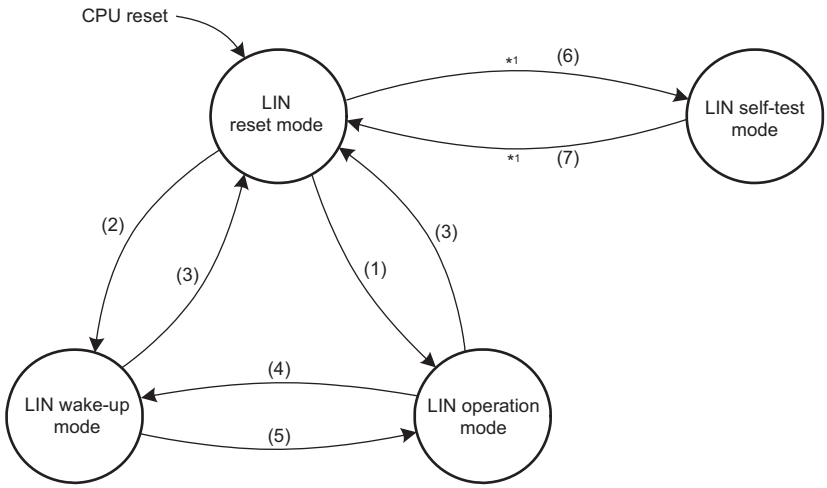
16.5 Modes

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 16.3 shows mode transitions. **Table 16.31** describes mode transition conditions. **Table 16.32** lists operations available in each mode.



Note 1. See **Section 16.15, LIN Self-Test Mode** for the details.

Figure 16.3 Mode Transitions

Table 16.31 Transition Condition of Each Mode

	Mode Transition			Transition Condition
(1)	LIN reset mode	→	LIN operation mode	RLN24nmLiCUC / RLN21nmLiCUC.OM1,OM0 = 11_B
(2)	LIN reset mode	→	LIN wake-up mode	RLN24nmLiCUC / RLN21nmLiCUC.OM1,OM0 = 01_B
(3)	LIN wake-up mode LIN operation mode	→	LIN reset mode	RLN24nmLiCUC / RLN21nmLiCUC.OM0 = 0_B
(4)	LIN operation mode	→	LIN wake-up mode	RLN24nmLiCUC / RLN21nmLiCUC.OM1,OM0 = 01_B
(5)	LIN wake-up mode	→	LIN operation mode	RLN24nmLiCUC / RLN21nmLiCUC.OM1,OM0 = 11_B
(6)	LIN reset mode	→	LIN self-test mode	See **Section 16.15, LIN Self-Test Mode**.
(7)	LIN self-test mode	→	LIN reset mode	See **Section 16.15, LIN Self-Test Mode**.

Table 16.32 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 caused 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 / RLN21nmLiMST register.

For a description of LIN self-test mode, see **Section 16.15, LIN Self-Test Mode**.

16.6 LIN Reset Mode

Setting the OM0 bit in the RLN24nmLiCUC / RLN21nmLiCUC register to 0 (in 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 RLN24nmLiMST / RLN21nmLiMST register has been set to 0 (in 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 changes to LIN reset mode, the following registers are initialized to their reset values, and they retain their initial values.

- RLN24nmLiTRC / RLN21nmLiTRC register
- RLN24nmLiST / RLN21nmLiST register
- RLN24nmLiEST / RLN21nmLiEST register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN24nGLWBR / RLN21nGLWBR register
- RLN24nGLBRP0 / RLN21nGLBRP0 register
- RLN24nGLBRP1 / RLN21nGLBRP1 register
- RLN24nmLiMD / RLN21nmLiMD register
- RLN24nmLiBFC / RLN21nmLiBFC register
- RLN24nmLiSC / RLN21nmLiSC register
- RLN24nmLiWUP / RLN21nmLiWUP register
- RLN24nmLiIE / RLN21nmLiIE register
- RLN24nmLiEDE / RLN21nmLiEDE register
- RLN24nmLiDFC / RLN21nmLiDFC register
- RLN24nmLiIDB / RLN21nmLiIDB register
- RLN24nmLiCBR / RLN21nmLiCBR register
- RLN24nmLiDBRb / RLN21nmLiDBRb register

16.7 LIN Operation Mode

In LIN operation mode, frame processing (header transmission, 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 RLN24nmLiCUC / RLN21nmLiCUC register to 11_B changes the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register to 11_B. Communication settings should be performed after the RLN24nmLiMST / RLN21nmLiMST register has become 11_B.

16.8 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 RLN24nmLiCUC / RLN21nmLiCUC register to 01_B changes the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register to 01_B. Communication settings should be performed after the RLN24nmLiMST / RLN21nmLiMST register has become 01_B.

16.9 Header Transmission/Response Transmission/Response Reception

16.9.1 Header Transmission

Figure 16.4 shows the operation of the LIN master interface in header transmission. **Table 16.33** provides processing in header transmission.

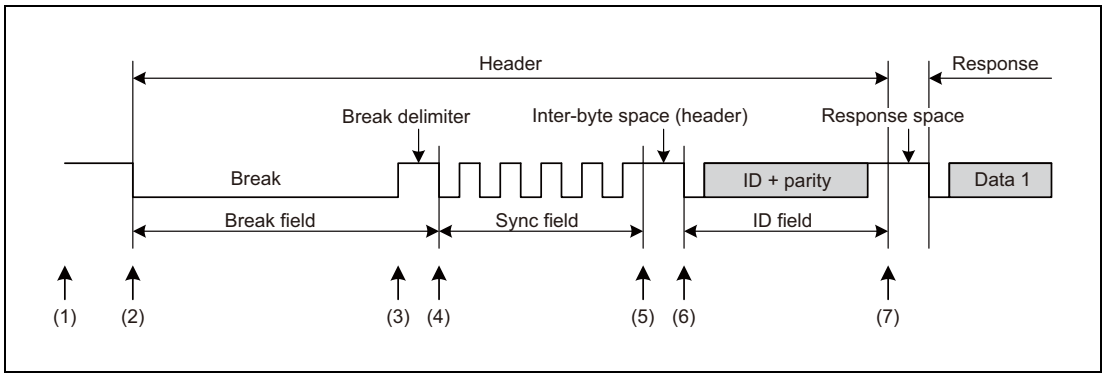


Figure 16.4 Operation in Header Transmission

Table 16.33 Processing in Header Transmission

	Software Processing	LIN Master Interface Processing
(1)	• Sets a baud rate • Enables interrupt • Enables error detection • Sets frame configuration parameters • Changes 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 RLN24nmLiTRC / RLN21nmLiTRC register by software (idle).
(2)	Sets the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC 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 (55h).
(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, see **Section 16.14, Error Status**.

16.9.2 Response Transmission

Figure 16.5 shows the operation of the LIN master interface in response transmission. **Table 16.34** provides processing in response transmission.

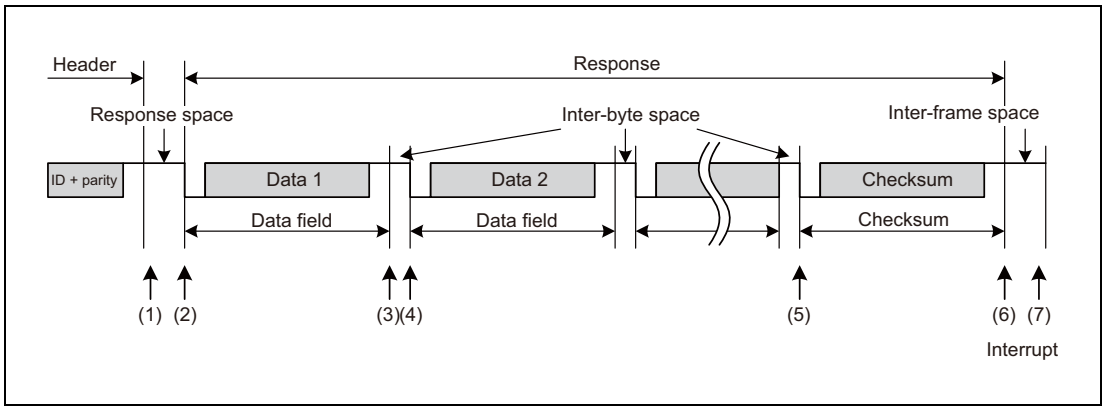


Figure 16.5 Operation in Response Transmission

Table 16.34 Processing in Response Transmission

	Software Processing	LIN Master Interface Processing
(1)	(When in frame separate mode) • Sets the RTS bit in the RLN24nmLiTRC / RLN21nmLiTRC 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 setting of the RTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register 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 the 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 the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RLN24nmLiDFC / RLN21nmLiDFC register.) : :
(5)		Transmits the checksum.
(6)		• Sets a successful frame/wake-up transmission flag. • Sets the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 0 (frame transmission or wake-up transmission/ reception stopped) (When in frame separate mode) • Sets the RTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 0 (response transmission stopped).
(7)	• Processing after communication Checks the RLN24nmLiST / RLN21nmLiST register, and clears flags.	Idle

NOTE

For information about error detection, see **Section 16.14, Error Status**.

16.9.3 Response Reception

Figure 16.6 shows the operation of the LIN master interface on response reception. **Table 16.35** provides processing in response reception.

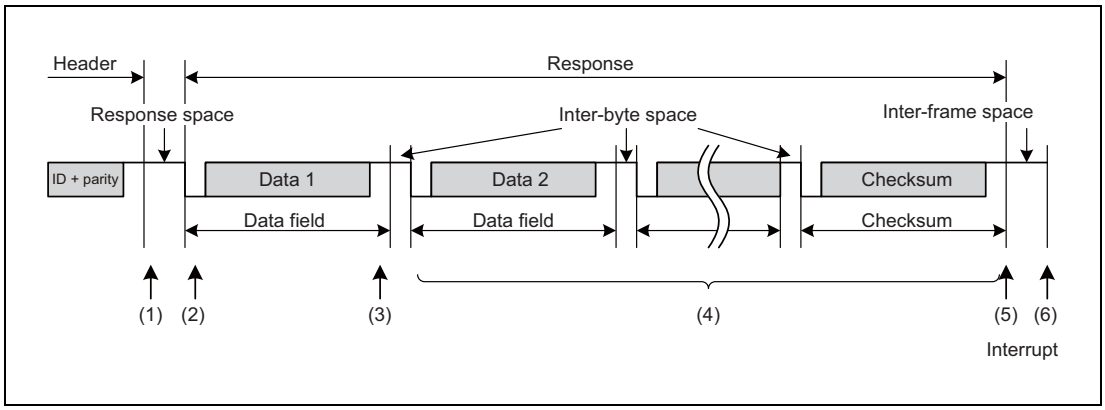


Figure 16.6 Operation in Response Reception

Table 16.35 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 successful data 1 reception flag.
(4)		• Receives the data 2 when the start bit is detected. • Receives the data 3 when the start bit is detected. (Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RLN24nmLiDFC / RLN21nmLiDFC 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 RLN24nmLiTRC / RLN21nmLiTRC register to 0 (frame transmission or wake-up transmission/reception stopped).
(6)	• Processing after communication Reads the received data. Checks the RLN24nmLiST / RLN21nmLiST register, and clears flags.	Idle

NOTE

For information about error detection, see **Section 16.14, Error Status**.

16.10 Data Transmission/Reception

16.10.1 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 is compared bit by bit, and the results are stored in the BER flag in the RLN24nmLiEST / RLN21nmLiEST register (see **Section 16.14, Error Status**).

In LIN mater interface, the sampling point for received data, 1 Tbit is generated to be 16fLIN, and thus is at the 13th clock cycle (81.25% position).

Figure 16.7 shows an example of data transmission timing.

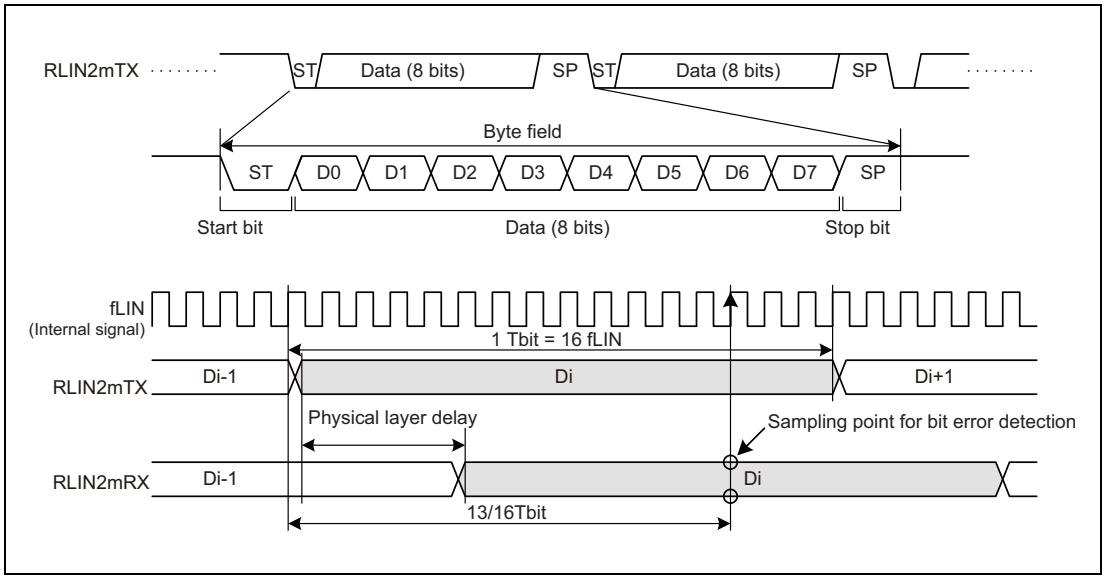


Figure 16.7 Example of Data Transmission Timing

16.10.2 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 after the clearing of the resetting is low-level-fixed or if a high level is detected on re-sampling.

After the start bit is detected, the system samples 1 bit per Tbit.

Figure 16.8 shows an example of data reception timing.

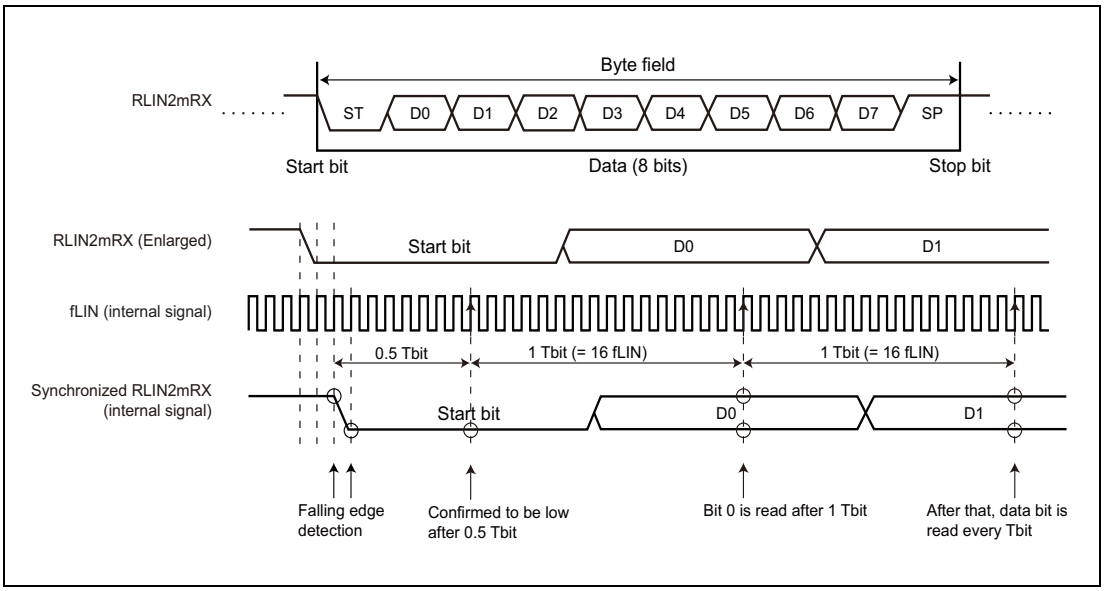


Figure 16.8 Example of Data Reception Timing

16.11 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN master interface sends or receives data continuously.

16.11.1 Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN24nmLiDBR1 / RLN21nmLiDBR1 to RLN24nmLiDBR8 / RLN21nmLiDBR8 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 / RLN21nmLiDBR1 to RLN24nmLiDBR4 / RLN21nmLiDBR4 are transmitted to data areas 1 to 4 of the LIN frame, but the contents of registers RLN24nmLiDBR5 / RLN21nmLiDBR5 to RLN24nmLiDBR8 / RLN21nmLiDBR8 are not transmitted. The transmitted checksum data is stored in the RLN24nmLiCBR / RLN21nmLiCBR register.

Figure 16.9 shows the LIN transmission processing and the required buffer.

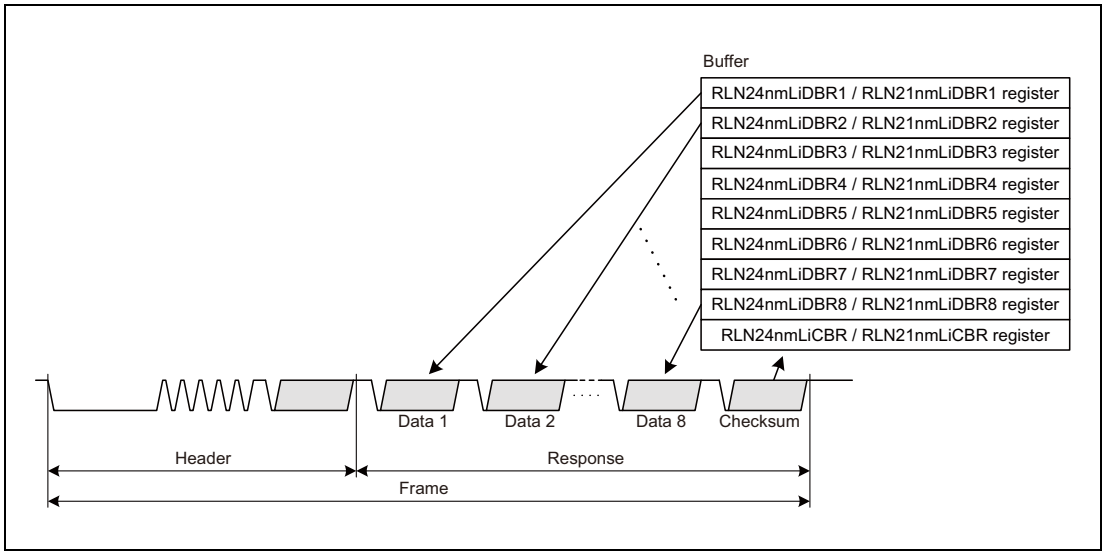


Figure 16.9 LIN Transmission Processing and Required Buffer

(1) Frame Separate Mode

Setting the FSM bit in the RLN24nmLiDFC / RLN21nmLiDFC 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 RLN24nmLiST / RLN21nmLiST register turns 1 (successful header transmission).

16.11.2 Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame is stored in registers RLN24nmLiDBR1 / RLN21nmLiDBR1 to RLN24nmLiDBR8 / RLN21nmLiDBR8, 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 / RLN21nmLiDBR1 to RLN24nmLiDBR4 / RLN21nmLiDBR4, respectively; however, no data is stored in registers RLN24nmLiDBR5 / RLN21nmLiDBR5 to RLN24nmLiDBR8 / RLN21nmLiDBR8. Also, the received checksum data is stored in the RLN24nmLiCBR / RLN21nmLiCBR register.

Figure 16.10 shows the LIN reception processing and the required buffer.

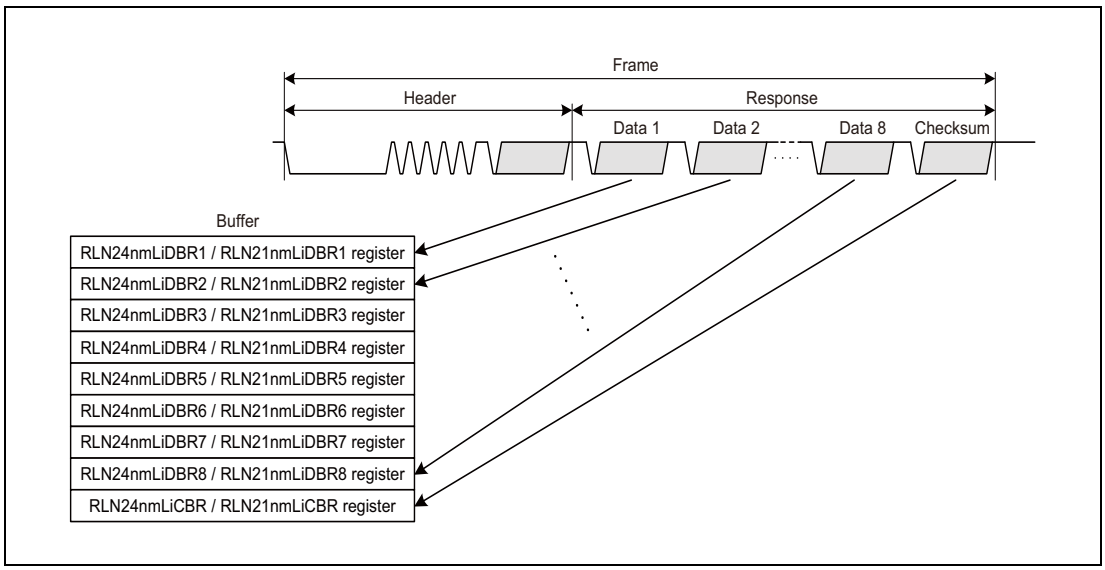


Figure 16.10 LIN Reception Processing and Required Buffer

(1) Reception of Data 1

When the reception of the first byte of data is finished, the D1RC flag in the RLN24nmLiST / RLN21nmLiST register turns 1 (successful data 1 reception).

16.12 Wake-Up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

16.12.1 Wake-Up Transmission

In LIN wake-up mode, setting the RFT bit in the RLN24nmLiDFC / RLN21nmLiDFC register to 1 (transmission) and the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC 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 should be set using the WUTL[3:0] bits in the RLN24nmLiWUP / RLN21nmLiWUP register. However, if the value of the LWBR0 bit of the RLN24nGLWBR / RLN21nGLWBR register is 1 (LIN2.x use), the LIN system clock (fLIN) becomes low level width at fa regardless of the setting of the LCKS bit of the RLN24nmLiMD / RLN21nmLiMD register. By setting the WUTL[3:0] bits of the RLN24nmLiWUP / RLN21nmLiWUP 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 / RLN21nmLiMD register.

If a wake-up low is output without any error, the FTC flag in the RLN24nmLiST / RLN21nmLiST register turns 1 (successful frame or wake-up transmission); when the FTCIE bit in the RLN24nmLiIE / RLN21nmLiIE 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 / RLN21nmLiEST register) is set to 1 (physical bus error detection / bit error detection).

Figure 16.11 shows the wake-up transmission timing.

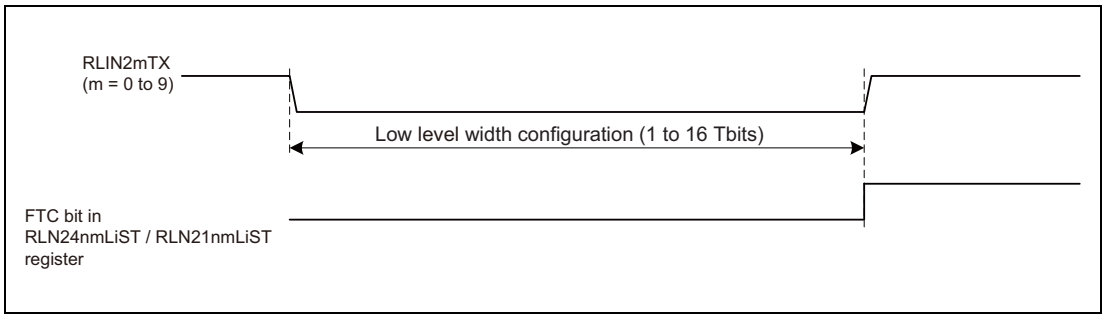


Figure 16.11 Wake-Up Transmission Timing

16.12.2 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 the 2.5-Tbit or longer low-level width of the input signal of fLIN to be measured.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN24nGLWBR / RLN21nGLWBR register to 0. When LIN Specification Package Revision 2.x is used, set the RLN24nGLWBR / RLN21nGLWBR register to 1.

When LWBR0 bit is set to 1, regardless of the setting of the LCKS bit in the RLN24nmLiMD / RLN21nmLiMD 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 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 RLN24nmLiMD / RLN21nmLiMD register.

When using this function, in LIN wake-up mode set the RFT bit in the RLN24nmLiDFC / RLN21nmLiDFC register to 0 (reception), and the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC 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 / RLN21nmLiST register becomes 1 (successful frame or wake-up reception). If the FRCIE bit in the RLN24nmLiIE / RLN21nmLiIE register is 1 (successful frame or wake-up reception interrupt enabled), an interrupt request is generated.

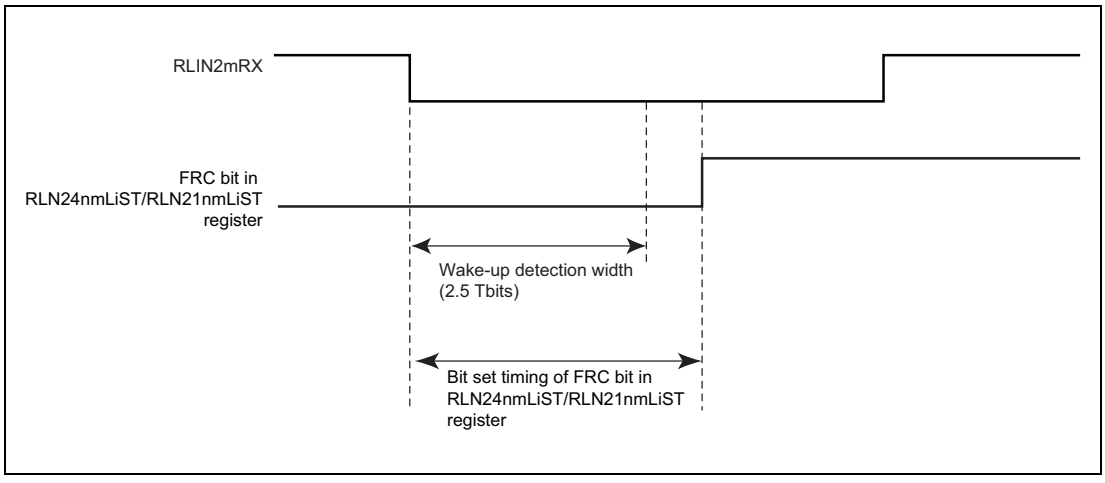


Figure 16.12 Input Signal Low Level Count Function

16.12.3 Wake-Up Collision

If the master node and the slave node transmit wake-up signals simultaneously, a collision will occur on the LIN bus, though a collision of wake-up signals is not detected in the LIN master interface.

16.13 Status

During LIN mode operation, the LIN master interface can detect seven types of statuses.

The three statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, can generate interrupt requests.

Table 16.36 shows the types of statuses available.

Table 16.36 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 / RLN21nmLiCUC register is set to not-LIN–reset-mode, if actually the LIN master interface is cleared from LIN reset mode.	After the OM0 bit in the RLN24nmLiCUC / RLN21nmLiCUC register is set to LIN reset mode, if actually the LIN master interface enters LIN reset mode.	All modes	OMM0 bit in the RLN24nmLiMST/ RLN21nmLiMST register	—
Operation mode	After the OM1 bit in the RLN24nmLiCUC / RLN21nmLiCUC register is set to LIN operation mode, if actually the LIN master interface enters LIN operation mode.	After the OM1 bit in the RLN24nmLiCUC / RLN21nmLiCUC register is set to LIN wake-up mode, if actually the LIN master interface enters LIN wake-up mode.	• LIN operation mode • LIN wake-up mode	OMM1 bit in the RLN24nmLiMST/ RLN21nmLiMST register	—
Frame/wake-up transmission end	When a frame (header transmission + response transmission) or a wake-up signal is transmitted successfully.	• When another 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/ RLN21nmLiST register	√
Frame/wake-up reception end	When a frame (header transmission + response reception) or a wake-up signal is received successfully.	• When another communication is started • When cleared by software*1 • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FRC flag in the RLN24nmLiST/ RLN21nmLiST register	√
Error detection	If any of the CSER flag, FER flag, FTER flag, PBER flag, and BER flag in the RLN24nmLiEST / RLN21nmLiEST register turns 1 (error detected).	• When another 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/ RLN21nmLiST register	√
Data 1 reception end	The RFT bit in the RLN24nmLiDFC/ RLN21nmLiDFC register is 0 (reception) and the first byte of the response field is received.*2	• When another communication is started • When cleared by software • After transition to LIN reset mode	LIN operation mode	D1RC flag in the RLN24nmLiST/ RLN21nmLiST register	—
Header reception end	When a header field is received successfully.	• When another communication is started • When cleared by software • After transition to LIN reset mode	LIN operation mode	HTRC flag in the RLN24nmLiST/ RLN21nmLiST register	—

Note 1. In LIN operation mode, the ERR flag in the RLN24nmLiST / RLN21nmLiST register is cleared to 0 by writing 0 to the CSER flag, FER flag, FTER flag, PBER flag, or BER flag in the RLN24nmLiEST / RLN21nmLiEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN24nmLiDFC / RLN21nmLiDFC register are 0000_B (0-byte + checksum).

16.14 Error Status

16.14.1 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 / RLN21nmLiEST register.

All error statuses represent interrupt sources.

Table 16.37 shows the types of error statuses.

Table 16.37 Types of Error Statuses

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	• LIN operation mode • LIN wake-up mode	Cancel	√	BER flag in the RLN24nmLiEST/ RLN21nmLiEST register
Physical bus error	• LIN bus is detected to be high level when sending a break • LIN bus is detected to be low level when sending a break delimiter • LIN bus is detected to be high level when sending a wake-up	• LIN operation mode • LIN wake-up mode	Cancel	√	PBER flag in the RLN24nmLiEST/ RLN21nmLiEST register
Frame timeout error	A frame transmission/reception does not terminate within a given time*2	LIN operation mode	Cancel	√	FTER flag in the RLN24nmLiEST/ RLN21nmLiEST register
Framing error	In response field reception, a stop bit of each data byte is low level	LIN operation mode	Cancel	√	FER flag in the RLN24nmLiEST/ RLN21nmLiEST register
Checksum error	In response field reception, the result of checksum test gives an error	LIN operation mode	—	×	CSER flag in the RLN24nmLiEST/ RLN21nmLiEST register

Note 1. If a bit error is detected, the process is canceled 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 canceled immediately after that area. 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. The timeout time depends on the response field data length (the RFDL[3:0] bits in the RLN24nmLiDFC / RLN21nmLiDFC register) and the checksum selection (the CSM bit in the RLN24nmLiDFC / RLN21nmLiDFC register), and this can be calculated according to the following formula:
On classic selection (when the CSM bit in the RLN24nmLiDFC / RLN21nmLiDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
On enhanced selection (when the CSM bit in the RLN24nmLiDFC / RLN21nmLiDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]

The aforementioned timeout time is a time greater than the TFRAME_MAX of LIN Specification Package Revision 1.3 on classic selection, or the TFRAME_MAX of LIN Specification Package Revision 2.x on enhanced selection.

The error status is cleared when the next communication is started, by software, or at a transition to LIN reset mode.

16.14.2 Target Time Area for Error Detection

Figure 16.13 shows the time domain in which the LIN master interface performs monitoring for error detection.

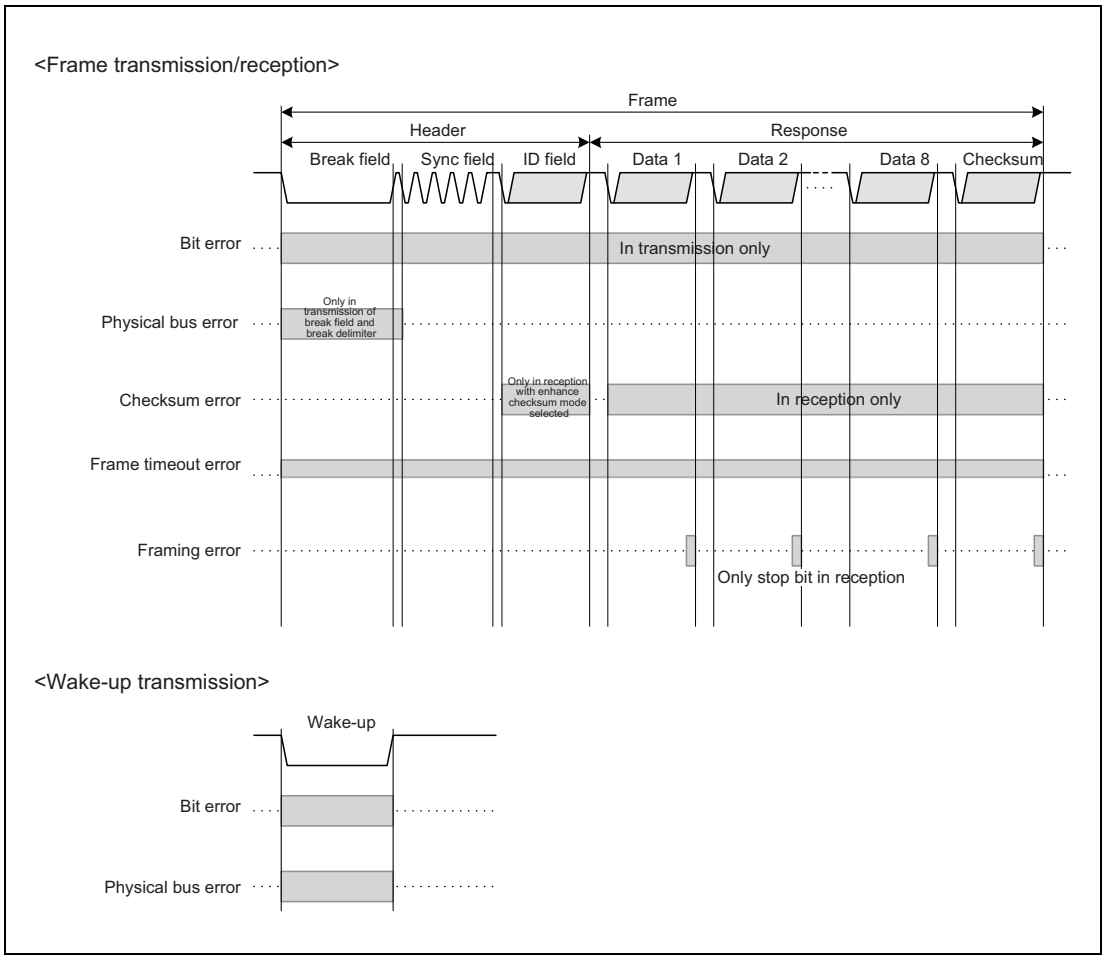


Figure 16.13 Target Time Area for Error Detection

16.15 LIN Self-Test Mode

The LIN master interface provides LIN self-test mode. When LIN self-test mode is turned on, RLIN2mTX and RLIN2mRX are disconnected from the external pin, and the internal RLIN2mTX and RLIN2mRX are connected. Thus, the frame transmitted from RLIN2mTX is returned to the internal RLIN2mRX (loop back).

The functions of LIN self-test mode operate in the following conditions:

- 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 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 function
- Frame separate mode

Do not use above functions.

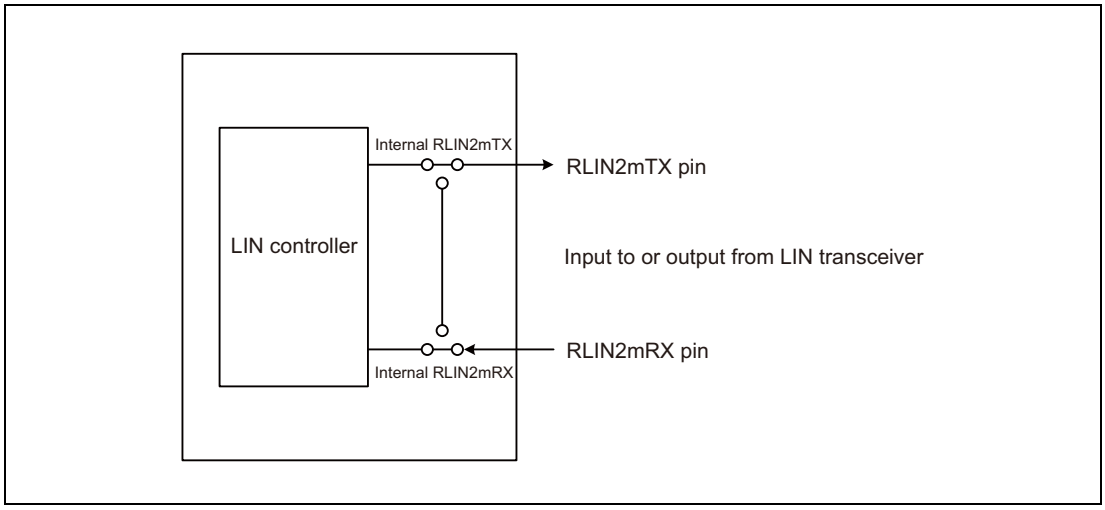


Figure 16.14 Connection in LIN Reset Mode, LIN Wake-Up Mode, and LIN Operation Mode

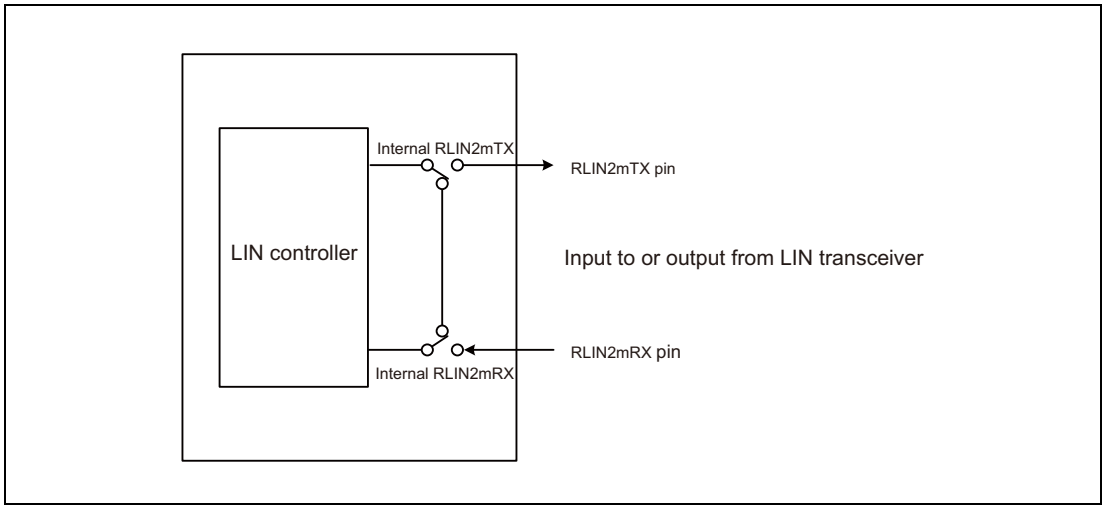


Figure 16.15 Connection in LIN Self-Test Mode

16.15.1 Change to LIN Self-Test Mode

Writing to the RLN24nGLSTC / RLN21nGLSTC register enables LIN self-test mode.

The LSTM bit in the RLN24nGLSTC / RLN21nGLSTC register becoming 1 indicates that the mode is transit to LIN self-test mode.

When changing 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 follows:

- Change to LIN reset mode regarding all channels in the unit.
  Set the OM0 bit in the RLN24nmLiCUC / RLN21nmLiCUC register to 0 (in LIN reset mode).
  Read the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register; verify that it is 0 (LIN reset mode).
- 1st write: RLN24nGLSTC / RLN21nGLSTC register = $1010\ 0111_B$ ($A7_H$)
- 2nd write: RLN24nGLSTC / RLN21nGLSTC register = $0101\ 1000_B$ (58_H)
- 3rd write: RLN24nGLSTC / RLN21nGLSTC register = $0000\ 0001_B$ (01_H)
- Verify the transition to LIN self-test mode
  Read the LSTM bit in the RLN24nGLSTC / RLN21nGLSTC register; verify that it is 1 (in 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 in the same unit is performed during transition to LIN self-test mode (three consecutive write operations to the RLN24nGLSTC / RLN21nGLSTC register), the transition is also canceled.

16.15.2 Transmission in LIN Self-Test Mode

To execute a self-test on transmission, perform the procedure below:

- Set the baud rate related registers.
  RLN24nGLBRP0 / RLN21nGLBRP0 register = xxxx $xxxx_B$ *1
  RLN24nGLBRP1 / RLN21nGLBRP1 register = xxxx $xxxx_B$ *1
  RLN24nmLiMD / RLN21nmLiMD register = 0000 $xx00_B$ *1
- Set interrupt enable register and error enable related registers.
  RLN24nmLiIE / RLN21nmLiIE register = 0000 $0xxx_B$ *2
  RLN24nmLiEDE / RLN21nmLiEDE register = 0000 $x0xx_B$
- Set the break field and space related registers.
  RLN24nmLiBFC / RLN21nmLiBFC register = 00xx $xxxx_B$
  RLN24nmLiSC / RLN21nmLiSC register = 00xx $0xxx_B$
- Cancel the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN24nmLiCUC / RLN21nmLiCUC register, and check that the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register are 11_B.
- Set the transmit frame related registers.
  RLN24nmLiDFC / RLN21nmLiDFC register = 00x1 $xxxx_B$
  RLN24nmLiIDB / RLN21nmLiIDB register = xxxx $xxxx_B$
  RLN24nmLiDBR1 / RLN21nmLiDBR1 to RLN24nmLiDBR8/RLN21nmLiDBR8 registers = xxxx $xxxx_B$
- Header transmission → response transmission started
  Set the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started).
  The LIN 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 master interface.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN24nmLiIDB / RLN21nmLiIDB, RLN24nmLiDBRb / RLN21nmLiDBRb, and RLN24nmLiCBR / RLN21nmLiCBR 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 / RLN21nmLiTRC 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 / RLN21nmLiTRC register is cleared.

NOTE

x: Don't care

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The LCKS bit in the RLN24nGLBRP0 / RLN21nGLBRP0 register, RLN24nGLBRP1 / RLN21nGLBRP1 register, and RLN24nmLiMD / RLN21nmLiMD register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Exception/Interrupts**.

16.15.3 Reception in LIN Self-Test Mode

To execute a self-test on LIN master transmission, perform the procedure below:

- Set the baud rate related registers.
  RLN24nGLBRP0 / RLN21nGLBRP0 register = xxxx $xxxx_B$*1
  RLN24nGLBRP1 / RLN21nGLBRP1 register = xxxx $xxxx_B$*1
  RLN24nmLiMD / RLN21nmLiMD register = 0000 $xx00_B$*1
- Set the interrupt enable and error enable related registers.
  RLN24nmLiIE / RLN21nmLiIE register = 0000 $0xxx_B$*2
  RLN24nmLiEDE / RLN21nmLiEDE register = 0000 $x0xx_B$
- Set the break field and space related registers.
  RLN24nmLiBFC / RLN21nmLiBFC register = 00xx $xxxx_B$
  RLN24nmLiSC / RLN21nmLiSC register = 00xx $0xxx_B$*1
- Cancel the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN24nmLiCUC / RLN21nmLiCUC register, and check that the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register are 11_B.
- Set the receive frame related registers.
  RLN24nmLiDFC / RLN21nmLiDFC register = 00x0 $xxxx_B$
  RLN24nmLiIDB / RLN21nmLiIDB register = xxxx $xxxx_B$
  RLN24nmLiDBR1 / RLN21nmLiDBR1 to RLN24nmLiDBR8/RLN21nmLiDBR8 registers = xxxx $xxxx_B$
  RLN24nmLiCBR / RLN21nmLiCBR register = xxxx $xxxx_B$
  Since the checksum value to be transmitted is not automatically calculated, set the calculation value to the RLN24nmLiCBR / RLN21nmLiCBR register. If an incorrect checksum is set at this time, the checksum error can be tested.
- Header transmission → response reception started
  Set the FTS bit in the RLN24nmLiTRC / RLN21nmLiTRC register to 1 (frame transmission or wake-up transmission/reception started).
  The LIN self-test mode (reception) is executed. In this mode, interrupt are generated, and status and error status are also updated.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN24nmLiIDB / RLN21nmLiIDB, RLN24nmLiDBRb / RLN21nmLiDBRb, and RLN24nmLiCBR / RLN21nmLiCBR 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 / RLN21nmLiTRC 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 / RLN21nmLiTRC register is cleared.

NOTE

x: Don't care

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The LCKS bit in the RLN24nGLBRP0 / RLN21nGLBRP0 register, the RLN24nGLBRP1 / RLN21nGLBRP1 register, and the RLN24nmLiMD / RLN21nmLiMD register, and the IBS bit

and IBHS bit (response space only) in the RLN24nmLiSC / RLN21nmLiSC register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in **Section 6, Exception/Interrupts**.

16.15.4 Terminating LIN Self-Test Mode

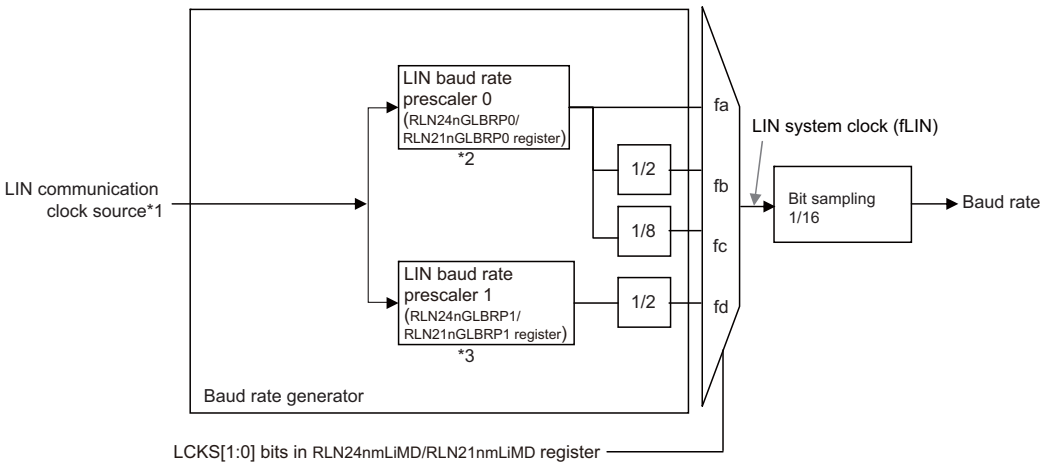
To terminate LIN self-test mode, perform the procedure below:

- All the channels in the unit make a transition to LIN reset mode
  Write 0 to the OM0 bit in the RLN24nmLiCUC / RLN21nmLiCUC register to make a transition to LIN reset mode. However, if the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST 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 / RLN21nmLiCUC register in any one channel. Check that the OMM1 and OMM0 bits in the RLN24nmLiMST / RLN21nmLiMST register are set to 11_B, and then make a transition to LIN reset mode.
- Verify the cancelation of LIN self-test mode.
  Read the LSTM bit in the RLN24nGLSTC / RLN21nGLSTC register; confirm that it is not 0 (not in LIN self-test mode)
- Verify the transition to LIN reset mode.
  Read the OMM0 bit in the RLN24nmLiMST / RLN21nmLiMST register; verify that it is 0 (in LIN reset mode).

16.16 Baud Rate Generator

The LIN system clock (fLIN) is the clock that is made by dividing the LIN communication clock source by the baud rate generator, and the baud rate is made by dividing that clock by 16. The inverse of this baud rate is the bit time (Tbit).

Figure 16.16 shows a block diagram of baud rate generation.



Note 1. For the LIN communication clock source, see **10.5.3, Clock Domain Settings**.
Note 2. According to the RLN24nGLBRP0 / RLN21nGLBRP0 register, the clock frequency is divided by N + 1 (N = 0 to 255).
Note 3. According to the RLN24nGLBRP1 / RLN21nGLBRP1 register, the clock frequency is divided by M + 1 (M = 0 to 255).

Figure 16.16 Block Diagram of Baud Rate Generation

Set the LIN communication clock source to the range from 4 MHz to 40 MHz.

By setting the RLN24nGLBRP0 / RLN21nGLBRP0 register so that fa is 307200 Hz (= 19200 × 16), the resulting bit rates are fa = 19200 × 16, fb = 9600 × 16, and fc = 2400 × 16. These bit rates are frequency-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 RLN24nGLBRP1 / RLN21nGLBRP1 register so that fd is 166672 Hz (= 10417 × 16), the resulting bit rate is fd = 10417 × 16. This bit rate is frequency-divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

The formula for baud rate is as shown below.

Baud rate:

= {Frequency of LIN communication clock source} ÷ ((RLN24nGLBRP0 or RLN21nGLBRP0) + 1) ÷ 16 [bps] (When fa is selected)

= {Frequency of LIN communication clock source} ÷ ((RLN24nGLBRP0 or RLN21nGLBRP0) + 1) ÷ 2 ÷ 16 [bps] (When fb is selected)

= {Frequency of LIN communication clock source} ÷ ((RLN24nGLBRP0 or RLN21nGLBRP0) + 1) ÷ 8 ÷ 16 [bps] (When fc is selected)

= {Frequency of LIN communication clock source} ÷ ((RLN24nGLBRP1 or RLN21nGLBRP1) + 1) ÷ 2 ÷ 16 [bps] (When fd is selected)

Section 17 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/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RLIN3.

17.1 Features of RH850/F1L RLIN3

17.1.1 Number of Units and Channels

This microcontroller has the following number of RLIN3 units.

RLIN3 unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section

Table 17.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of units	1	2	3	4	6	6
Name	RLIN3n (n = 0)	RLIN3n (n = 0, 1)	RLIN3n (n = 0 to 2)	RLIN3n (n = 0 to 3)	RLIN3n (n = 0 to 5)	RLIN3n (n = 0 to 5)

Table 17.2 Unit Configurations and Channels

Unit Name (Channel Name) RLIN3n	Channels per Unit	RH850/F1L 48 pins (1 ch)	RH850/F1L 64 pins (2 ch)	RH850/F1L 80 pins (3 ch)	RH850/F1L 100 pins (4 ch)	RH850/F1L 144 pins (6 ch)	RH850/F1L 176 pins (6 ch)
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 17.3 Index

Index	Meaning
n	Throughout this section, the individual RLIN3 units are identified by the index “n” (n = 0 to 5): for example, RLIN3nLCUC 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 corresponding to each product.

Table 17.4 Index Correspondence of Each Product

Index Correspondence of Each Product
All products
b = 1 to 8

17.1.2 Register Base Address

RLIN3 base addresses are listed in the following table.

RLIN3 register addresses are given as offsets from the base addresses in general.

Table 17.5 Register Base Address

Base Address Name	Base Address
<RLIN30_base>	FFCF 0000$_H$
<RLIN31_base>	FFCF 0040$_H$
<RLIN32_base>	FFCF 0080$_H$
<RLIN33_base>	FFCF 00C0$_H$
<RLIN34_base>	FFCF 0100$_H$
<RLIN35_base>	FFCF 0140$_H$

17.1.3 Clock Supply

The RLIN3 clock supply is shown in the following table.

Table 17.6 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
RLIN3n	LIN communication clock sources	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 to the range from 4 MHz to 40 MHz.

17.1.4 Interrupt Request

RLIN3 interrupt requests are listed in the following table.

Table 17.7 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
RLIN30			
INTRLIN3n (n = 0)	RLIN30 interrupt	25	—
INTRLIN3nUR0 (n = 0)	RLIN30 transmit interrupt	26	10 (channels 0 to 7)
INTRLIN3nUR1 (n = 0)	RLIN30 receive completion interrupt	27	11 (channels 0 to 7)
INTRLIN3nUR2 (n = 0)	RLIN30 status interrupt	28	—
RLIN31			
INTRLIN3n (n = 1)	RLIN31 interrupt	112	—
INTRLIN3nUR0 (n = 1)	RLIN31 transmit interrupt	113	22 (channels 8 to 15)
INTRLIN3nUR1 (n = 1)	RLIN31 receive completion interrupt	114	23 (channels 8 to 15)
INTRLIN3nUR2 (n = 1)	RLIN31 status interrupt	115	—
RLIN32			
INTRLIN3n (n = 2)	RLIN32 interrupt	156	—
INTRLIN3nUR0 (n = 2)	RLIN32 transmit interrupt	157	44 (channels 0 to 7)
INTRLIN3nUR1 (n = 2)	RLIN32 receive completion interrupt	158	45 (channels 0 to 7)
INTRLIN3nUR2 (n = 2)	RLIN32 status interrupt	159	—
RLIN33			
INTRLIN3n (n = 3)	RLIN33 interrupt	220	—
INTRLIN3nUR0 (n = 3)	RLIN33 transmit interrupt	221	47 (channels 8 to 15)
INTRLIN3nUR1 (n = 3)	RLIN33 receive completion interrupt	222	48 (channels 8 to 15)
INTRLIN3nUR2 (n = 3)	RLIN33 status interrupt	223	—
RLIN34			
INTRLIN3n (n = 4)	RLIN34 interrupt	224	—
INTRLIN3nUR0 (n = 4)	RLIN34 transmit interrupt	225	50 (channels 0 to 7)
INTRLIN3nUR1 (n = 4)	RLIN34 receive completion interrupt	226	51 (channels 0 to 7)
INTRLIN3nUR2 (n = 4)	RLIN34 status interrupt	227	—
RLIN35			
INTRLIN3n (n = 5)	RLIN35 interrupt	228	—
INTRLIN3nUR0 (n = 5)	RLIN35 transmit interrupt	229	49 (channels 8 to 15)
INTRLIN3nUR1 (n = 5)	RLIN35 receive completion interrupt	230	50 (channels 8 to 15)
INTRLIN3nUR2 (n = 5)	RLIN35 status interrupt	231	—

17.1.5 Reset Sources

RLIN3 reset sources are listed in the following table. RLIN3 is initialized by these reset sources.

Table 17.8 Reset Sources

Unit Name	Reset Source
RLIN3n	All reset sources (ISORES)

17.1.6 External Input/output Signals

External input/output signals of RLIN3 are listed below.

Table 17.9 External Input/Output Signals

Unit Signal Name	Outline	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

17.2 Overview

17.2.1 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 17.10 gives the LIN/UART interface specifications.

Table 17.10 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 is 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 frame/wake-up transmission • Successful frame/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 17.10 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 rate 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 frame/wake-up transmission • Header/frame/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 bits; 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 9 bits including the expansion bit supported.	
		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	
	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 baud rate generator incorporated, 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 17.10 LIN/UART Interface Specifications (3/3)

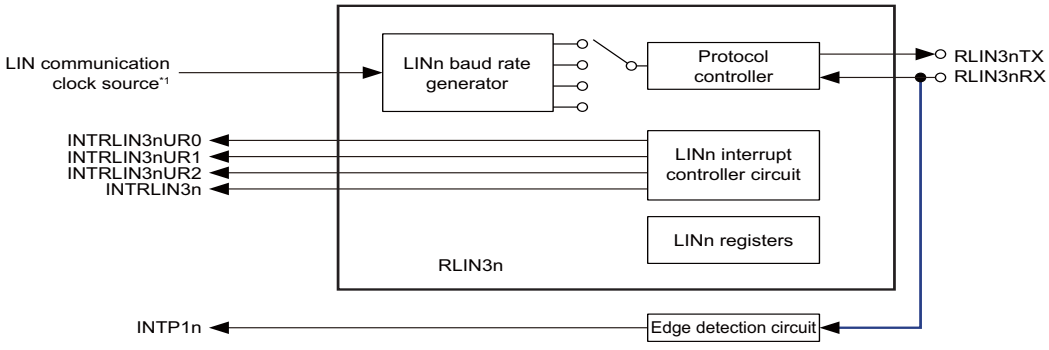
Item		Specifications
UART communication function	The stop bit received is guaranteed (start of transmission can be delayed when start of transmission is attempted during reception of the stop bit).	
	Interrupt function	• Transmission start/complete • Reception complete • Status/error detection

Note 1. Since the same register is used for setting, the inter-byte space (header) = response space.

Note 2. For wake-up reception, the low level width of the input signal is indicated.

17.2.2 Block Diagram

Figure 17.1 shows a block diagram of the LIN/UART interface.



Note 1. For the LIN communication clock source, see **Section 10.5.3, Clock Domain Settings**.

Figure 17.1 LIN/UART Interface Block Diagram

17.2.3 Description of Blocks

- RLIN3nTX, RLIN3nRX: LIN/UART interface I/O pins
- LINn baud rate generator: Generates the LIN/UART interface communication clock signal.
- LINn registers: LIN/UART interface registers
- LINn interrupt controller circuit: Controls interrupt requests generated by the LIN/UART interface

17.3 Registers

17.3.1 List of Registers

RLIN3 registers are listed in the following table.

For details about <RLIN3n_base>, see **Section 17.1.2, Register Base Address**.

Table 17.11 Registers (1/2)

Module	Register	Symbol	Address	LIN Master	LIN Slave	UART
RLIN3n	LIN wake-up baud rate selector register	RLN3nLWBR	<RLIN3n_base> + 01_H	√	√	√
RLIN3n	LIN/ UART baud rate prescaler 01 register	RLN3nLBRP01	<RLIN3n_base> + 02_H	—	√	√
RLIN3n	LIN/ UART baud rate prescaler 0 register	RLN3nLBRP0	<RLIN3n_base> + 02_H	√	√	√
RLIN3n	LIN/ UART baud rate prescaler 1 register	RLN3nLBRP1	<RLIN3n_base> + 03_H	√	√	√
RLIN3n	LIN self-test control register	RLN3nLSTC	<RLIN3n_base> + 04_H	√	√	—
RLIN3n	LIN/ UART mode register	RLN3nLMD	<RLIN3n_base> + 08_H	√	√	√
RLIN3n	LIN break field configuration register/ UART configuration register	RLN3nLBFC	<RLIN3n_base> + 09_H	√	√	√
RLIN3n	LIN / UART space configuration register	RLN3nLSC	<RLIN3n_base> + $0A_H$	√	√	√
RLIN3n	LIN wake-up configuration register	RLN3nLWUP	<RLIN3n_base> + $0B_H$	√	√	—
RLIN3n	LIN interrupt enable register	RLN3nLIE	<RLIN3n_base> + $0C_H$	√	√	—
RLIN3n	LIN / UART error detection enable register	RLN3nLEDE	<RLIN3n_base> + $0D_H$	√	√	√
RLIN3n	LIN/ UART control register	RLN3nLCUC	<RLIN3n_base> + $0E_H$	√	√	√
RLIN3n	LIN / UART transmission control register	RLN3nLTRC	<RLIN3n_base> + 10_H	√	√	√
RLIN3n	LIN/ UART mode status register	RLN3nLMST	<RLIN3n_base> + 11_H	√	√	√
RLIN3n	LIN / UART status register	RLN3nLST	<RLIN3n_base> + 12_H	√	√	√
RLIN3n	LIN/ UART error status register	RLN3nLEST	<RLIN3n_base> + 13_H	√	√	√
RLIN3n	LIN data field configuration register	RLN3nLDFC	<RLIN3n_base> + 14_H	√	√	√
RLIN3n	LIN / UART ID buffer register	RLN3nLIDB	<RLIN3n_base> + 15_H	√	√	√
RLIN3n	LIN checksum buffer register	RLN3nLCBR	<RLIN3n_base> + 16_H	√	√	—
RLIN3n	UART data 0 buffer register	RLN3nLUDB0	<RLIN3n_base> + 17_H	—	—	√
RLIN3n	LIN / UART data buffer 1 register	RLN3nLDBR1	<RLIN3n_base> + 18_H	√	√	√
RLIN3n	LIN/ UART data buffer 2 register	RLN3nLDBR2	<RLIN3n_base> + 19_H	√	√	√
RLIN3n	LIN/ UART data buffer 3 register	RLN3nLDBR3	<RLIN3n_base> + $1A_H$	√	√	√
RLIN3n	LIN / UART data buffer 4 register	RLN3nLDBR4	<RLIN3n_base> + $1B_H$	√	√	√
RLIN3n	LIN / UART data buffer 5 register	RLN3nLDBR5	<RLIN3n_base> + $1C_H$	√	√	√
RLIN3n	LIN / UART data buffer 6 register	RLN3nLDBR6	<RLIN3n_base> + $1D_H$	√	√	√
RLIN3n	LIN / UART data buffer 7 register	RLN3nLDBR7	<RLIN3n_base> + $1E_H$	√	√	√
RLIN3n	LIN/ UART data buffer 8 register	RLN3nLDBR8	<RLIN3n_base> + $1F_H$	√	√	√
RLIN3n	UART operation enable register	RLN3nLUOER	<RLIN3n_base> + 20_H	—	—	√
RLIN3n	UART option register 1	RLN3nLUOR1	<RLIN3n_base> + 21_H	—	—	√
RLIN3n	UART transmission data register	RLN3nLUTDR	<RLIN3n_base> + 24_H	—	—	√
RLIN3n	UART transmission data register L	RLN3nLUTDRL	<RLIN3n_base> + 24_H	—	—	√
RLIN3n	UART transmission data register H	RLN3nLUTDRH	<RLIN3n_base> + 25_H	—	—	√
RLIN3n	UART reception data register	RLN3nLURDR	<RLIN3n_base> + 26_H	—	—	√
RLIN3n	UART reception data register L	RLN3nLURDRL	<RLIN3n_base> + 26_H	—	—	√

Table 17.11 Registers (2/2)

Module	Register	Symbol	Address	LIN Master	LIN Slave	UART
RLIN3n	UART reception data register H	RLN3nLURDRH	<RLIN3n_base> + 27_H	—	—	√
RLIN3n	UART wait transmission data register	RLN3nLUWTDR	<RLIN3n_base> + 28_H	—	—	√
RLIN3n	UART wait transmission data register L	RLN3nLUWTDRL	<RLIN3n_base> + 28_H	—	—	√
RLIN3n	UART wait transmission data register H	RLN3nLUWTDRH	<RLIN3n_base> + 29_H	—	—	√

Note: √: Used, —: Not used
When writing to a register not used, write the value after reset.

17.3.2 LIN Master Related Registers

17.3.2.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read/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 17.12 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 sampling 1 1 1 1: 16 sampling 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. (for 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. (for LIN2.x)

Set the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (in 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 sampling).

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 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. Setting the LWBR0 bit to 1 selects the LIN system clock (fLIN) as fa regardless of the setting of the RLN3nLMD.LCKS bit

(the LCKS bit is not changed). This allows the 2.5-Tbit or longer low-level width of the input signal to 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 RLN3nLMD.LCKS bit.

17.3.2.2 RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register

Access: This register can be read/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 17.13 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

Set the RLN3nLBRP0 register when the OMM0 bit in the RLN3nLMST register is 0_B (in 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 by N + 1.

17.3.2.3 RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register

Access: This register can be read/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 17.14 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 RLN3LMST register is 0_B (in 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.

17.3.2.4 RLN3nLSTC — LIN Self-Test Control Register

Access: This register can be read/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 17.15 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 (in 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 17.9, 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 leaving LIN self-test mode, see **Section 17.9, 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.

17.3.2.5 RLN3nLMD — LIN Mode Register

Access: This register can be read/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 17.16 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filtering 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 (in LIN reset mode)

LRDNFS Bit (LIN Reception Data Noise Filtering 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 17.4, 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).

With 1_B is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x is used), and the RLN3nLMST register is 01_H (LIN wake-up mode), the protocol controller is provided with fa regardless of the setting of the 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.

17.3.2.6 RLN3nLBFC — LIN Break Field Configuration Register

Access: This register can be read/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 17.17 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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 (in 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)

This bit is used to set the break 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)

This BLT bits set the break low level width of transmission frame header.
13 Tbits to 28 Tbits can be set.

17.3.2.7 RLN3nLSC — LIN Space Configuration Register

Access: This register can be read/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 17.18 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to this bit, 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 (in 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.

17.3.2.8 RLN3nLWUP — LIN Wake-Up Configuration Register

Access: This register can be read/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 17.19 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 to these bits, write the value after reset.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (in 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 is used), fa is selected as the LIN system clock (fLIN) regardless of the setting of the RLN3nLMD.LCKS bit (the LCKS bit is not changed).

17.3.2.9 RLN3nLIE — LIN Interrupt Enable Register

Access: This register can be read/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 17.20 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	SHIE	Successful Header Transmission Interrupt Enable 0: Disables successful header transmission interrupt. 1: Enables successful header transmission interrupt.
2	ERRIE	Error Detection Interrupt Enable 0: Disables error detection interrupt. 1: Enables error detection interrupt.
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Enable 0: Disables successful frame/wake-up reception interrupt. 1: Enables successful frame/wake-up reception interrupt.
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Enable 0: Disables successful frame/wake-up transmission interrupt. 1: Enables successful frame/wake-up transmission interrupt.

Set the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

SHIE Bit (Successful Header Transmission Interrupt Enable)

The SHIE bit enables or disables interrupt request 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 Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.
With 0 set, the interrupt request for RLIN3n status 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.
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 Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a frame or a wake-up signal (counting of low level width of the input signal).
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 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.

17.3.2.10 RLN3nLEDE —LIN Error Detection Enable Register

Access: This register can be read/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 17.21 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 to these bits, write the value after reset.
3	FERE	Framing Error Detection Enable 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 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 (in LIN reset mode)

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.
With 0 set, the timeout function applies to frame timeout.
With 1 set, the timeout function applies to response timeout.
For details on the timeout error, see **Section 17.7.7, 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.
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 17.7.7, 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 17.7.7, 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 17.7.7, 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.
When this bit is set to 1, the detection result is indicated in the BER flag in the RLN3nLEST register.
For details on the bit error, see **Section 17.7.7, Error Status**.

17.3.2.11 RLN3nLCUC — LIN Control Register

Access: This register can be read/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 17.22 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode is caused. 1: LIN operation mode is caused.
0	OM0	LIN Reset 0: LIN reset mode is caused. 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to cause a transition to LIN wake-up mode after canceling LIN reset mode, and set the register to 03_H to cause a transition to LIN operation 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 is caused.
With 1 set, LIN operation mode is caused.
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 is caused.
With 1 set, LIN reset mode is canceled.

17.3.2.12 RLN3nLTRC — LIN Transmission Control Register

Access: This register can be read/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 17.23 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 of the RLN3nLMST register is 0_B (in 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 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 (counting of the low level width of the input signal).

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 (in LIN reset mode).

This bit is set to 0 upon error detection of frame or wake-up communication (including error detection) and transition to LIN reset mode.

17.3.2.13 RLN3nLMST — LIN Mode Status Register

Access: This register can only 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 17.24 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: The module is in LIN wake-up mode. 1: The module is in LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: The module is in LIN reset mode. 1: The module is 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.

17.3.2.14 RLN3nLST — LIN Status Register

Access: This register can be read/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 17.25 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 These bits are always read as 0. The write value should always be 0.
5, 4	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to this bit, write the value after reset.
1	FRC	Frame/Wake-up Reception Complete 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 Complete 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. Here, an interrupt 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 the value of 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 (Frame/Wake-up Reception Complete 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 (Frame/Wake-up Transmission Complete 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.

17.3.2.15 RLN3nLEST — LIN Error Status Register

Access: This register can be read/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 17.26 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 to this bit, 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 to this bit, 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.

17.3.2.16 RLN3nLDFC — LIN Data Field Configuration Register

Access: This register can be read/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 17.27 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 mode 1: Enhanced checksum mode
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 the RTS bit in the RLN3nLTRC register being set.

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 17.7.4.1, 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 17.7.7, 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 bits set 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 (low level width of the input signal is counted).
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.

17.3.2.17 RLN3nLIDB — LIN ID Buffer Register

Access: This register can be read/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 17.28 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 17.9, 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 for P0 and IDP1 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 bit sets the 6-bit ID value to be transmitted in the ID field of the LIN frame.

17.3.2.18 RLN3nLCBR — LIN Checksum Buffer Register

Access: This register can only be read in 8-bit units. In LIN self-test mode, this register can be read/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 17.29 RLN3nLCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the checksum value 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.

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 17.9, 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).

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.

17.3.2.19 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

Access: This register can be read/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 17.30 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or allows the received data to be read. Setting range: 00_H to FF_H

- For response transmission:
  The LDBRn 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 bit 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:
  The LDBRn 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 the LDBRn 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 17.9, LIN Self-Test Mode**.

17.3.3 LIN Slave Related Registers

17.3.3.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read/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 17.31 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 sampling 0 0 1 1: 4 sampling 0 1 1 1: 8 sampling 1 1 1 1: 16 sampling 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 to this bit, write the value after reset.

Set the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (in 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 frame 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 sampling).
When the 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 sampling) or "0111_B" (8 sampling).

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.

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).

17.3.3.2 RLN3nLBRP01 — LIN Baud Rate Prescaler 01 Register

Access: RLN3nLBRP01 can be read/written in 16-bit units.
RLN3nLBRP0 can be read/written in 8-bit units.
RLN3nLBRP1 can be read/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 17.32 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 (in 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.

17.3.3.3 RLN3nLSTC — LIN Self-Test Control Register

Access: This register can be read/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	R	R	R	R	R	R	R/W

Table 17.33 RLN3nLSTC Register Contents

Bit Position	Bit Name	Function
7 to 0	Reserved	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 (in 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 17.9, 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 leaving LIN self-test mode, see **Section 17.9, 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.

17.3.3.4 RLN3nLMD — LIN Mode Register

Access: This register can be read/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 17.34 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filtering 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 reception status interrupt are used.
3, 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 (in LIN reset mode).

LRDNFS Bit (LIN Reception Data Noise Filtering 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 17.4, 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).

17.3.3.5 RLN3nLBFC — LIN Break Field Configuration Register

Access: This register can be read/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 17.35 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, 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 (in LIN reset mode).

LBLT Bit (Reception Break (Low-Level) Detection Width Setting)

- When RLN3nLMD.LMD is "10_B" (in 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" (in 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.

17.3.3.6 RLN3nLSC — LIN Space Configuration Register

Access: This register can be read/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 17.36 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to this bit, 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 (in LIN reset mode).

This register is enabled only during response transmission, and disabled during response reception.

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)

These 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.

17.3.3.7 RLN3nLWUP — LIN Wake-Up Configuration Register

Access: This register can be read/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 17.37 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 to these bits, write the value after reset.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

These bits set the low-level width of the wake-up frame transmission.
1 Tbit to 16 Tbits can be set.

17.3.3.8 RLN3nLIE — LIN Interrupt Enable Register

Access: This register can be read/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 17.38 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	SHIE	Successful Header Reception Interrupt Enable 0: Disables successful header reception interrupt. 1: Enables successful header reception interrupt.
2	ERRIE	Error Detection Interrupt Enable 0: Disables error detection interrupt. 1: Enables error detection interrupt.
1	FRCIE	Successful Response/Wake-up Reception Interrupt Enable 0: Disables successful Response/wake-up reception interrupt. 1: Enables successful Response/wake-up reception interrupt.
0	FTCIE	Successful Response/Wake-up Transmission Interrupt Enable 0: Disables successful Response/wake-up transmission interrupt. 1: Enables successful Response/wake-up transmission interrupt.

Set the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

SHIE Bit (Successful Header Reception Interrupt 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 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 Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a response or a wake-up frame (counting of low level width of the input signal).
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 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.

17.3.3.9 RLN3nLEDE — LIN Error Detection Enable Register

Access: This register can be read/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 17.39 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 to this bit, 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 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 to this bit, 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 (in LIN reset mode).

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.
With 0 set, the timeout function applies to frame timeout.
With 1 set, the timeout function applies to response timeout.
For details on the timeout error, see **Section 17.7.7, 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 17.7.7, 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 17.7.7, 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.
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 17.7.7, 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 indicated 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 17.7.7, 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.
When this bit is set to 1, the detection result is indicated in the BER flag in the RLN3nLEST register.
For details on the bit error, see **Section 17.7.7, Error Status**.

17.3.3.10 RLN3nLCUC — LIN Control Register

Access: This register can be read/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 17.40 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode is caused. 1: LIN operation mode is caused.
0	OM0	LIN Reset 0: LIN reset mode is caused. 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to cause a transition to LIN wake-up mode after canceling LIN reset mode, and set the register to 03_H to cause a transition to LIN operation 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 wake-up mode is caused.
With 1 set, LIN operation mode is caused.
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 is caused.
With 1 set, LIN reset mode is canceled.

17.3.3.11 RLN3nLTRC — LIN Transmission Control Register

Access: This register can be read/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 17.41 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, 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)

After receiving the header and checking the received ID, set this bit to 1 if no response is transmitted/received.

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 for the completion of the header. (Do not use this bit for the second or latter data group.)

RTS Bit (Response Transmission/Reception Start)

After receiving the header and checking the received ID, set this bit to 1 at the response transmission or at the start of response reception.

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.
Also set this bit to 1 to allow wake-up reception (counting of the low level width of the input signal).
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 frame or wake-up communication and transition to LIN reset mode.

17.3.3.12 RLN3nLMST — LIN Mode Status Register

Access: This register can only 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 17.42 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: The module is in LIN wake-up mode. 1: The module is in LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: The module is in LIN reset mode. 1: The module is 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.

17.3.3.13 RLN3nLST — LIN Status Register

Access: This register can be read/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 17.43 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Reception Flag 0: Header transmission has not been completed. 1: Header transmission has been completed.
6	D1RC	Successful Data 1 Reception Flag These bits are always read as 0. The write value should always be 0.
5, 4	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to this bit, 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 the value of 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 bit to 0, write 0 to the RPER, IPER, CSER, SFER, FER, TER, and BER flags in the RLN3nLEST register.

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 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 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.

17.3.3.14 RLN3nLEST — LIN Error Status Register

Access: This register can be read/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 17.44 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 to this bit, 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. This 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.

17.3.3.15 RLN3nLDFC — LIN Data Field Configuration Register

Access: This register can be read/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 17.45 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 to this bit, 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 it to “0”.
This should be set 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 17.7.7, Error Status**.
When the length of the response field data is 0 byte (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.
This should be set 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 (low-level width of the input signal is counted).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.
This bit should be set 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.
These bits should be set 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, 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.

17.3.3.16 RLN3nLIDB — LIN/UART ID Buffer Register

Access: This register can be read/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 17.46 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7, 6	IDP[1:0]	Parity Setting Stores the parity bits (P0 and P1) to be received in the ID field.
5 to 0	ID[5:0]	ID Setting Stores the 6-bit ID value to be 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 17.9, LIN Self-Test Mode**.

IDP[1:0] Bits (Parity Setting)

The IDP bits store the parity bits (P0 and P1) to be 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 enable), 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 to be received in the ID field of the LIN frame.

17.3.3.17 RLN3nLCBR — LIN Checksum Buffer Register

Access: This register can be read/written 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 17.47 RLN3nLCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the checksum value transmitted or received.

In LIN 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 17.9, 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).

17.3.3.18 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

Access: This register can be read/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 17.48 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or allows the received data to be read. Setting range: 00_H to FF_H

- For response transmission:
  The RLN3nLDBRb registers set the data to be transmitted in the response field.
  These registers should be set 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 17.9, LIN Self-Test Mode**.

17.3.4 UART Related Registers

17.3.4.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read/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 17.49 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 sampling 0 1 0 1: 6 sampling 0 1 1 0: 7 sampling 0 1 1 1: 8 sampling 1 0 0 0: 9 sampling 1 0 0 1: 10 sampling 1 0 1 0: 11 sampling 1 0 1 1: 12 sampling 1 1 0 0: 13 sampling 1 1 0 1: 14 sampling 1 1 1 0: 15 sampling 1 1 1 1: 16 sampling 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 to this bit, write the value after reset.

Set the LN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (in 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 bit rate).
In UART mode, it is possible to set the NSPB bits from 6 sampling to 16 sampling.

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.

17.3.4.2 RLN3nLBRP01 — UART Baud Rate Prescaler 01 Register

Access: RLN3nLBRP01 register can be read/written in 16-bit units.
RLN3nLBRP0 register can be read/written in 8-bit units.
RLN3nLBRP1 register can be read/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 17.50 RLN3nLBRP01 Register Contents

Bit Position	Bit Name	Function
15 to 0	LBRP0[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 (in 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.

17.3.4.3 RLN3nLMD — UART Mode Register

Access: This register can be read/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 17.51 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	LRDNFS	UART Reception Data Noise Filtering 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 to these bits, 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 (in LIN reset mode).

LRDNFS Bit (UART Reception Data Noise Filtering 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.

17.3.4.4 RLN3nLBFC — UART Configuration Register

Access: This register can be read/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 17.52 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6	UTPS	UART Output Polarity Switch 0: Transmit data normal output 1: Transmit data with inversion output
5	URPS	UART Input Polarity Switch 0: Reception data normal output 1: Reception data with inversion output
4, 3	UPS[1:0]	UART Parity Select 00: Parity prohibited 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 (in 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, receive data is output with inversion.
The setting of this bit is valid in all the bits of the UART frame.
In half-duplex communication, this setting should match with 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 in all the bits of the UART frame.
In half-duplex communication, this setting should match with 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 reversed value of the expected value to be received as expected value to the UEBDL bit in the RLN3nLUOR1 register and RLN3nLIDB register for comparison of the reversed 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 communicated 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", data is communicated 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", data is communicated with 0 parity.

  [Transmission]

  Regardless of the number 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", data is communicated 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 ignored.

17.3.4.5 RLN3nLSC — UART Space Configuration Register

Access: This register can be read/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 17.53 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, write the value after reset.

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS bits set the width of the space between the UART frame in UART buffer transmit
0 Tbit to 3 Tbits can be set.

17.3.4.6 RLN3nLEDE —UART Error Detection Enable Register

Access: This register can be read/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 17.54 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to this bit, 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 (in 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 17.7.7, 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 17.7.7, 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.
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 is 0101_B (6 sampling) and the LRDNFS bit in the RLN3nLMD register is 0 (noise filter is enabled).

For details on the bit error, see **Section 17.7.7, Error Status**.

17.3.4.7 RLN3nLCUC — UART Control Register

Access: This register can be read/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 17.55 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	OM0	LIN Reset 0: LIN reset mode is caused. 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 is caused.
With 1 set, reset mode is canceled.

17.3.4.8 RLN3nLTRC — UART Transmission Control Register

Access: This register can be read/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 17.56 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to this bit, 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 enable) 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 (in LIN reset mode).
When writing 1 to this bit while the UTSW bit in the RLN3nLRFC 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.

17.3.4.9 RLN3nLMST — UART Mode Status Register

Access: This register can only 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 17.57 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: The module is in LIN reset mode. 1: The module is not in LIN reset mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

17.3.4.10 RLN3nLST — UART Status Register

Access: This register can be read/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 17.58 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, 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 conditions.

- 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 the value of 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, EXBT, OER, and ER flags in the RLN3nLEST register.

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.
Regardless of errors, this bit is set to 1 upon completion of the number of data transmission specified by the MDL bit in the RLN3nLDFC register from the UART buffer. At this time, an interrupt request for RLIN3n transmission is generated.
However, when this bit is 1, an interrupt is not generated upon completion of transmission from the UART buffer. Write 0 to the bit to be cleared.

17.3.4.11 RLN3nLEST — UART Error Status Register

Access: This register can be read/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 17.59 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, 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	Expanded Bit Detection Flag 0: Expanded bit has not been detected. 1: Expanded 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 to this bit, 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 when the module transitions 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 has been retained before 1 is written.
This flag is set to 1 upon parity error detection. To clear the bit, write 0 to the bit.

IDMT Flag (ID Matching 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.

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 (expansion bit/data comparison enabled)
  - The received expansion bit and the value of the UEBDL bit of the RLN3nLUOR1 register are matched.
  - The 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register are matched.

To clear the bit, write 0 to the bit.

EXBT Flag (Expanded Bit Detection 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 UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enable), if the received expansion bit matches with the UEBDL bit in the RLN3nLUOR1 register, this flag is set to 1.
To clear the bit, write 0 to the bit.

FER Flag (Framing 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.
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). To clear the bit, write 0 to the bit.

OER Flag (Overrun 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.
The 0ER flag is set to 1 upon overrun error detection while the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). To clear the bit, write 0 to the 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.
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).
To clear the bit, write 0 to the bit.

17.3.4.12 RLN3nLDFC — UART Data Field Configuration Register

Access: This register can be read/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 17.60 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	UTSW	Transmission Start Wait 0: When UART transmission is requested, transmission is started immediately. 1: When UART 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 to this bit, 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 transmit 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)

This bit specifies 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).

17.3.4.13 RLN3nLIDB — UART ID Buffer Register

Access: This register can be read/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 17.61 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 Bit (ID Bit)

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 (expansion bit/data comparison enabled), set the receive data and the value to be compared. Write to the RLN3nLIDB register when the URS bit in the RLN3nLST register is 0 (receive operation is not in progress).

17.3.4.14 RLN3nLUDB0 — UART Data 0 Buffer Register

Access: This register can be read/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 17.62 RLN3nLUDB0 Register Contents

Bit Position	Bit Name	Function
7 to 0	UDB[7:0]	Transmission data is set

If the data length selection corresponds to 9 data bytes (RLN3nLDFC.MDL bit is "0_H" or "9_H") for multi-byte UART transmission, then the first data value for UART communication is present in this buffer.

Write to the RLN3nLUDB0 register when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

For details about the UART buffer, see **Section 17.8.1.2, UART Buffer Transmission**, **(1) UART Buffer Transmission**.

17.3.4.15 RLN3nLDBRb — UART Data Buffer b Register (b = 1 to 8)

Access: This register can be read/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 17.63 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted.

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).

For details about the UART buffer, see **Section 17.8.1.2, UART Buffer Transmission**, **(1) UART Buffer Transmission**.

17.3.4.16 RLN3nLUOER — UART Operation Enable Register

Access: This register can be read/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	R

Table 17.64 RLN3nLUOER Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 shift to the LIN reset mode. However, the transmit operation is also suspended at this time.

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 shift to the LIN reset mode. However, the receive operation is also suspended at this time.

17.3.4.17 RLN3nLUOR1 — UART Option Register 1

Access: This register can be read/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 17.65 RLN3nLUOR1 Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
4	UECD	Expansion Bit Comparison Disable 0: Enables comparison between the received expansion bit and the UEBDL bit value. 1: Disables comparison between the received expansion bit and the UEBDL bit value.
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 (in 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 (expansion bit/data comparison enable).

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 receive 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 receive 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 (in 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 (in 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 (in LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.

17.3.4.18 RLN3nLUTDR — UART Transmission Data Register

Access: RLN3nLUTDR register can be read/written in 16-bit units.
RLN3nLUTDRL register can be read/written in 8-bit units.
RLN3nLUTDRH register can be read/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 17.66 RLN3nLUTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
8 to 0	UTD[8:0]	Sets the data to be transmitted from the transmission buffer. 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 17.67 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

17.3.4.19 RLN3nLURDR — UART Reception Data Register

Access: RLN3nLURDR register can only be read in 16-bit units.
RLN3nLURDRL register can only be read in 8-bit units.
RLN3nLURDRH register can only 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 17.68 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]	Set the reception 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 at the stop bit of the reception data.

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 detection enabled). The value of this register is updated upon occurrence of an overrun error when the OERE bit is 0 (overrun 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 17.69 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

17.3.4.20 RLN3nLUWTDR — UART Wait Transmission Data Register

Access: RLN3nLUWTDR register can be read/written in 16-bit units.
RLN3nLUWTDRL register can be read/written in 8-bit units.
RLN3nLUWTDRH register can be read/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 17.70 RLN3nLUWTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing to these bits, 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 17.71 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

17.4 Interrupt Sources

The LIN/UART interface generates four types of interrupt requests.

- RLIN3n successful 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 request.

Table 17.72 lists the sources for each interrupt.

Table 17.72 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.

17.5 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 17.2 shows mode transitions. **Table 17.77** describes mode transition conditions. **Table 17.77** lists operations available in each mode.

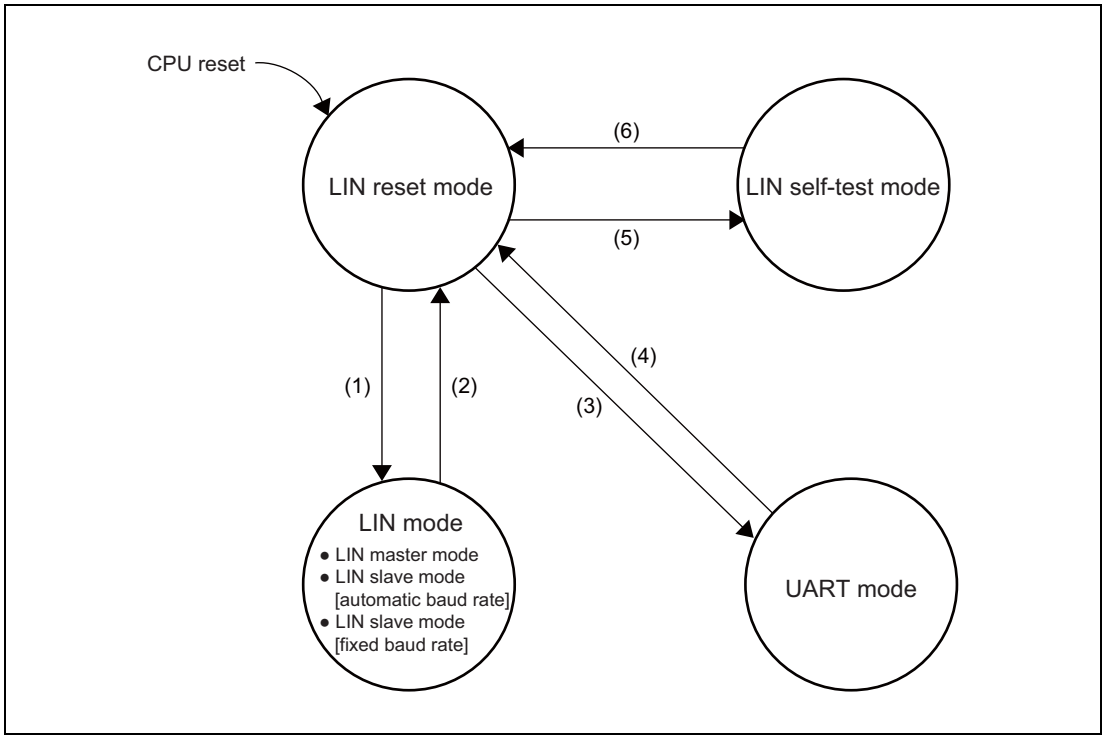


Figure 17.2 Mode Transitions

Table 17.73 Transition Condition of Each Mode

	Mode Transition			Transition Condition
(1)	LIN reset mode	→	LIN mode	
			• LIN master mode	• LMD bit 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 bit 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 bit in RLN3nLMD register = 10_B and OM0 and OM1 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 bit 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 17.9, LIN Self-Test Mode**.
(6)	LIN self-test mode	→	LIN reset mode	See **Section 17.9, LIN Self-Test Mode**.

Table 17.74 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 a transition has been caused to LIN reset mode, the LIN mode, or the UART mode can be verified by reading the LMD bits in the RLN3nLMD register or the OMM0 bit in the RLN3nLMST register.

For a description of the LIN self-test mode, see **Section 17.9, LIN Self-Test Mode**.

17.6 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

17.7 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 the range from 1kbps to 20kbps.

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).

In LIN slave mode [fixed baud rate] allows automatic detection of the break field, the sync field, and the ID filed 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 bit 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 17.3 shows the transition of operation modes. **Table 17.75** describes the transition conditions of operation modes.

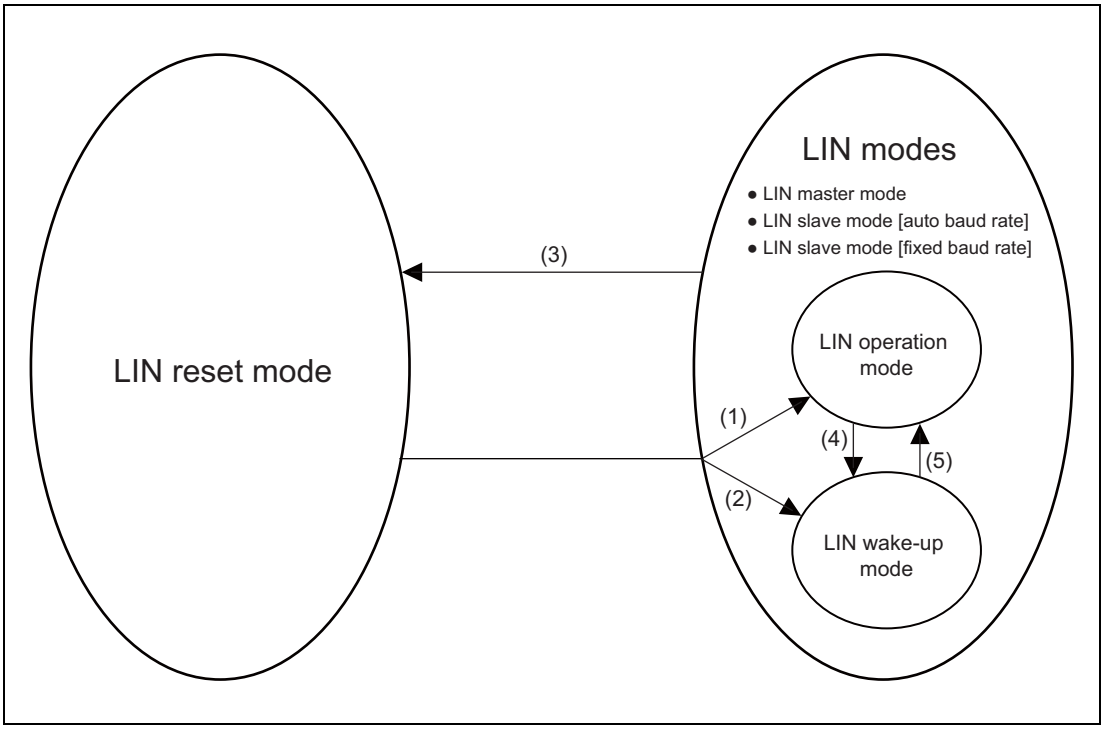


Figure 17.3 Transition of Operation Modes

Table 17.75 Transition Condition for Operation Mode

	Operation Mode Transition			Transition Condition
(1)	LIN reset mode	→	LIN mode • LIN operation mode	LMD bit 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 bit 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 going on (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.

17.7.1 LIN Master Mode

17.7.1.1 Header Transmission

Figure 17.4 shows the operation of the LIN/UART interface (LIN master mode) in header transmission. **Table 17.76** provides processing in header transmission.

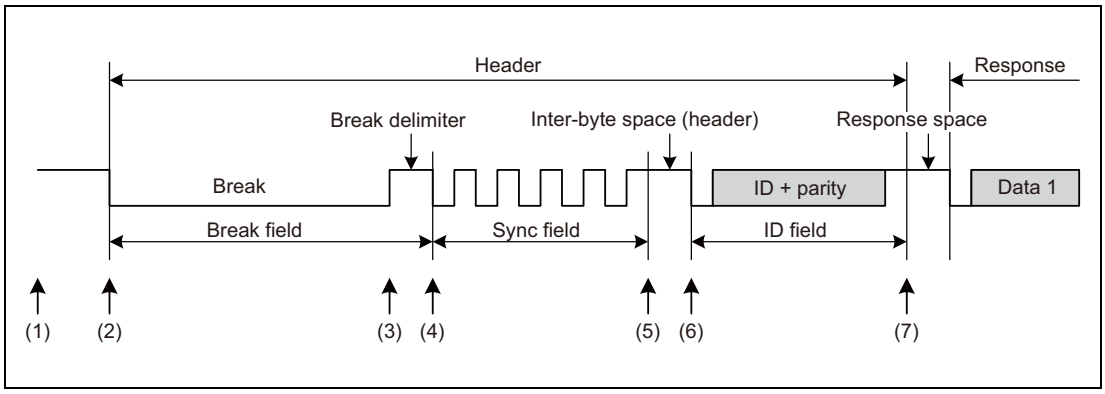


Figure 17.4 Operation in Header Transmission

Table 17.76 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 17.7.7, Error Status**.

17.7.1.2 Response Transmission

Figure 17.5shows the operation of the LIN/UART interface (LIN master mode) in response transmission. **Table 17.77** provides processing in response transmission.

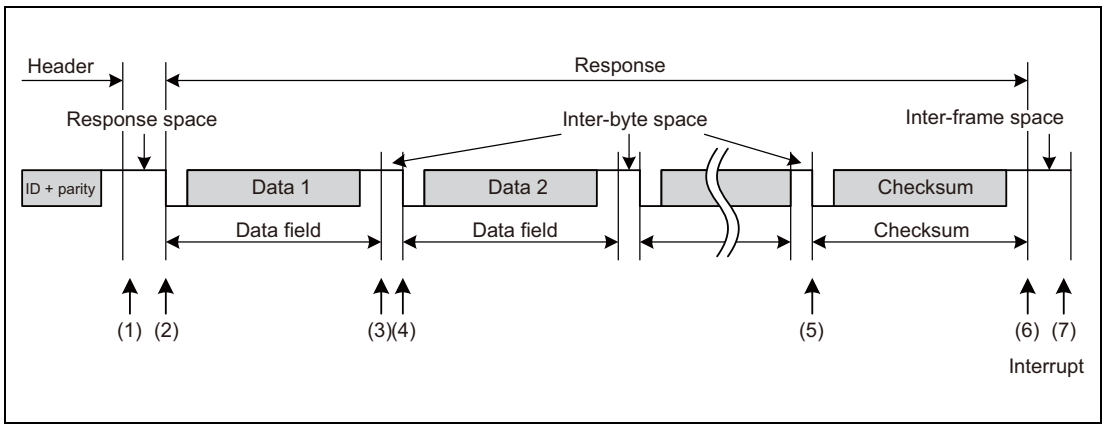


Figure 17.5 Operation in Response Transmission

Table 17.77 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 the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RFC 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, see **Section 17.7.7, Error Status**.

17.7.1.3 Response Reception

Figure 17.6 shows the operation of the LIN/UART interface (LIN master mode) on response reception. **Table 17.78** provides processing in response reception.

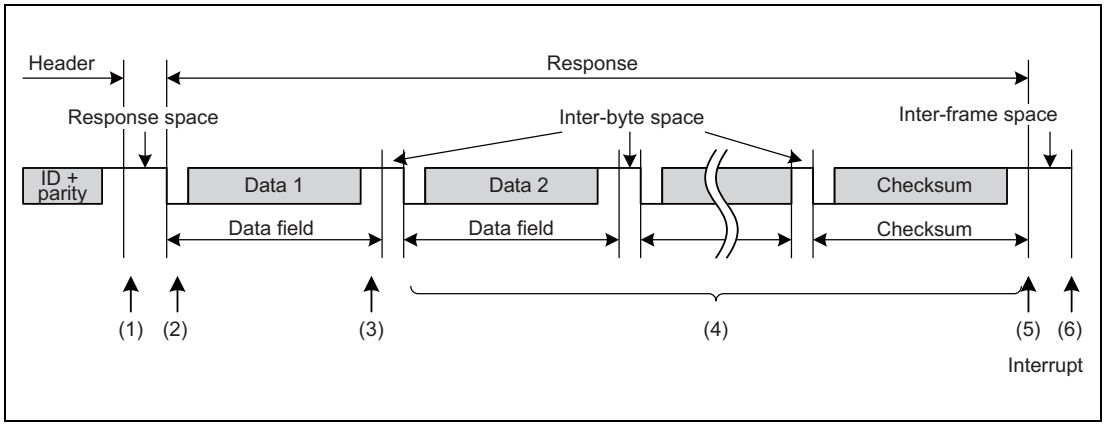


Figure 17.6 Operation in Response Reception

Table 17.78 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 the transmission of inter-byte spaces 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 17.7.7, Error Status**.

17.7.2 LIN Slave Mode

17.7.2.1 Header Reception

Figure 17.7 shows the operation of the LIN/UART interface (LIN slave mode) in header reception.
Table 17.79 provides processing in header reception.

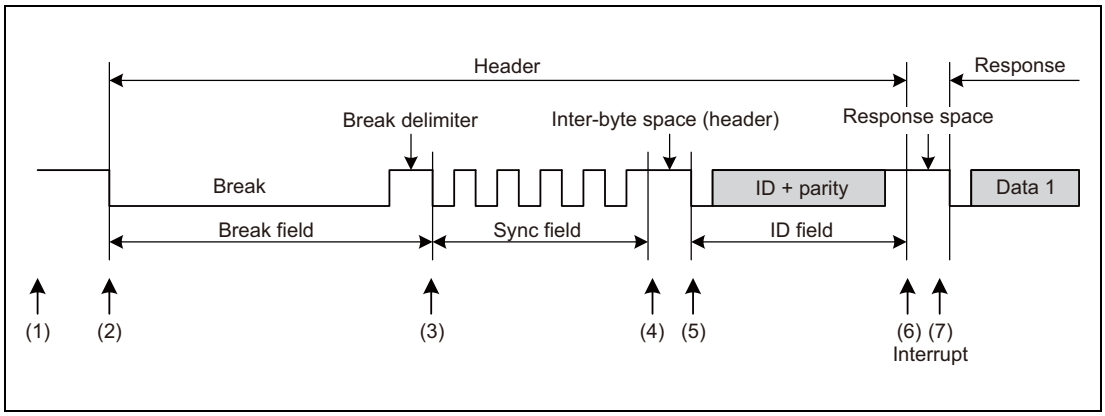


Figure 17.7 Operation in Header Reception

Table 17.79 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. In the case of break field detection (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) • Baud rate generator setting (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. For information about error detection conditions, see **Section 17.7.7, 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 calculated from the average of the starting 2 bits (the period of the consecutive fall 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).

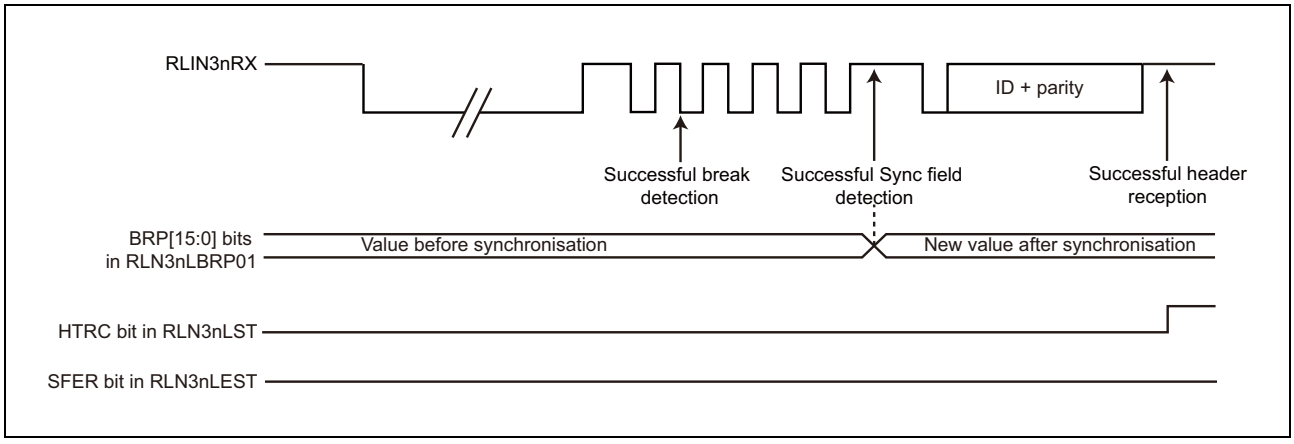


Figure 17.8 Header Reception in LIN Slave Mode [Auto Baud Rate] (in Normal Operation)

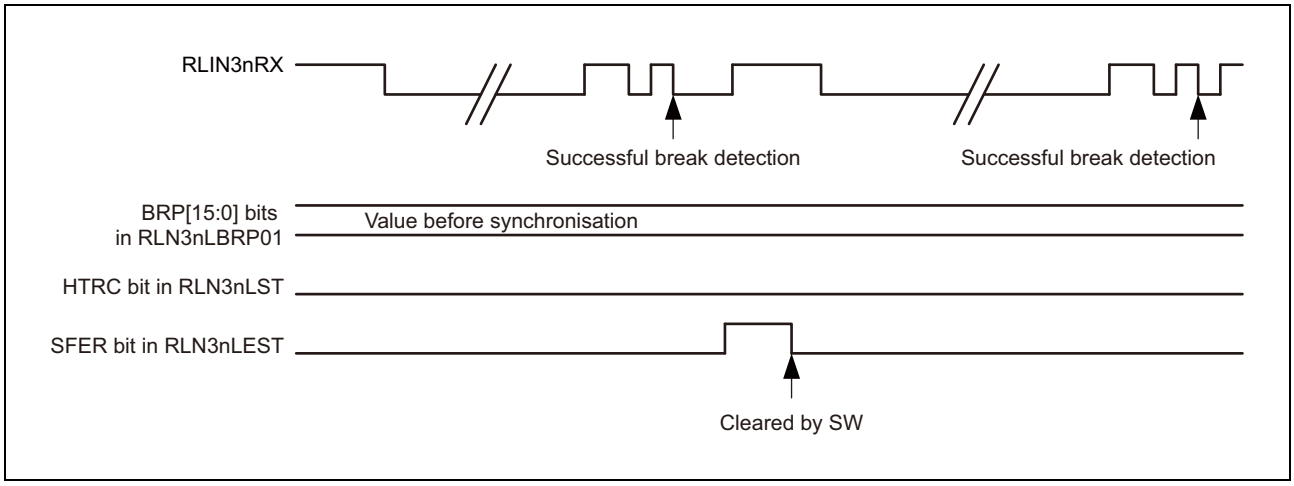


Figure 17.9 Header Reception in LIN Slave Mode [Auto Baud Rate] (Sync Field Error)

17.7.2.2 Response Transmission

Figure 17.10 shows the operation of the LIN/UART interface (in LIN slave mode) in response transmission. **Table 17.80** provides processing in response transmission.

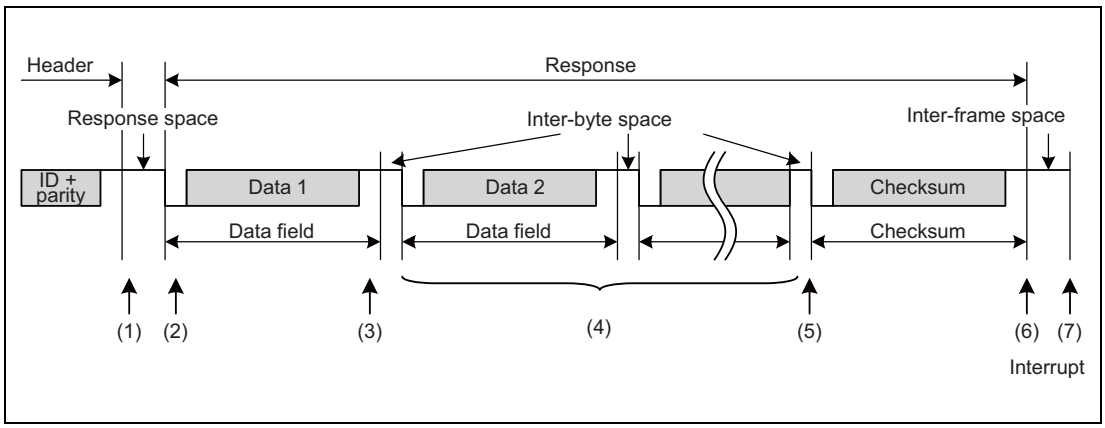


Figure 17.10 Operation in Response Transmission

Table 17.80 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 the transmission of inter-byte spaces 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) • 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 17.7.7, 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.

17.7.2.3 Response Reception

Figure 17.11 shows the operation of the LIN/UART interface (LIN slave mode) in response reception. **Table 17.81** provides processing in response reception.

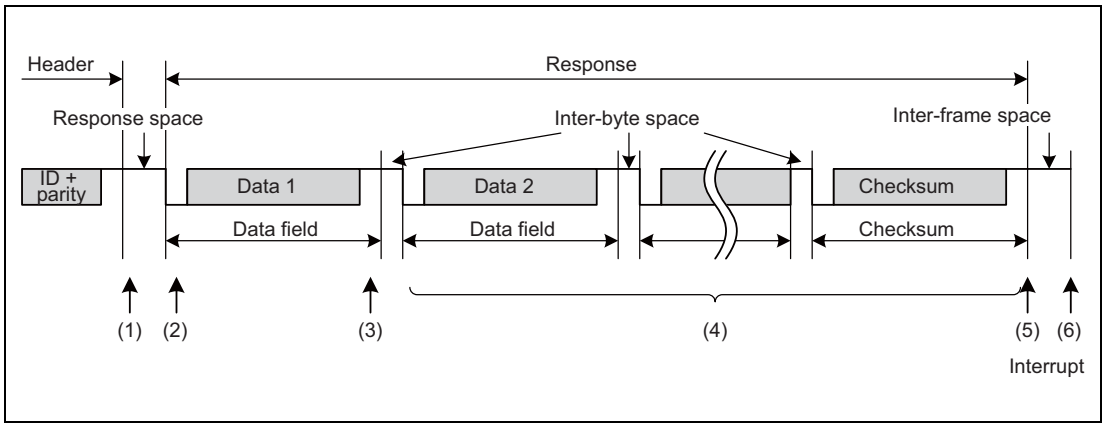


Figure 17.11 Operation in Response Reception

Table 17.81 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 the transmission of inter-byte spaces 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 frame/wake-up reception flag or an 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 17.7.7, 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.

17.7.2.4 No-response Request

Figure 17.12 shows the operation of the LIN/UART interface (LIN slave mode) when no response is requested. **Table 17.82** shows the processing that occurs when no response is requested.

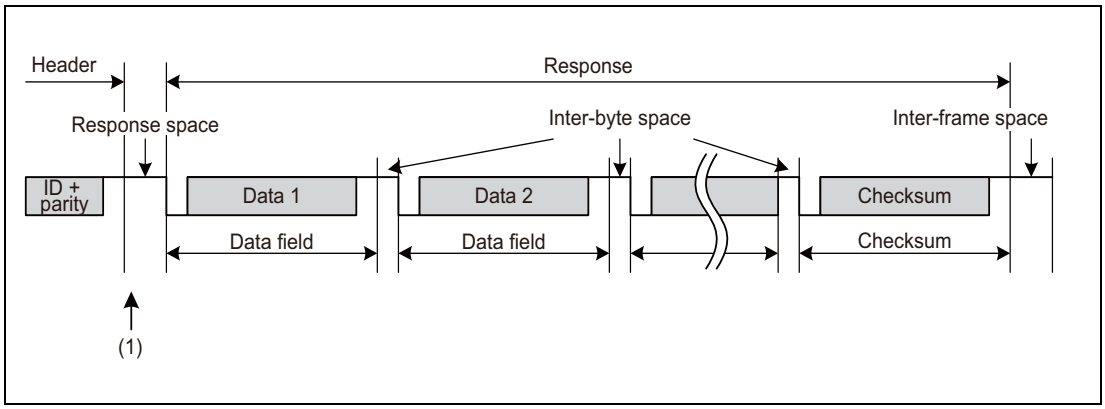


Figure 17.12 Operation when No Response is Requested

Table 17.82 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

17.7.3 Data Transmission/Reception

17.7.3.1 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 is compared bit by bit, and the results are stored in the BER flag in the RLN3nLEST register (see **Section 17.7.7, 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 17.13 shows an example of data transmission timing.

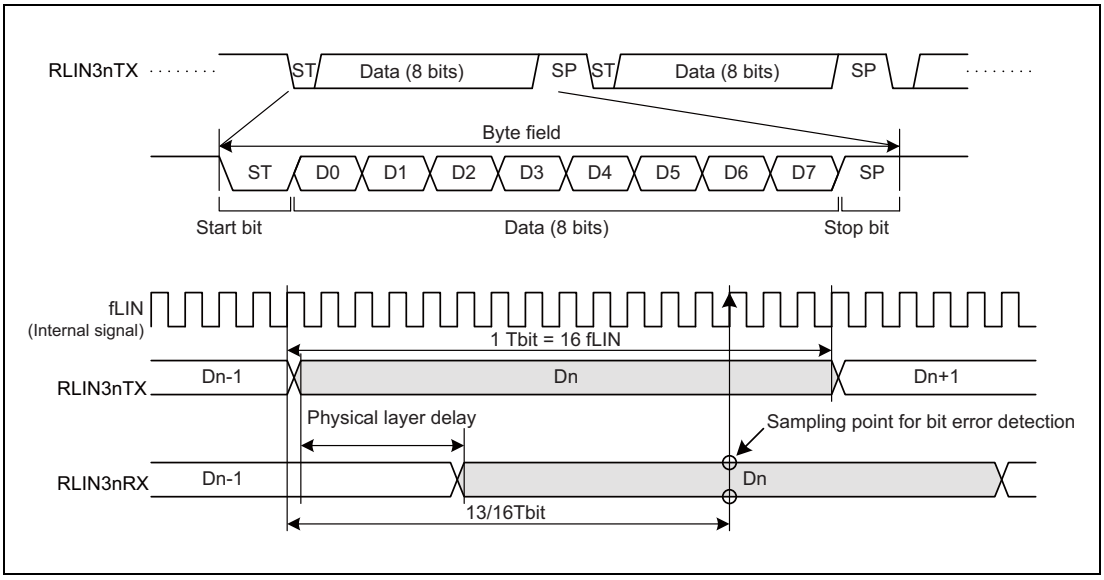


Figure 17.13 Example of Data Transmission Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])

17.7.3.2 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 clearing of the resetting is low-level-fixed 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 with respect to reception data. If the LRDNFS bit in the RLN3nLMD register is 0, the LIN/UART interface uses a noise filter, and for a sampling value the value determined by a 3-sampling majority rule on prescaler clocks is used. If the LRDNFS bit in the RLN3nLMD register is 1, the LIN/UART interface does not use a noise filter, and for a sampling value the value of the synchronized RLIN3nRX value at the sampling position is used as is.

Figure 17.14 shows an example of data reception timing.

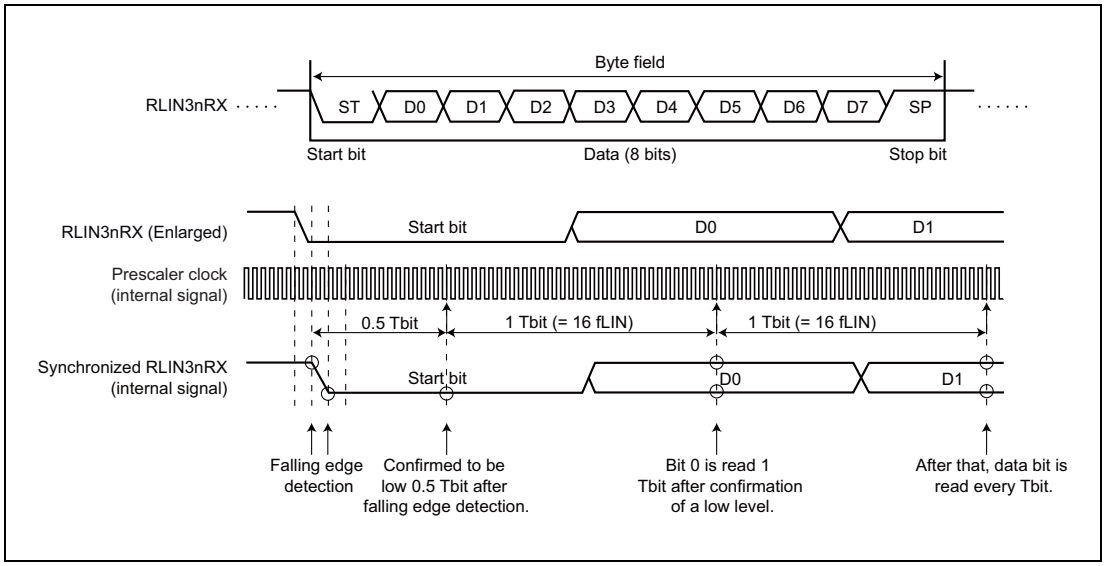


Figure 17.14 Example of Data Reception Timing (LIN Master Mode, LIN Slave Mode [Fixed Baud Rate])

17.7.4 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN/UART interface sends or receives data continuously.

17.7.4.1 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 17.15 depicts the LIN transmission processing and the required buffer.

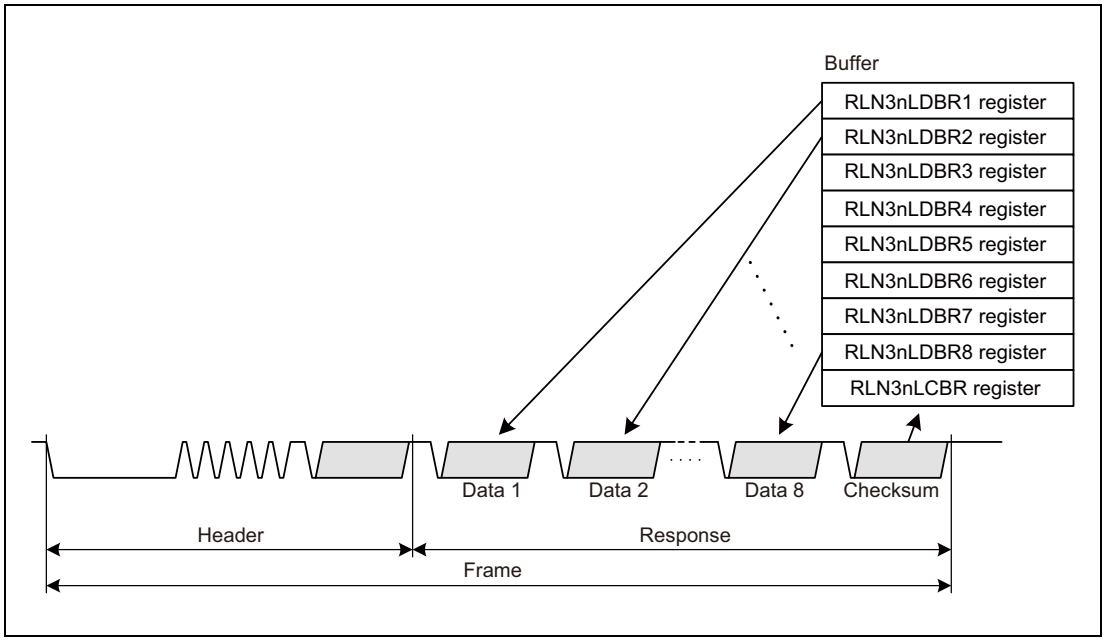


Figure 17.15 LIN Transmission Processing and Required Buffer

[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 turns 1 (successful header transmission).

Use frame separate mode when sending or receiving response data of 9 bytes or greater in LIN master mode.

17.7.4.2 Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame is stored in registers RLN3nLDBR1 to RLN3nLDBR8, 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 RLN3nLDBR1 to RLN3nLDBR4, respectively; however, no data is stored in registers RLN3nLDBR5 to RLN3nLDBR8. Also, the received checksum data is stored in the RLN3nLCBR register.

Figure 17.16 depicts the LIN reception processing and the required buffer.

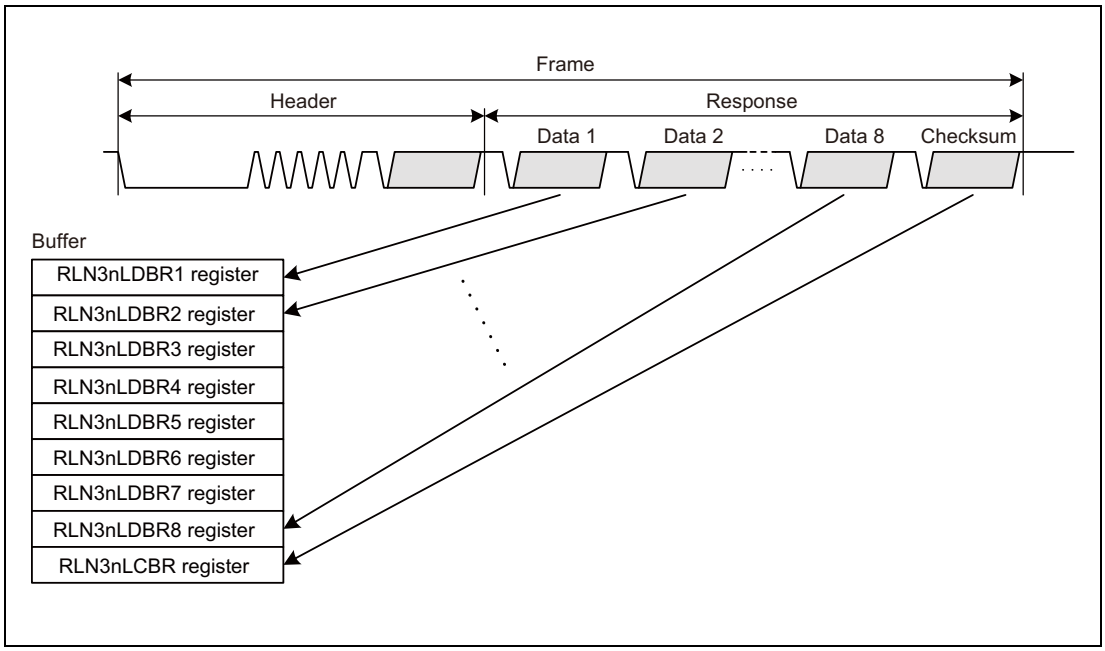


Figure 17.16 LIN Reception Processing and Required Buffer

[Reception of Data 1]

When the reception of the first byte of data is finished, the D1RC flag in the RLN3nLST register turns 1 (successful data 1 reception).

17.7.4.3 Multi-Byte Response Transmission/Reception Function

Normally in LIN communications, a response is 9 bytes or less including a checksum field; however, responses in 10 bytes or greater can also be sent 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 greater than 8 bytes, the LSS bit in RLN3nLDFC register should be set to 1 (indicating that the next data group to be sent or received is not the final data group) in the first data group (variable in 0 to 8 bytes) before sending or receiving the data group. After the transmission or reception, the user should determine whether the next data group is the final data group. If it is the final data group, the LSS bit should be set to 0 (indicating that the next data group to be sent or received is the final data group), and a checksum should be appended to the final 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 in the 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.

17.7.5 Wake-up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

17.7.5.1 Wake-up Transmission

In LIN wake-up mode, setting the RCDS bit in the RLN3nLDFC register to 1 (transmission) and 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 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 use), 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] bit 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 is output without any bit error, the FTC flag in the RLN3nLST register turns 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 17.17 shows the wake-up transmission timing.

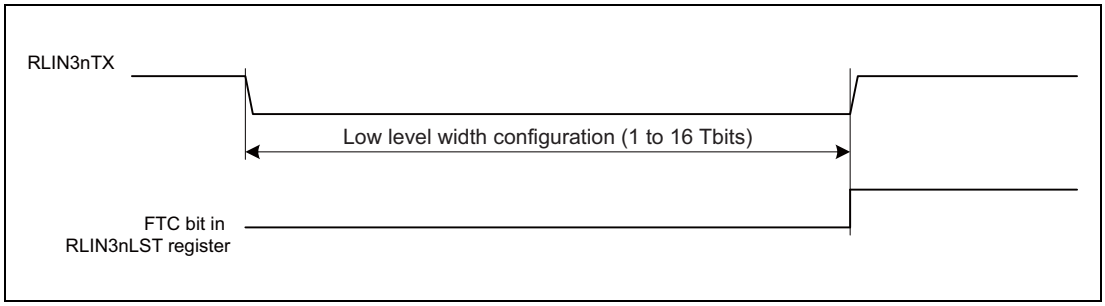


Figure 17.17 Wake-up Transmission Timing

17.7.5.2 Wake-up Reception

The detection of a wake-up 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 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 is executed without changing the baud rate generator setting at a transition 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 selects 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, the 130 µs or longer low-level width of the input signal to be measured regardless of the setting of the LCKS bit in the RLN3nLMD register.

When using this function, in LIN wake-up mode set the RFT bit in the RLN3nLDFC register to 0 (LIN master mode: response reception), the RCDS bit in the RLN3nLDFC register to 0 (LIN slave mode: response reception), or 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 turns 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.

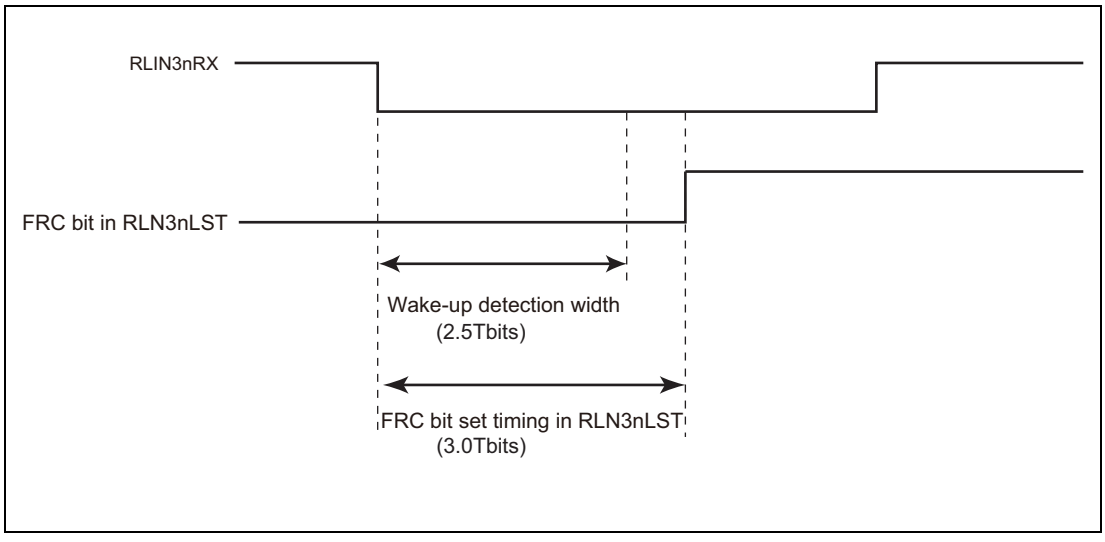


Figure 17.18 Input Signal Low level Count Function

17.7.5.3 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 in the LIN/UART interface.

17.7.6 Status

During 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, successful header transmission/reception, can generate interrupt requests.

Table 17.83 shows the types of statuses available in LIN master mode. **Table 17.84** lists the types of statuses available in LIN slave mode [auto baud rate] and in LIN slave mode [fixed baud rate].

Table 17.83 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 flags in the RLN3nLEST register turns 1 (error detected).	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software**Note 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.*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 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 flags 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 17.84 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	—
Frame/wake-up transmission end	When a response field, a wake-up signal, or a data group is transmitted successfully.	• 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 response field, a wake-up signal, or a data group is received successfully.	• 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, IPER flag, CSER flag, SFER flag, FER flag, TER flag, and BER flag in the RLN3nLEST register turns 1 (error detected).	• 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 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.*2	• When cleared by software • After 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 • After transition to LIN reset mode	LIN operation mode	HTRC flag in RLN3nLST register	√

Note 1. In 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).

17.7.7 Error Status

17.7.7.1 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 checked by means of the corresponding bits in the RLN3nLEST register.

All error statuses represent interrupt events.

Table 17.85 shows the types of error statuses.

Table 17.85 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 is detected to be high level when sending a break • LIN bus is detected to be low level when sending a break delimiter • LIN bus is detected to be 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 does not terminate 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 test gives 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: • The first reception data byte is received after completion of header transmission but before a response transmission/reception request is set • The first reception data byte is received after the completion of previous data group reception before a transmission/reception request for another data group is set.	LIN operation mode	Cancel	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, the process is canceled 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 canceled immediately after that area. 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 detected also 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 this can be calculated according to the following formula:

[Frame timeout]
On classic selection (when the CSM bit in RLN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
On enhanced selection (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 greater than the TFRAME_MAX of LIN Specification Package Revision 1.3 on classic selection, or the TFRAME_MAX of LIN Specification Package Revision 2.x on enhanced selection.
[Response timeout]
Timeout time = (number of data bytes + 1) × 14 [Tbit]

When en error is detected, time-out 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 17.19 shows the time domain in which the LIN/UART interface in LIN master mode performs monitoring for error detection.

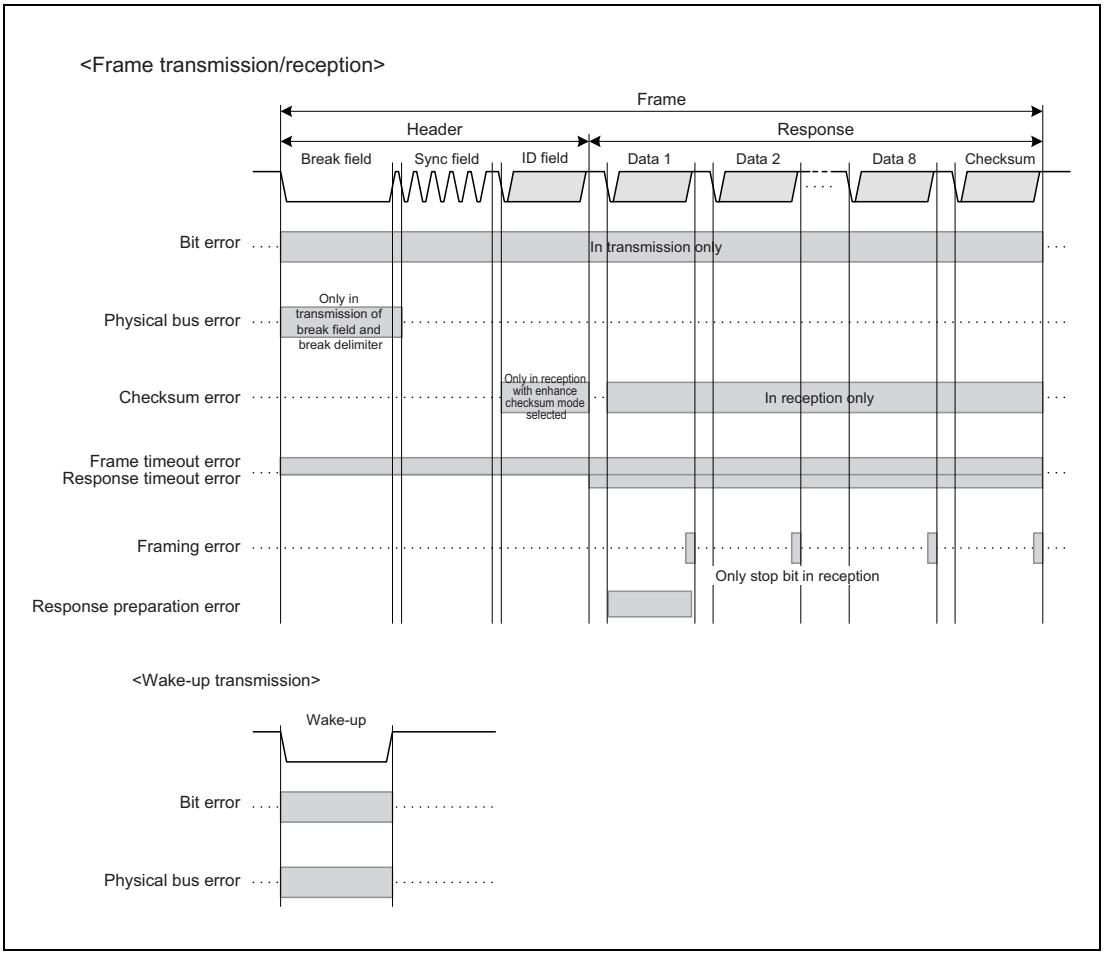


Figure 17.19 Target Time Area for LIN Error Detection (LIN Master Mode)

17.7.7.2 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 17.86 shows the types of error statuses.

Table 17.86 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 does not terminate within a given time*3	LIN operation mode	Cancel	Enabled	TER 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
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 test gives 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, before the first reception data byte is received, response preparation is not made in time. • Before the completion of receiving the first reception data byte for the next data group in a multi-byte response reception, response preparation for the next group is not made in time	LIN operation mode	Cancel	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, the process is canceled 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 canceled immediately after that area. 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 error can be 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 time-out 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] bit of the RLN3nLDFC register). When the LNRR bit is set, the timeout function stops.

[Frame timeout]

On classic selection (when the CSM bit in RLN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]

On enhanced selection (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 greater than the TFRAME_MAX of LIN Specification Package Revision 1.3 on classic selection, or the TFRAME_MAX of LIN Specification Package Revision 2.x on enhanced selection.

[Response timeout]

Timeout time = (number of data bytes + 1) × 14 [Tbit]

When en error is detected, time-out 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 receive complete 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 17.20 shows the time domain in which the LIN/UART interface in slave mode performs monitoring for error detection.

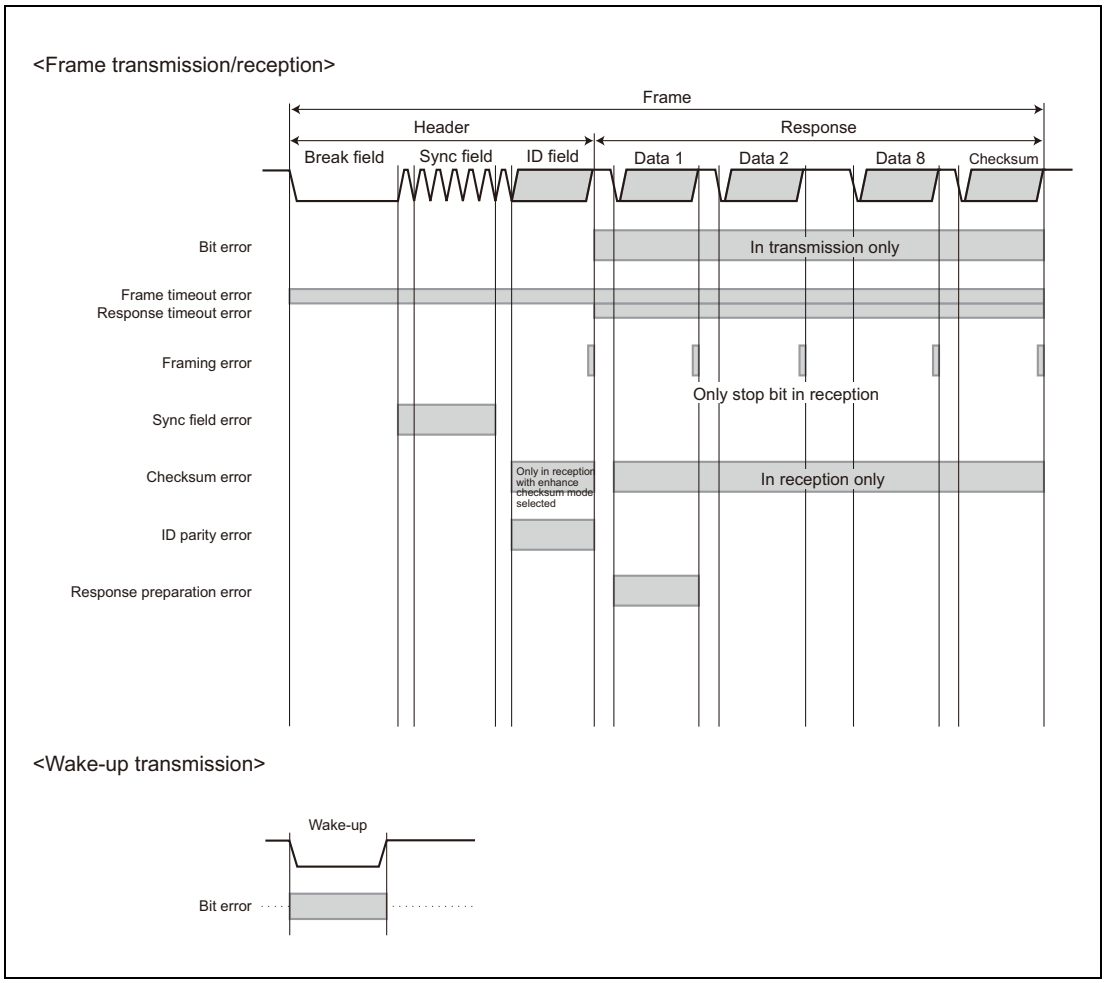


Figure 17.20 Target Time Area for LIN Error Detection (LIN Slave Mode)

17.8 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.

17.8.1 Transmission

Figure 17.21 shows LIN/UART interface (in UART mode) transmission operations; **Table 17.87** shows LIN/UART interface (in UART mode) transmission processing.

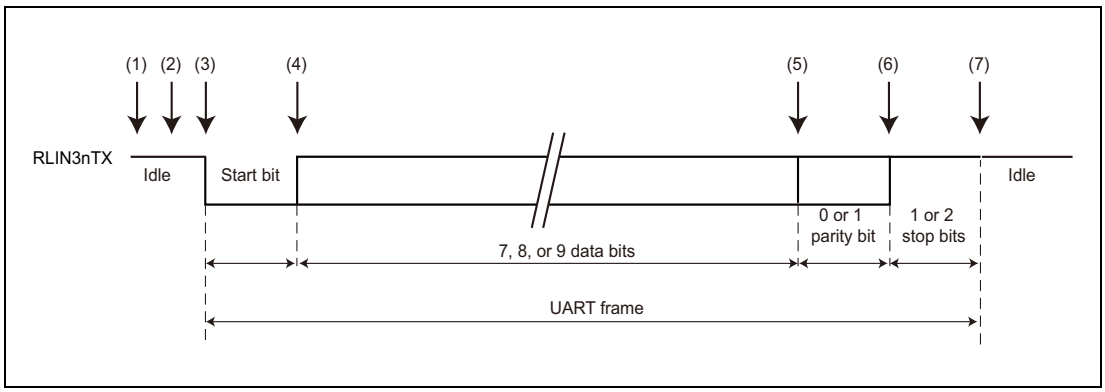


Figure 17.21 LIN/UART Interface (in UART Mode) Transmission Operation

Table 17.87 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. • Sets error detection enable. • 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. [When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • 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.	• 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 17.8.1.4, Transmission Start Wait Function**. [When the UTIGTS bit is 0 (a transmission interrupt is generated)] • Outputs a transmission interrupt.
(4)		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 17.87 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 piece 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 output upon end of transmission)] • When transmitting data continuously, goes to step (2).	[When the UTIGTS bit is 1 (a transmission interrupt is output upon end of transmission)] • Generates a RLIN3n transmission interrupt request. • Clears the transmission status flag.

17.8.1.1 Continuous Transmission

The LIN/UART interface (in UART mode) can transmit multiple sets of data continuously by using the RLN3nLUTDR register. **Figure 17.22** 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.

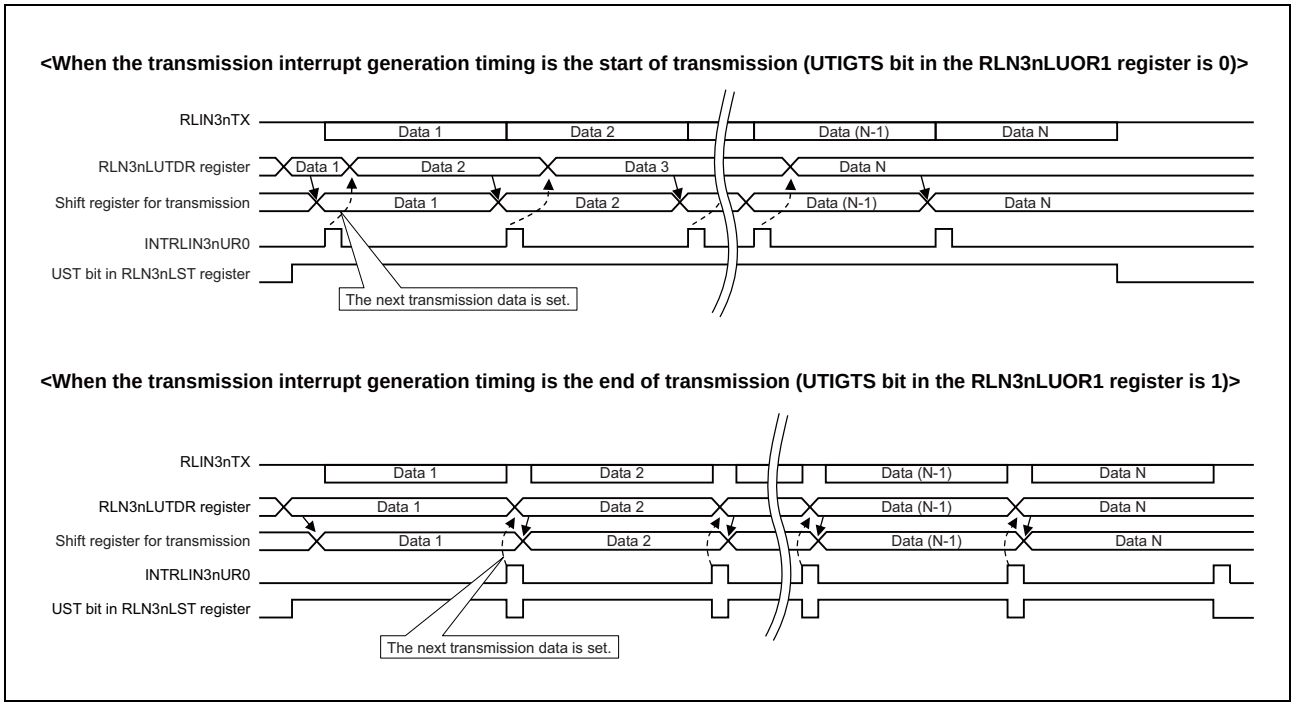


Figure 17.22 Operation Example of LIN/UART Interface (UART Mode) Continuous Transmission

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 only that the transmission interrupt generation timing is the start of transmission and the end of transmission of final data needs to be known.

17.8.1.2 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 17.23 shows the UART buffer transmission operation in the LIN/UART interface (in UART mode). **Table 17.88** shows the UART buffer transmission processing.

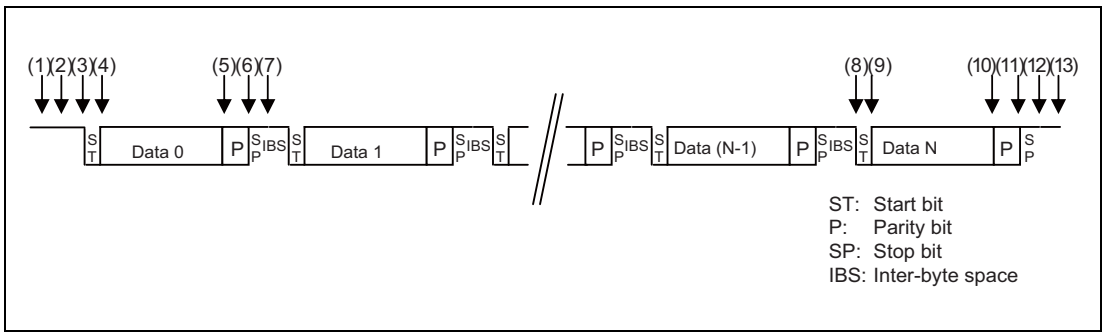


Figure 17.23 UART Buffer Transmission in LIN/UART Interface (in UART Mode)

Table 17.88 UART Buffer Transmission Processing in LIN/UART Interface (in UART Mode) (1/2)

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Sets error detection enable • 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 m 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 17.8.1.4, 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 setting bit, proceed to (12).)
(7)		Transmits an inter-byte space (idle).
		Repeats steps (3) to (7) until the number of data –1 that was set in the UART buffer data length select bits is reached.

Table 17.88 UART Buffer Transmission Processing in 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 buffer transmission end 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 transmission of the data that is set in the MDL [3:0] bits of the RLN3nLDFC register. The spaces between transmission data items can be set in the IBS bit in the RLN3nLSC register.

Figure 17.24 shows a 9-byte UART buffer and the transmission processing.

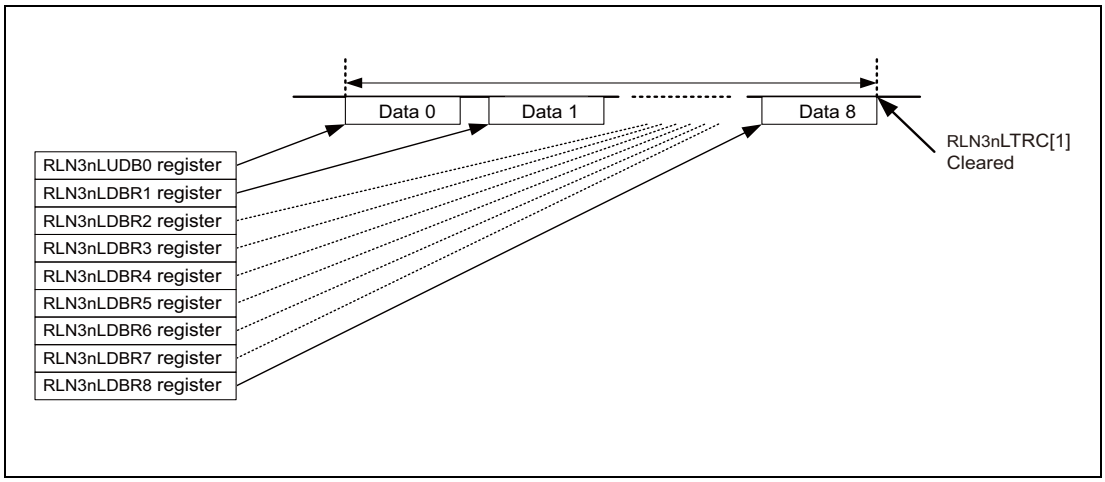


Figure 17.24 UART Buffer and Transmission Processing (for 9-Byte Transmission)

17.8.1.3 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 17.7.7, 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 17.89**.

Table 17.89 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 Sampling) is shown in **Figure 17.25**

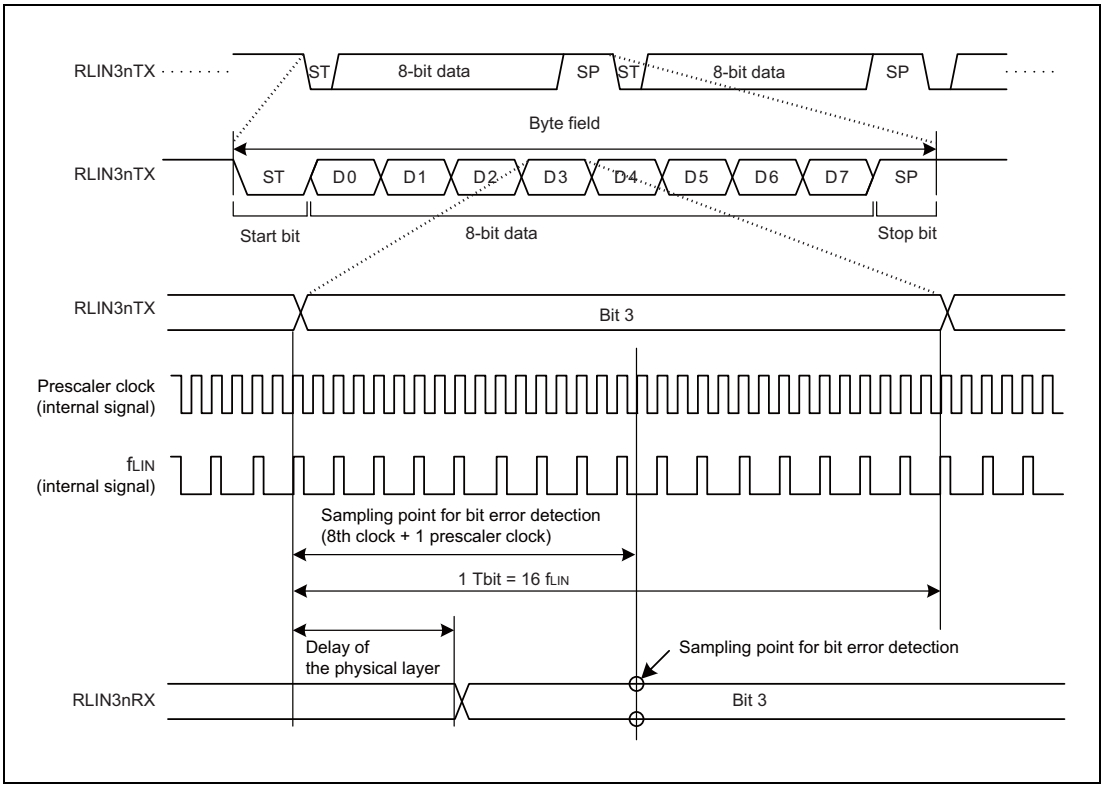


Figure 17.25 Example of Data Transmission Timing (When 1 Tbit = 16 Sampling)

17.8.1.4 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.

It should be noted that even if the UART stop bit length select bit (USBLS) in RLN3nBLFC register is 1 (stop bits = 2 bits), delay is made only for 1 bits.

Figure 17.26 shows the operation of transmission wait function.

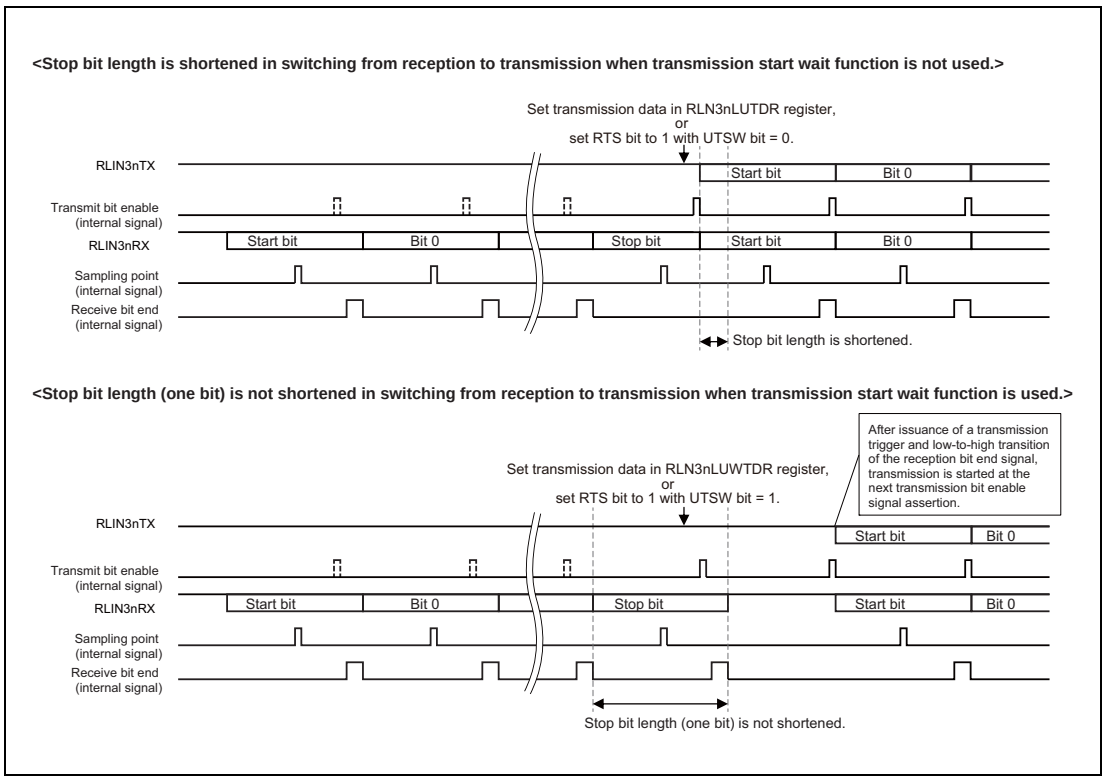


Figure 17.26 Transmission Wait Function (If Transmission Data is Set during the Stop Bits in the Received Data)

17.8.2 Reception

Figure 17.23 shows the LIN/UART interface (in UART mode) reception operation.**Table 17.90** shows the LIN/UART interface (in UART mode) reception processing.

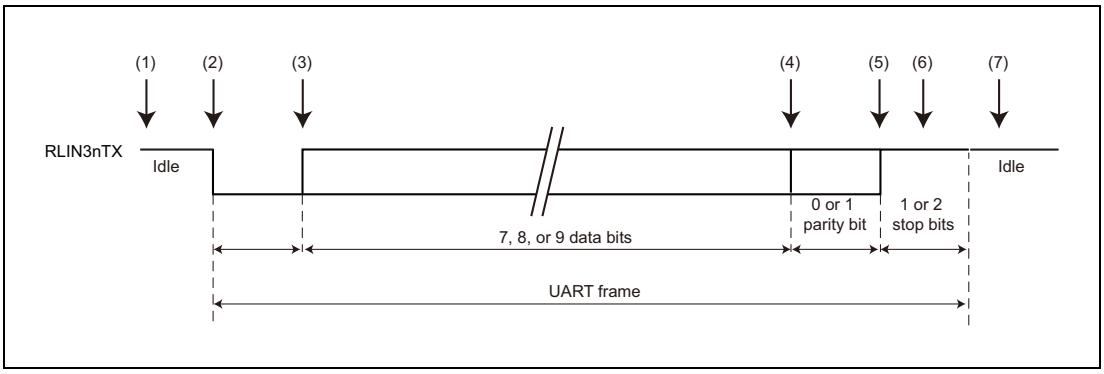


Figure 17.27 LIN/UART Interface (in UART Mode) Reception Operation

Table 17.90 LIN/UART Interface (in UART Mode) Reception Processing

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Sets error detection enable. • Sets data format. • Clears the LIN/UART interface from LIN reset mode. • Sets the receive enable bit (UROE bit) to 1.	• Waits for reception enable state switching 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.

17.8.2.1 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 the sampling per 1 Tbit is even and {(the number of the 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 is fixed at low level after the reset is cleared 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 filter function with respect to 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-sampling 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 17.28 shows an example of data reception timing.

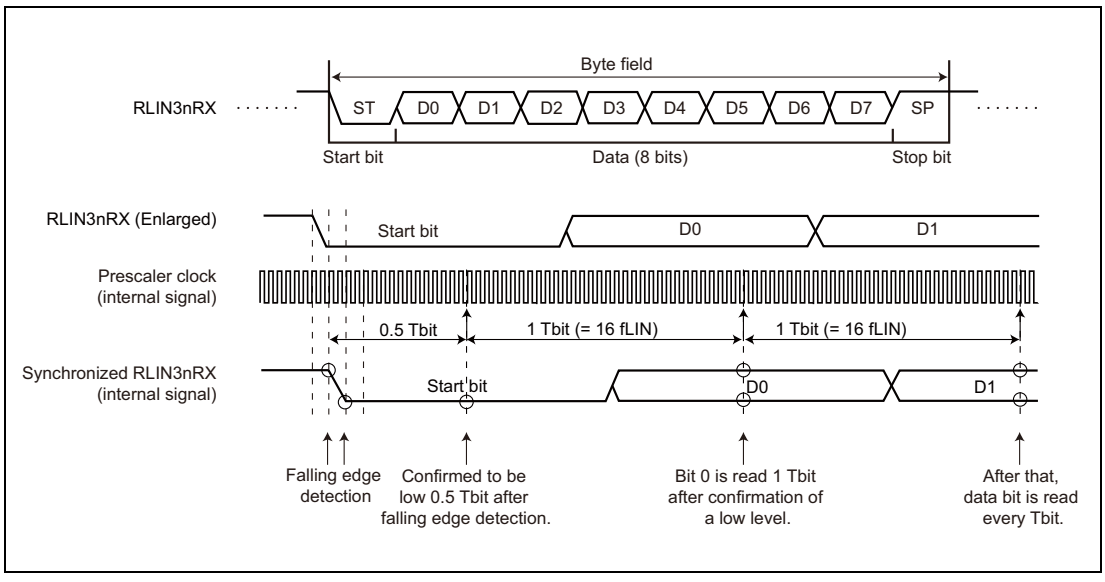


Figure 17.28 Example of Data Reception Timing (When Sampling Count is 16 in 1 Tbit)

17.8.3 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.

17.8.3.1 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).

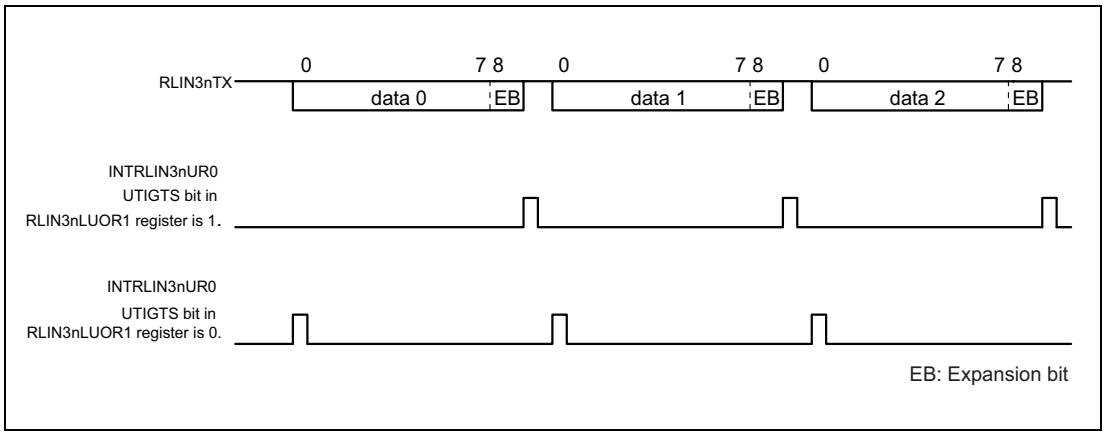


Figure 17.29 Transmission Example When Expansion Bit is Enabled (LSB First)

17.8.3.2 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 select bit (UEBDL) in the UART option register 1 (RLN3nLUOR1), a successful RLIN3n reception interrupt is generated when 9-bit data is received.

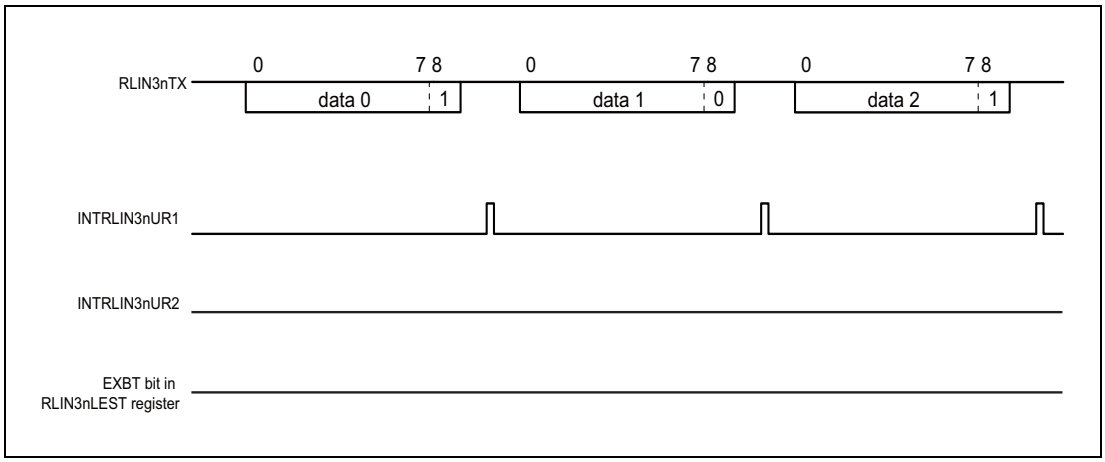


Figure 17.30 Expansion Bit Reception Example (LSB First)

17.8.3.3 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 and 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 select 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, a 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 17.31 shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.

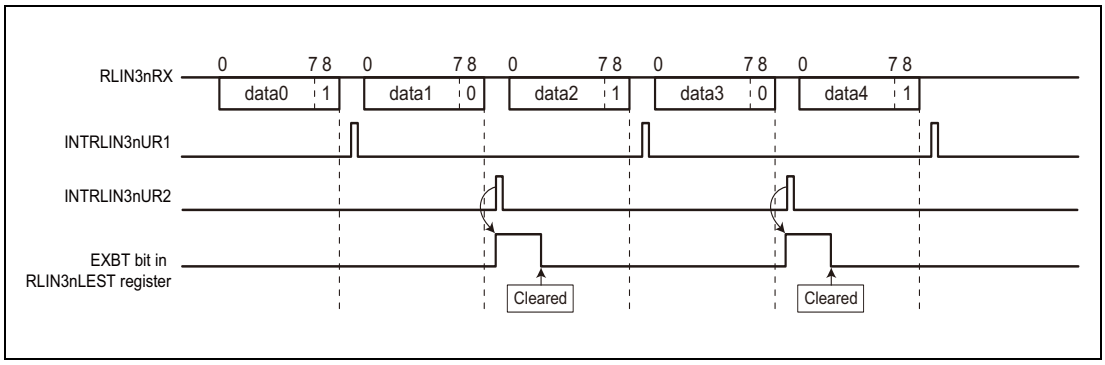


Figure 17.31 Expansion Bit Reception Example (with Expansion Bit Comparison) (LSB First, UEBDL = 0)

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.

17.8.3.4 Expansion Bit Reception (with Data Comparison)

If the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1 and 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 select bit (RLN3nUEBDL) is detected, the LIN/UART interface compares the 8 bits, exclusive of 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, a 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 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 17.32 shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.

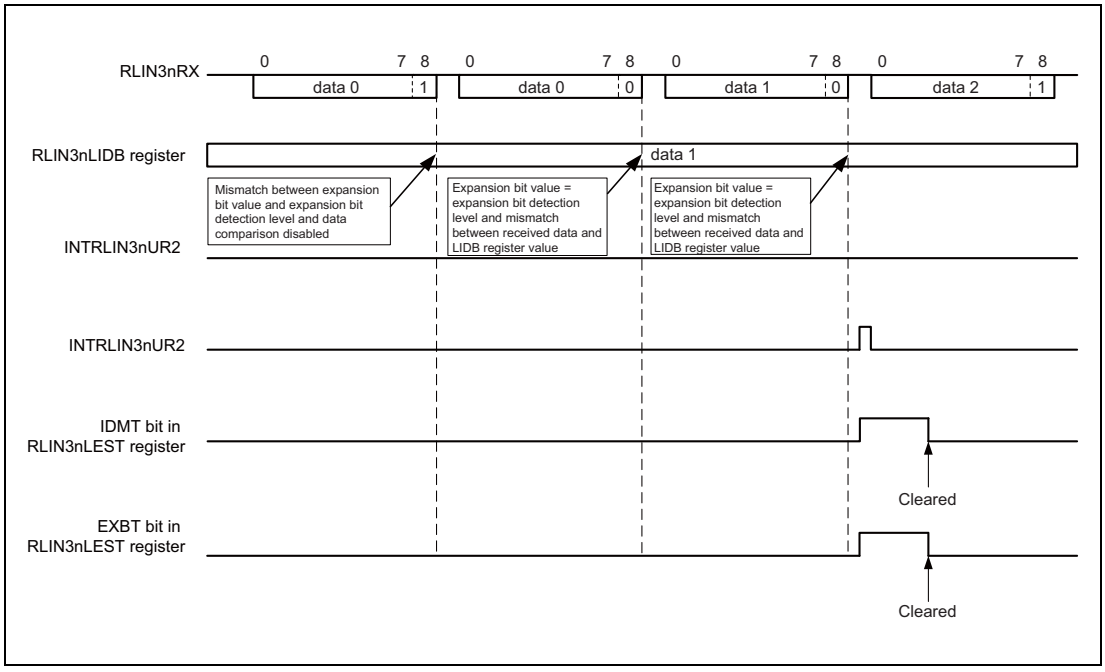


Figure 17.32 Expansion Bit Reception Example (with Data Comparison) (LSB First, UEBDL = 0)

17.8.4 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 17.91 shows the types of statuses available in UART mode.

Table 17.91 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 flags in the RLN3nLEST register turns 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 a 1 is written to the RTS bit in the RLN3nLTRC register.	• The transmission of the data set in the RLN3nLUTDR or RLN3nLUWTDR register is complete, but another transmission data item is not set • The transmission of the data in the UART buffer is complete, 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 turns the ERR flag in the RLN3nLST register to 0.

17.8.5 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 checked by means of the corresponding bits in the RLN3nLEST register.

Table 17.92 lists applicable status types.

Table 17.92 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.

17.9 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 in the LIN/UART interface. 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 types.

- 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, the operate is 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]. (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.

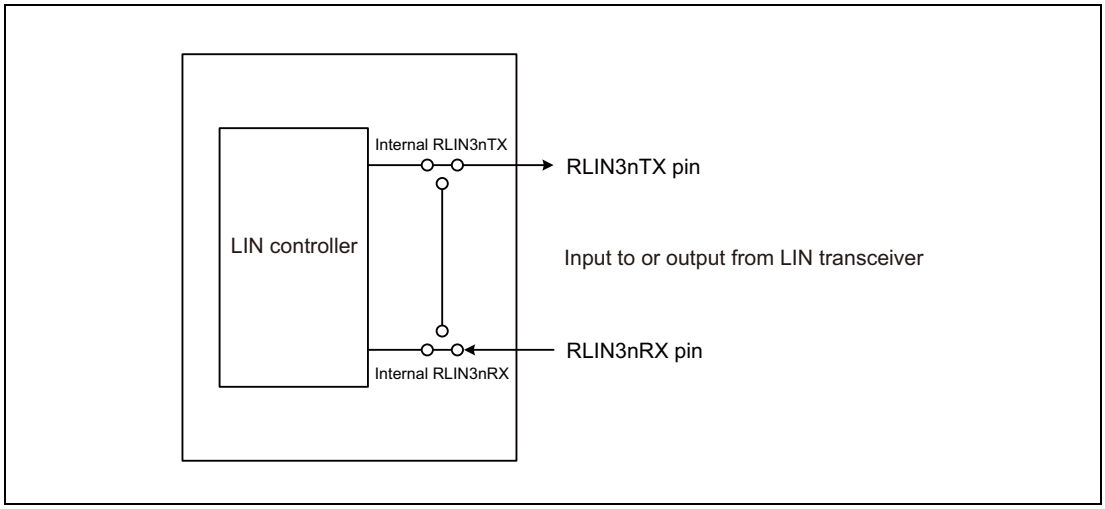


Figure 17.33 Connection in LIN Reset Mode, LIN Mode, and UART Mode

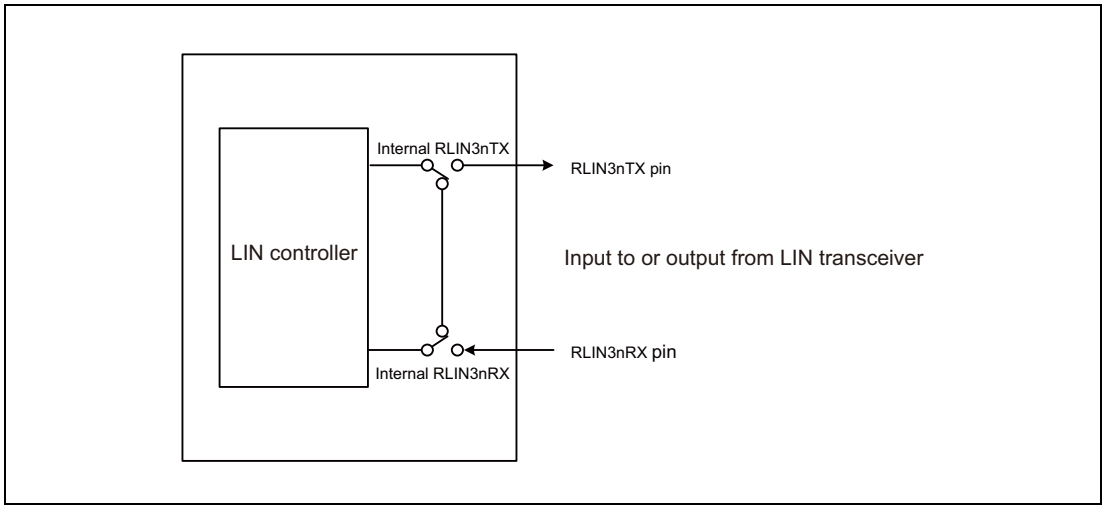


Figure 17.34 Connection in LIN Self-Test Mode

17.9.1 Change to LIN Self-Test Mode

Writing to the RLN3nLSTC register makes a transition to the LIN self-test mode.
When the LSTM bit in the RLN3nLSTC register is set to 1, the shift to the LIN self-test mode is checked.

When changing 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 follows:

- Change 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 ($A7_H$)
- 2nd write: RLN3nLSTC register = 0101 1000 (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.

17.9.2 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 $0xxx_B$
- Cancel 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.
- 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 reversed before being stored 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 applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Don't care

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 6, Exception/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 set of the successful header transmission flag to the set of the successful frame/wake-up transmission flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]
1 Tbit = frequency of LIN communication clock source × 16

17.9.3 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
- Cancel 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, set the calculation value to the RLN3nLCBR register.
- 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.
- 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 reversed before being stored 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 applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Don't care

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 6, Exception/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 set of the successful header transmission flag to the set 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

17.9.4 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 $x00x_B$
- Set the break field and space related registers.
  RLN3nLBFC register = 0000 000x$_B$*3
  RLN3nLSC register = 00xx 0001_B
- Cancel 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).
  (Without any setting of the RTS bit in the RLN3nLTRC register, the header reception and the response transmission are executed in this order.)
  The LIN slave self 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.
- 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 reversed before being stored 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 applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Don't care

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 6, Exception/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 set of the successful header reception flag to the set of the successful response/wake-up reception flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = frequency of 1/LIN communication clock source × 16

17.9.5 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
- Cancel 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, set the calculation value to the RLN3nLCBR register.
- 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.
- 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 reversed before being stored 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 applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Don't care

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 6, Exception/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 set of the successful header reception flag to the set of the successful response/wake-up reception flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = frequency of 1/LIN communication clock source × 16

17.9.6 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 have turned 11_B, change to LIN reset mode.
- Verify the cancelation of LIN self-test mode.
  Read the LSTM bit in the RLN3nLSTC register; confirm that it is not 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).

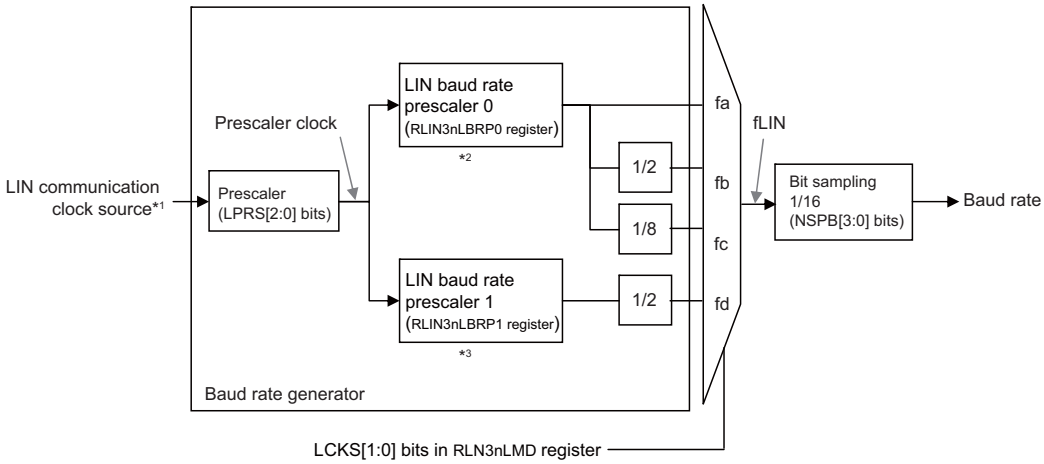
17.10 Baud Rate Generator

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 samples is the baud rate. The reciprocal of this baud rate is called the bit time (Tbit).

The LIN/UART interface has two kinds of baud rate generators. The baud rate generators switch over according to the mode used.

17.10.1 LIN Master Mode

Figure 17.35 shows a block diagram of baud rate generation in LIN master mode.



Note 1. For the LIN communication clock source, see **10.5.3, 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 N + 1.

Figure 17.35 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 bit rates are fa = 19200 × 16, fb = 9600 × 16, and fc = 2400 × 16. These bit rates are frequency-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 bit rate is fd = 10417 × 16. This bit rate is frequency-divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

Baud rate of LIN slave

= {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 or 16 [bps] (When fb is selected for f_{LIN})

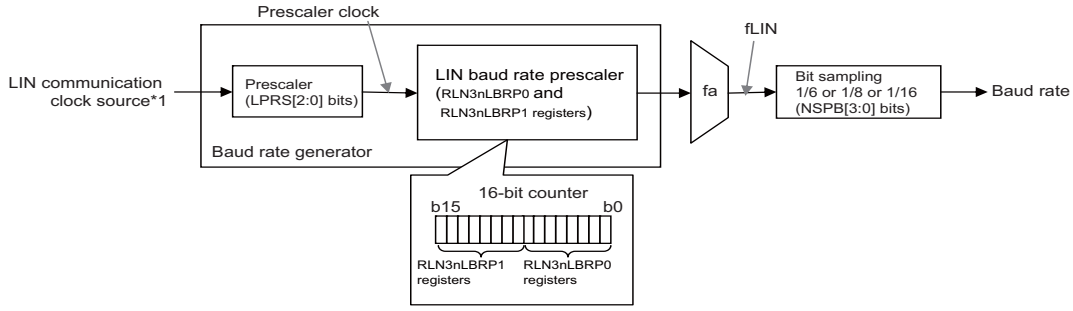
= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)
÷ (RLN3nLBRP0 + 1) ÷ 8 ÷ 16 [bps] (When fc is selected for f_{LIN})

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)

÷ (RLN3nLBRP1 + 1) ÷ 2 ÷ 16 [bps] (When fd is selected for f_{LIN})

17.10.2 LIN Slave Mode

Figure 17.36 shows a block diagram of baud rate generation in LIN slave mode.



Note 1. For the LIN communication clock source, see **10.5.3, Clock Domain Settings**.

Figure 17.36 Block Diagram of Baud Rate Generation in LIN Slave Mode

In LIN slave mode (auto baud rate), the baud rate can be set to the range from 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)

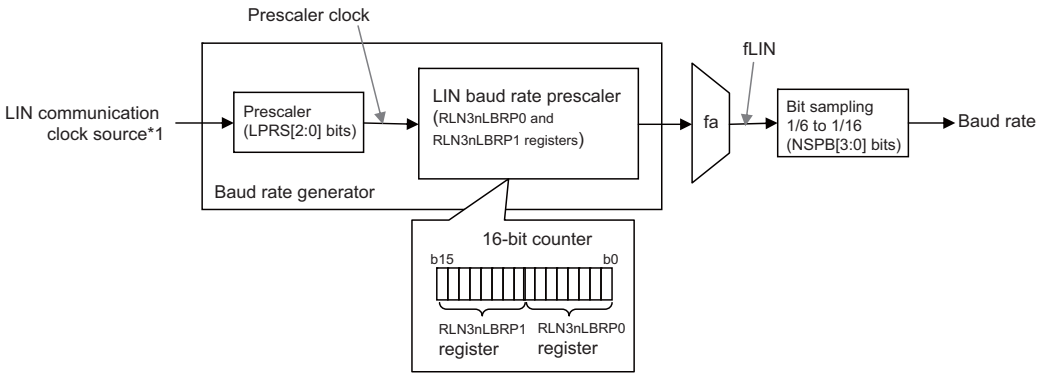
÷ (RLN3nLBRP01 + 1) ÷ 4 or 8 [bps] ([Auto baud rate])

NOTE

For a LIN slave with fixed baud rate, set the NSPB[3:0] bit 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).

17.10.3 UART Mode

Figure 17.37 shows a block diagram of baud rate generation in UART mode.



Note 1. For the LIN communication clock source, see **10.5.3, Clock Domain Settings**.

Figure 17.37 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] select clock) ÷ (RLN3nLBRP01 + 1) ÷ {RLN3nLWBR.NSPB[3:0] select count} [bps]

The formula for baud rate is described below.

Baud rate for UART

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)
÷ (RLN3nLBRP01 + 1) ÷ the number of selected NSPB[3:0] [bps]

17.11 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 (to use the noise filter), the noise filter is activated. The noise filter samples the level of the synchronized RLIN3nRX with the prescaler clock, and outputs the sampling value determined by a 3-sampling majority rule. The value of each bit of the receive data is determined based on the noise filter output.

Figure 17.38 shows the configuration of the noise filter, **Figure 17.39** shows an example of a noise filter circuit, and **Figure 17.40** shows the determination of the received data when the noise filter is used.

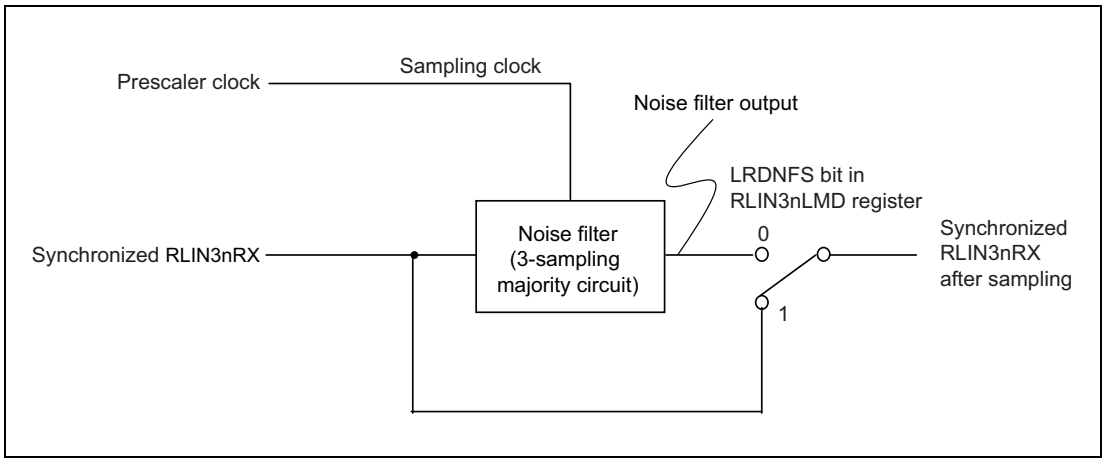


Figure 17.38 Configuration of Noise Filter

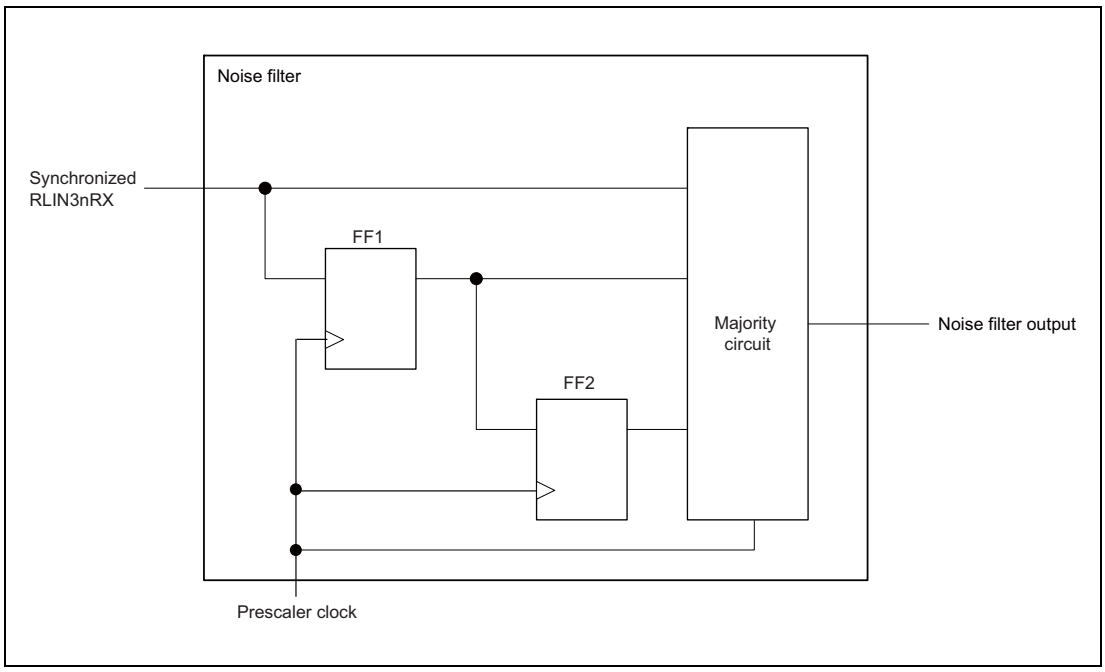


Figure 17.39 Example of Noise Filter Circuit

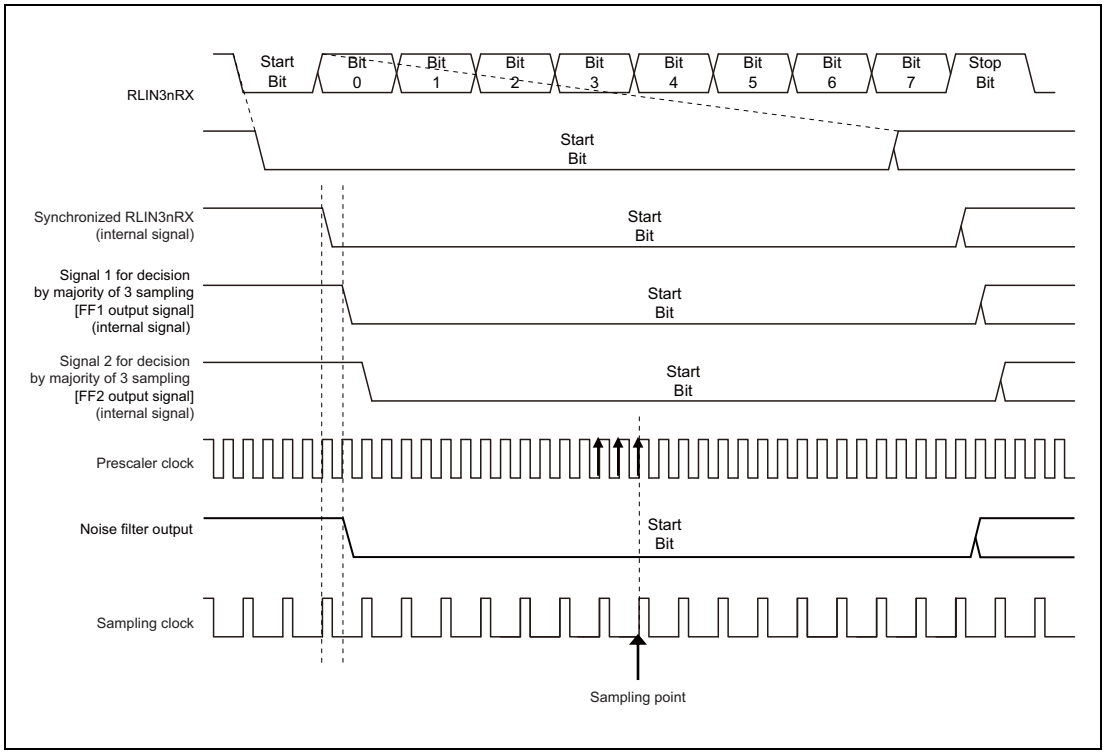


Figure 17.40 Determination of Received Data when Noise Filter is Used

Section 18 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/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RIIC.

18.1 Features of RH850/F1L RIIC

18.1.1 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 18.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1					
Name	RIICn (n = 0)					

Table 18.2 Index

Index	Meaning
n	Throughout this section, the individual RIIC units are identified by the index “n” (n = 0): for example, RIICnCR1 is the I2C bus control register1.

18.1.2 Register Base Address

RIIC base addresses are listed in the following table.

RIIC register addresses are given as offsets from the base addresses in general.

Table 18.3 Register Base Address

Base Address Name	Base Address
<RIIC0_base>	FFCA 0000H

18.1.3 Clock Supply

The RIIC clock supply is shown in the following table.

Table 18.4 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
RIICn	PCLK	CPUCLK2

18.1.4 Interrupt Requests

RIIC interrupt requests are listed in the following table.

Table 18.5 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
RIIC0			
INTIICnEE	RIIC communication error / event generation interrupt	71	—
INTIICnRI	RIIC Receive end interrupt	70	20 (channels 0 to 7)
INTIICnTI	RIIC Transmit data empty interrupt	68	19 (channels 0 to 7)
INTIICnTEI	RIIC Transmit end interrupt	69	—

18.1.5 Reset Sources

RIIC reset sources are listed in the following table. RIIC is initialized by these reset sources.

Table 18.6 Reset Sources

Unit Name	Reset Source
RIIC0	All reset sources (ISORES)

18.1.6 External Input/Output Signals

External input/output signals of RIIC are listed below.

Table 18.7 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
RIIC0		
RIICnSCL	Serial clock I/O pin	RIIC0SCL
RIICnSDA	Serial data I/O pin	RIIC0SDA

18.2 Overview

18.2.1 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 made.
- 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 on detection of a not-acknowledge bit.
- For reception, the acknowledge bit is automatically transmitted
  - If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible.

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 conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line.
  - In master operation, loss of arbitration is detected by testing for non-matching of internal and line levels for transmit data.
- Loss of arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the issuing of double start conditions).
- Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching is detectable.
- Loss of arbitration due to non-matching of internal and line levels for data is detectable in slave transmission.

Timeout function

The internal time-out function is capable of detecting long-interval stop of the SCL (clock signal).

Noise removal

The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.

Interrupt sources

- Four sources:
  - Error in transfer or occurrence of events (detection of arbitration lost, NACK, time-out, 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

18.2.2 Block Diagram

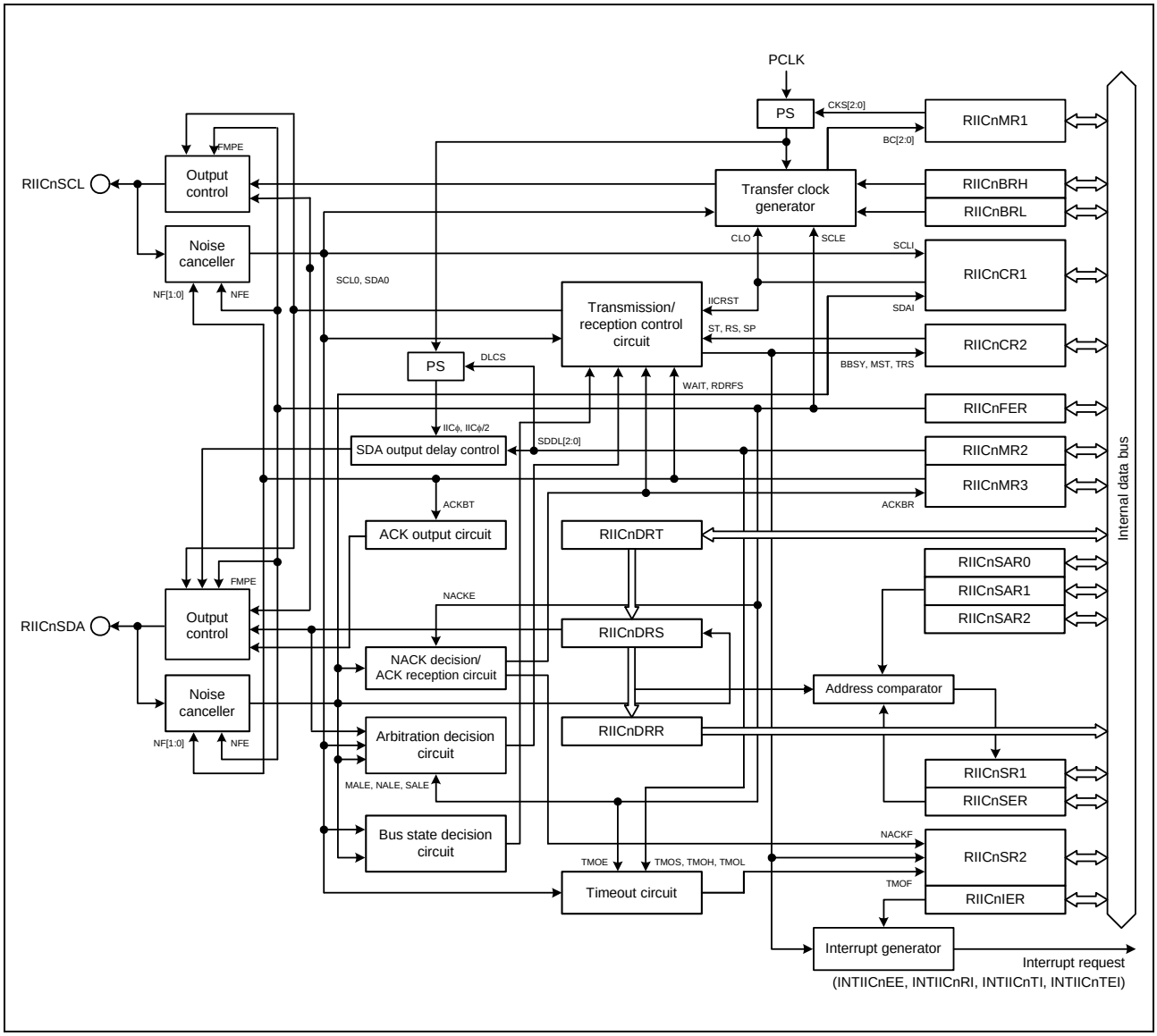


Figure 18.1 Block Diagram of RIIC

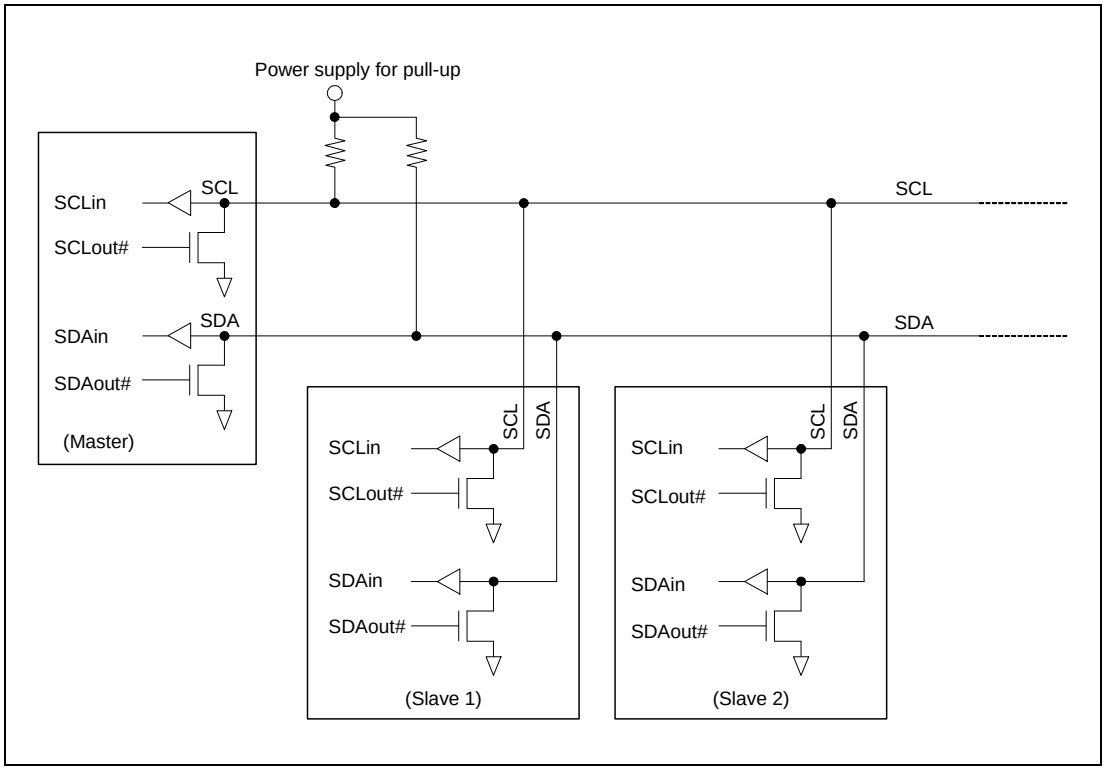


Figure 18.2 Connections to the External Circuit by the I/O Pins (I2C Bus Configuration Example)

18.3 Registers

18.3.1 List of Registers

RIIC registers are listed in the table below.

For details about <RIICn_base>, see **Section 18.1.2, Register Base Address**.

Table 18.8 List of Registers

Module 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	—

18.3.2 RIICnCR1 — I2C Bus Control Register 1

Access: RIICnCR1 is a 32-bit readable/writable register.
RIICnCR1L and RIICnCR1H are 16-bit readable/writable registers.
RIICnCR1LL, RIICnCR1LH, RIICnCR1HL, and RIICnCR1HH are 8-bit readable/writable registers.

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	R/W	R/W	R/W	R	R

Table 18.9 RIICnCR1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be 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 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. (The CLO bit is cleared automatically after one clock cycle is output.)
4	SOWP	SCLO/SDAO Write Protect 0: Bits SCLO and SDAO can be written. 1: Bits SCLO and SDAO are protected. (This bit is read as 1.)
3	SCLO	SCL Output Control/Monitor • Read: 0: The RIIC has driven the RIICnSCL pin low. 1: The RIIC has released the RIICnSCL pin. • Write: 0: The RIIC drives the RIICnSCL pin low. 1: The RIIC releases the RIICnSCL pin.

Table 18.9 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 RIIC has released the RIICnSDA pin. • Write: 0: The RIIC drives the RIICnSDA pin low. 1: The RIIC 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: RIICnSCL line is low. 1: RIICnSCL 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, repeated START condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

When reading these bits, the state of signals output from the RIIC can be read.

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 18.13.2, Extra SCL Clock Cycle Output Function**.

IICRST Bit (I2C Bus Interface 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 18.10** 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 RIICnICSR2) as well as the internal states of the RIIC. For the reset conditions for each register, see **Section 18.14, 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.

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 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 18.10 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 18.10, 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.

18.3.3 RIICnCR2 — I2C Bus Control Register 2

Access: RIICnCR2 is a 32-bit readable/writable register.
RIICnCR2L and RIICnCR2H are 16-bit readable/writable registers.
RIICnCR2LL, RIICnCR2LH, RIICnCR2HL, and RIICnCR2HH are 8-bit readable/writable registers.

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 18.11 RIICnCR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7	BBSY	Bus Busy Detection Flag 0: The I2C bus is released (bus free state). 1: The I2C bus is occupied (bus busy state or in the bus free state).
6	MST*1	Master/Slave Mode 0: Slave mode 1: Transmit mode
5	TRS*1	Transmit/Receive Mode 0: Receive mode 1: Transmit mode
4	Reserved	This bit is read as the value after reset. The write value should be 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	This bit is read as the value after reset. The write value should be 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 18.12, 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 18.12, 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 transmit 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 18.12, 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 R/W# bit added to the slave address is set to 0 in master mode
- 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 mode, a match between the received slave address and the address enabled in RIICnSER

when the value of the received R/W# bit is 0 (including cases where the received address is the general call address)

- In slave mode, a restart condition is detected (a start 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)

CAUTION

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.

18.3.4 RIICnMR1 — I2C Bus Mode Register 1

Access: RIICnMR1 is a 32-bit readable/writable register.
RIICnMR1L and RIICnMR1H are 16-bit readable/writable registers.
RIICnMR1LL, RIICnMR1LH, RIICnMR1HL, and RIICnMR1HH are 8-bit readable/writable registers.

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	W	R/W	R/W	R/W	W	R/W	R/W	R/W

Table 18.12 RIICnMR1 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be 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.)
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, set the value while the SCL line is at a low level.

The values of the BC[2:0] bits automatically return to 000_B at the end of a data transfer including the acknowledge bit or when a start condition including a restart condition is detected.

[Clearing conditions]

When 1 is written to the RIICnCR1.IICRST bit and a RIIC reset or internal reset is initiated.

18.3.5 RIICnMR2 — I2C Bus Mode Register 2

Access: RIICnMR2 is a 32-bit readable/writable register.
RIICnMR2L and RIICnMR2H are 16-bit readable/writable registers.
RIICnMR2LL, RIICnMR2LH, RIICnMR2HL, and RIICnMR2HH are 8-bit readable/writable registers.

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 18.13 RIICnMR2 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be 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	This bit is read as the value after reset. The write value should be the value after reset.
2	TMOH	Timeout H Count Control 0: Count is disabled while the SCL line is at a high level. 1: Count is enabled while the SCL line is at a high level.

Table 18.13 RIICnMR2 Register Contents (2/2)

Bit Position	Bit Name	Function
1	TMOL	Timeout L Count Control 0: Count is disabled while the SCL line is at a low level. 1: Count is enabled 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 18.13.1, 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 18.7, 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])

18.3.6 RIICnMR3 — I2C Bus Mode Register 3

Access: RIICnMR3 is a 32-bit readable/writable register.
RIICnMR3L and RIICnMR3H are 16-bit readable/writable registers.
RIICnMR3LL, RIICnMR3LH, RIICnMR3HL, and RIICnMR3HH are 8-bit readable/writable registers.

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 18.14 RIICnMR3 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	These bits are read as the value after reset. The write value should be 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 (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 + analog noise filter: 120 ns (reference values)] 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 stop condition issuance is detected (when a stop condition is detected with the RIICnCR2.SP bit set to 1)
- 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.

18.3.7 RIICnFER — I2C Bus Function Enable Register

Access: RIICnFER is a 32-bit readable/writable register.
RIICnFERL and RIICnFERH are 16-bit readable/writable registers.
RIICnFERLL, RIICnFERLH, RIICnFERHL, and RIICnFERHH are 8-bit readable/writable registers.

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 18.15 RIICnFER Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	These bits are read as the value after reset. The write value should be 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.)
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 18.13.1, Timeout Function**.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

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 from the slave device 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 the SCL input clock. 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 I2C 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.

18.3.8 RIICnSER — I2C Bus Status Enable Register

Access: RIICnSER is a 32-bit readable/writable register.
RIICnSERL and RIICnSERH are 16-bit readable/writable registers.
RIICnSERLL, RIICnSERLH, RRIICnSERHL, and RIICnSERHH are 8-bit readable/writable registers.

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 18.16 RIICnSER Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	These bits are read as the value after reset. The write value should be 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	This bit is read as the value after reset. The write value should be 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 18.9.3, Device-ID Address Detection**.

18.3.9 RIICnIER — I2C Bus Interrupt Enable Register

Access: RIICnIER is a 32-bit readable/writable register.
RIICnIERL and RIICnIERH are 16-bit readable/writable registers.
RIICnIERLL, RIICnIERLH, RIICnIERHL, and RIICnIERHH are 8-bit readable/writable registers.

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 18.17 RIICnIER Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be 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.

An INTIICnRI interrupt can be released by setting the RDRF flag to 0 or by setting the RIE bit to 0.

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.

18.3.10 RIICnSR1 — I2C Bus Status Register 1

Access: RIICnSR1 is a 32-bit readable/writable register.
RIICnSR1L and RIICnSR1H are 16-bit readable/writable registers.
RIICnSR1LL, RIICnSR1LH, RIICnSR1HL, and RIICnSR1HH are 8/1-bit readable/writable registers.

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
	—	—	—	—	—	—	—	—	—	—	RIICn DID	—	RIICn GCA	RIICn AAS2	RIICn AAS1	RIICn 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 18.18 RIICnSR1 Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
5	RIICnDID	Device-ID Address Detection Flag 0: Device-ID address is not detected. 1: Device-ID address is detected.
4	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
3	RIICnGCA	General Call Address Detection Flag 0: General call address is not detected. 1: General call address is detected.
2	RIICnAAS2	Slave Address 2 Detection Flag 0: Slave address 2 is not detected. 1: Slave address 2 is detected.
1	RIICnAAS1	Slave Address 1 Detection Flag 0: Slave address 1 is not detected. 1: Slave address 1 is detected.
0	RIICnAAS0	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.
- 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.
  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. 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

18.3.11 RIICnSR2 — I2C Bus Status Register 2

Access: RIICnSR2 is a 32-bit readable/writable register.
RIICnSR2L and RIICnSR2H are 16-bit readable/writable registers.
RIICnSR2LL, RIICnSR2LH, RIICnSR2HL, and RIICnSR2HH are 8/1-bit readable/writable registers.

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
	—	—	—	—	—	—	—	—	RIICnTDRE	RIICnTEND	RIICnRDRF	RIICnNACKF	RIICnSTOP	RIICnSTART	RIICnAL	RIICnTMOF
Value after reset	0	0	0	0	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 18.19 RIICnSR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7	RIICnTDRE	Transmit Data Empty Flag 0: RIICnDRT contains transmit data. 1: RIICnDRT contains no transmit data.
6	RIICnTEND	Transmit End Flag 0: Data is being transmitted. 1: Data has been transmitted.
5	RIICnRDRF	Receive Complete Flag 0: RIICnDRR contains no receive data. 1: RIICnDRR contains receive data.
4	RIICnNACKF	NACK Detection Flag 0: NACK is not detected. 1: NACK is detected.
3	RIICnSTOP	Stop Condition Detection Flag 0: Stop condition is not detected. 1: Stop condition is detected.
2	RIICnSTART	Start Condition Detection Flag 0: Start condition is not detected. 1: Start condition is detected.
1	RIICnAL	Arbitration-Lost Flag 0: Arbitration is not lost. 1: Arbitration is lost.
0	RIICnTMOF	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]

When the SCL line state remains unchanged for the period specified by bits RIICnMR2.TMOH, TMOL, and TMOS with the RIICnFER.TMOE bit set to 1 (timeout detection enabled) in master mode or in the slave specification state.

[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 18.20 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
  This flag is set to 1 at the rising edge of the eighth or ninth SCL clock cycle (selected by the RIICnMR3.RDRFS bit)
- When the received slave address matches after a start condition (or a restart condition) is detected with the RIICnCR2.TRS bit cleared to 0

[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 (or a restart condition) is detected
  - When the RIIC enters transmit mode from receive mode
  - When 1 is written to the RIICnCR2.TRS bit while the RIICnMR1.MTWP bit is 1
- When the received slave address matches while the TRS bit is 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.

18.3.12 RIICnSARy — I2C Slave Address Register y (y = 0 to 2)

Access: RIICnSARy is a 32-bit readable/writable register.
RIICnSARyL and RIICnSARyH are 16-bit readable/writable registers.
RIICnSARyLL, RIICnSARyLH, RIICnSARyHL, and RIICnSARyHH are 8-bit readable/writable registers.

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 18.21 RIICnSARy Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. The write value should be 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	These bits are read as the value after reset. The write value should be the value after reset.
9 to 1	SVA[9:1]	7-Bit Address/10-Bit Address Lower 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 Lower 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.

18.3.13 RIICnBRL — I2C Bus Bit Rate Low-Level Register

Access: RIICnBRL is a 32-bit readable/writable register.
RIICnBRLL and RIICnBRLH are 16-bit readable/writable registers.
RIICnBRLLL, RIICnBRLLH, RIICnBRLHL, and RIICnBRLHH are 8-bit readable/writable registers.

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 18.22 RIICnBRL Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	These bits are read as the value after reset. The write value should be 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 18.10, Automatically 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])

18.3.14 RIICnBRH — I2C Bus Bit Rate High-Level Register

Access: RIICnBRH is a 32-bit readable/writable register.
RIICnBRHL and RIICnBRHH are 16-bit readable/writable registers.
RIICnBRHLL, RIICnBRHLH, RIICnBRHHL, and RIICnBRHHH are 8-bit readable/writable registers.

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 18.23 RIICnBRH Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	These bits are read as the value after reset. The write value should be 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 rising time [ns]*2
tr: SCL line rising time [ns]*2
nf: Digital noise filter stage

Note 1. As for IICφ, see CKS[2:0] in **Section 18.3.4, 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 18.24 lists examples of the RIICnBRH and RIICnBRL register settings when the SCL synchronization circuit is not used.

Table 18.24 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	FC_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.

18.3.15 RIICnDRT — I2C Bus Transmit Data Register

Access: RIICnDRT is a 32-bit readable/writable register.
RIICnDRTL and RIICnDRTH are 16-bit readable/writable registers.
RIICnDRTLL, RIICnDRTLH, RIICnDRTHL, and RIICnDRTHH are 8-bit readable/writable registers.

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.

18.3.16 RIICnDRR — I2C Bus Receive Data Register

Access: RIICnDRR is a 32-bit readable register.
RIICnDRRL and RIICnDRRH are 16-bit readable registers.
RIICnDRRLL, RIICnDRRLH, RIICnDRRHL, and RIICnDRRHH are 8-bit readable registers.

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.

18.3.17 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.

18.4 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 18.25 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 18.25 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).

18.5 Operation

18.5.1 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 18.3 shows the I2C bus format, and **Figure 18.4** shows the I2C bus timing.

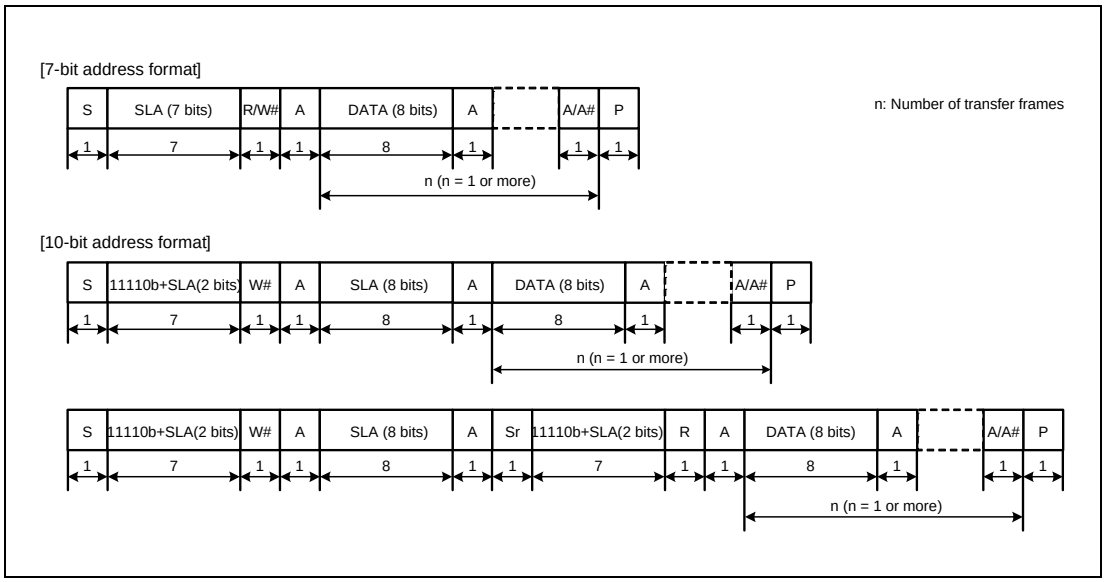


Figure 18.3 I2C Bus Format

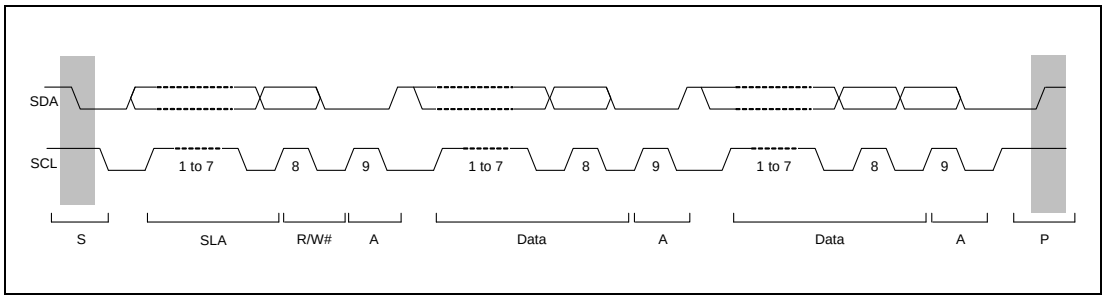


Figure 18.4 I2C Bus Timing (SLA = 7 Bits)

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.)

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.

18.5.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in **Figure 18.5**.

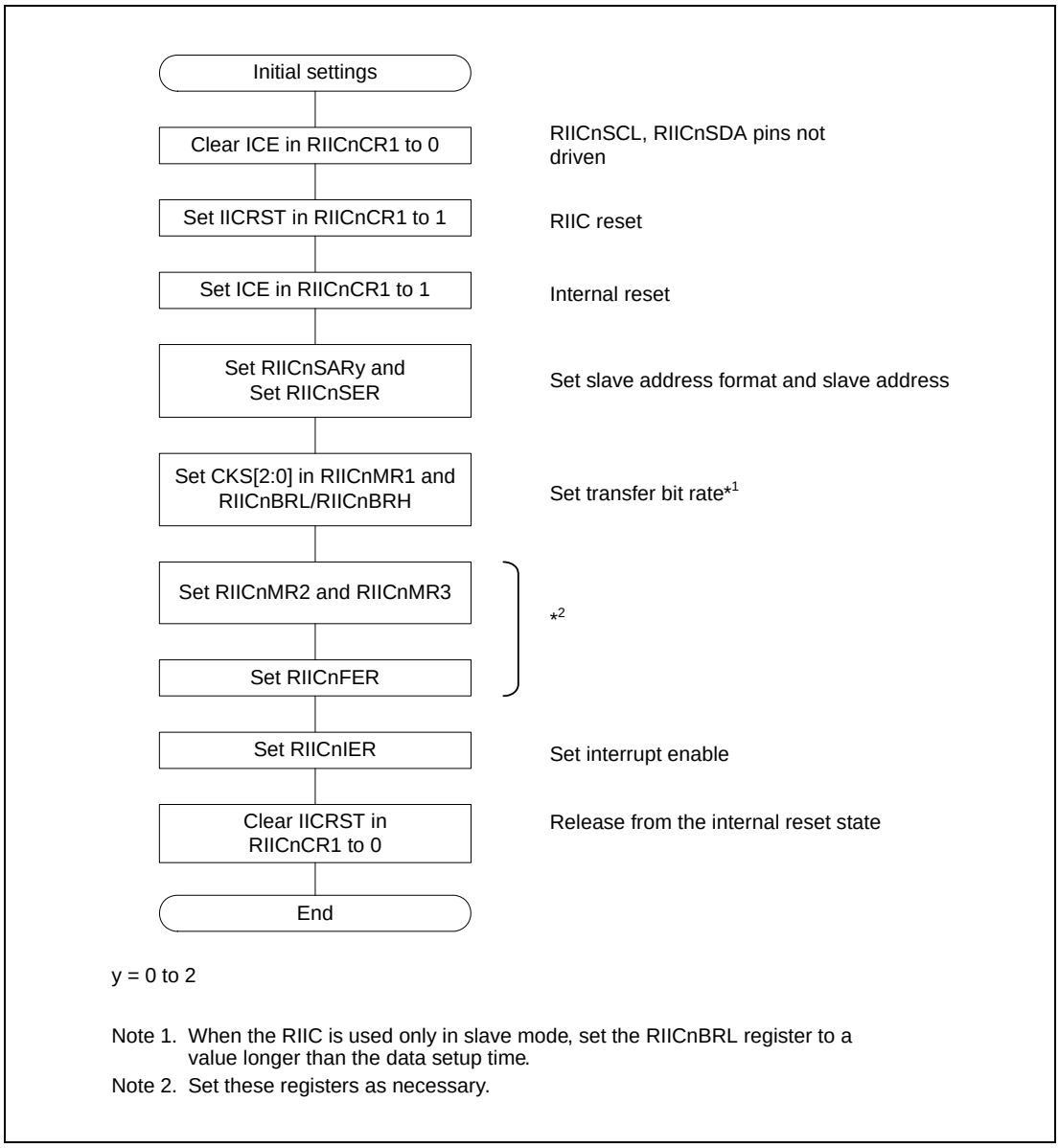


Figure 18.5 Example of RIIC Initialization Flowchart

18.5.3 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 18.6** shows an example of usage of master transmission and **Figure 18.7** to **Figure 18.9** 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 18.5**). 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 bit 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.

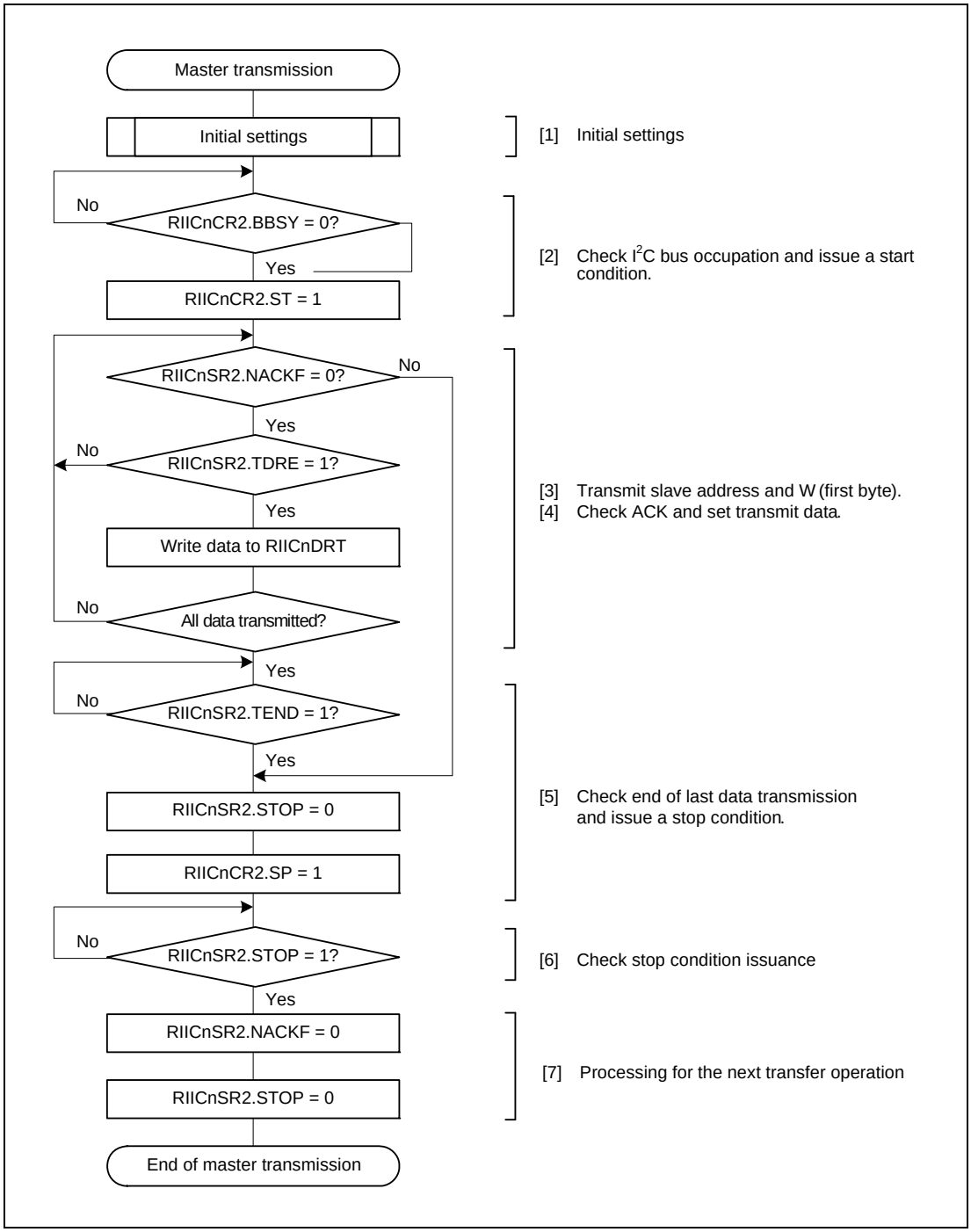


Figure 18.6 Example of Master Transmission Flowchart

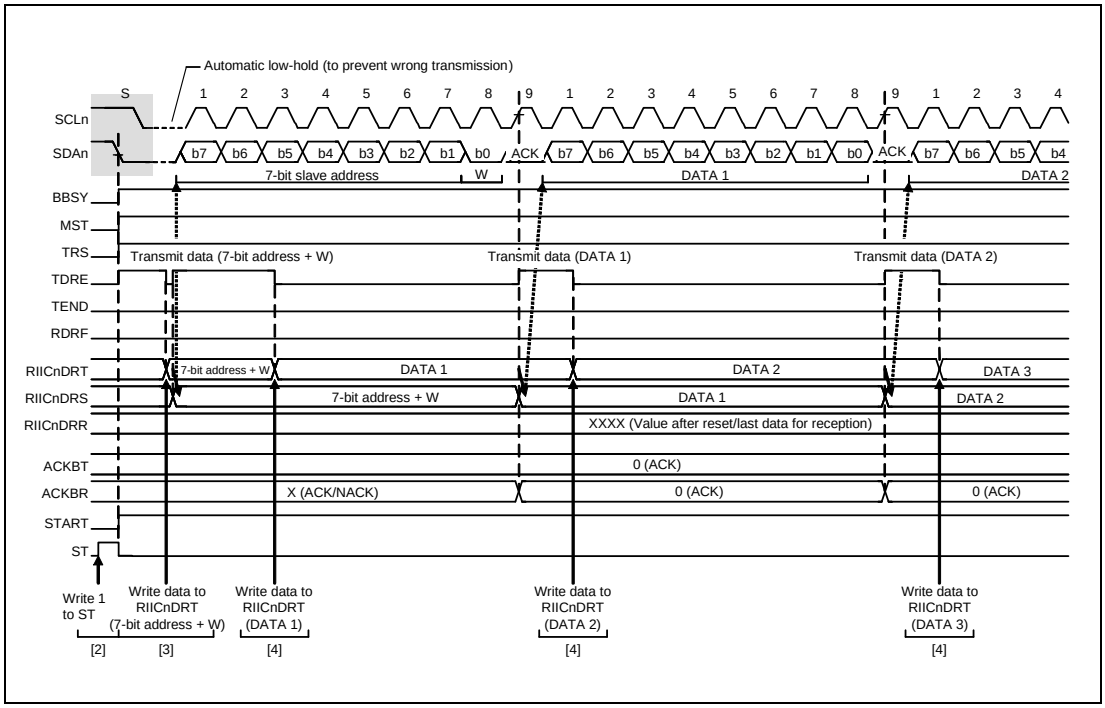


Figure 18.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

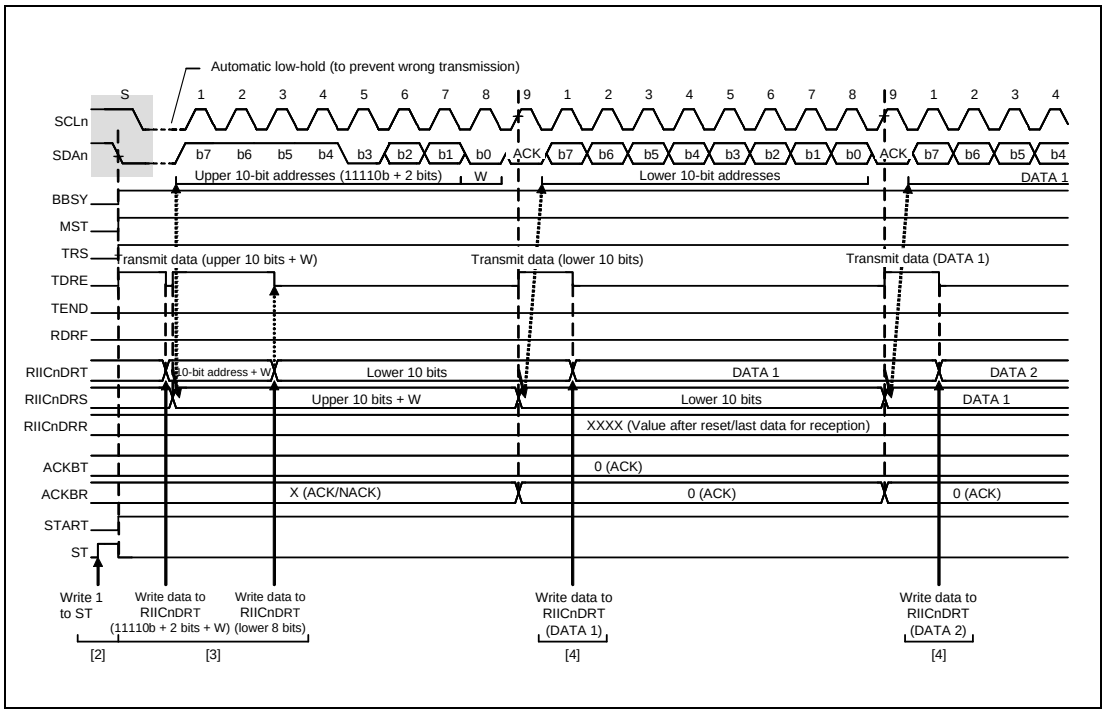


Figure 18.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

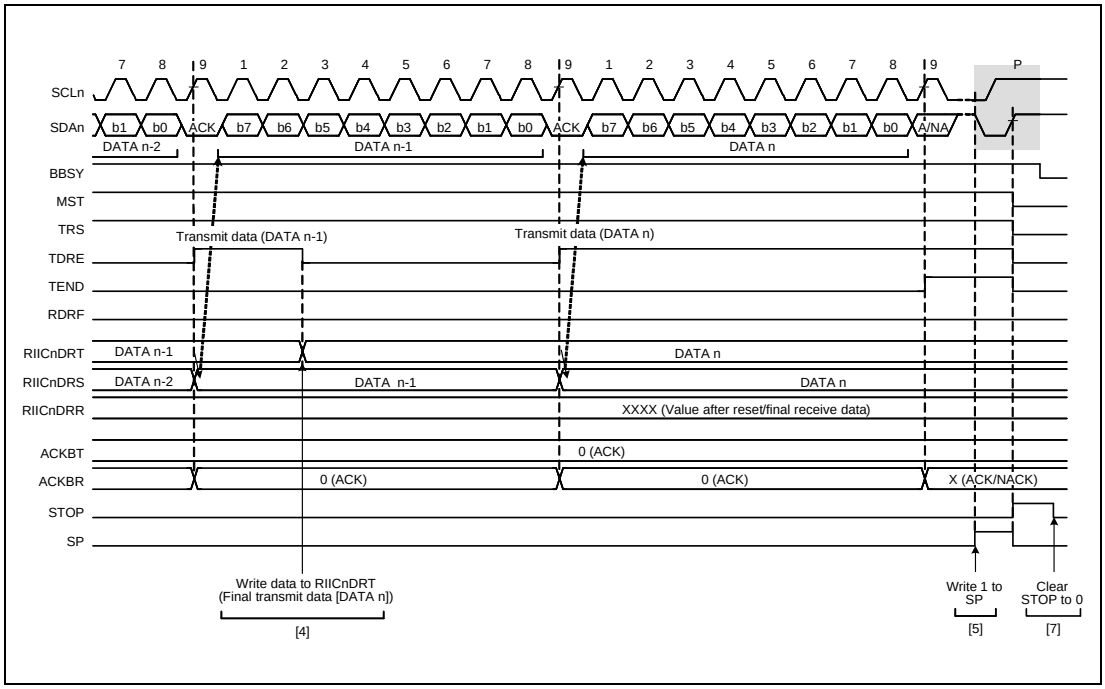


Figure 18.9 Master Transmit Operation Timing (3)

18.5.4 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 18.10 shows an example of master reception flowchart (7-bit address format, 1 or 2 bytes), **Figure 18.11** shows an example of master reception flowchart (7-bit address format, 3 bytes or more), and **Figure 18.12** to **Figure 18.14** 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 18.5**). 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 issue the 10-bit address, and then issue a restart condition. After that, transmitting 1111 0_B, 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.

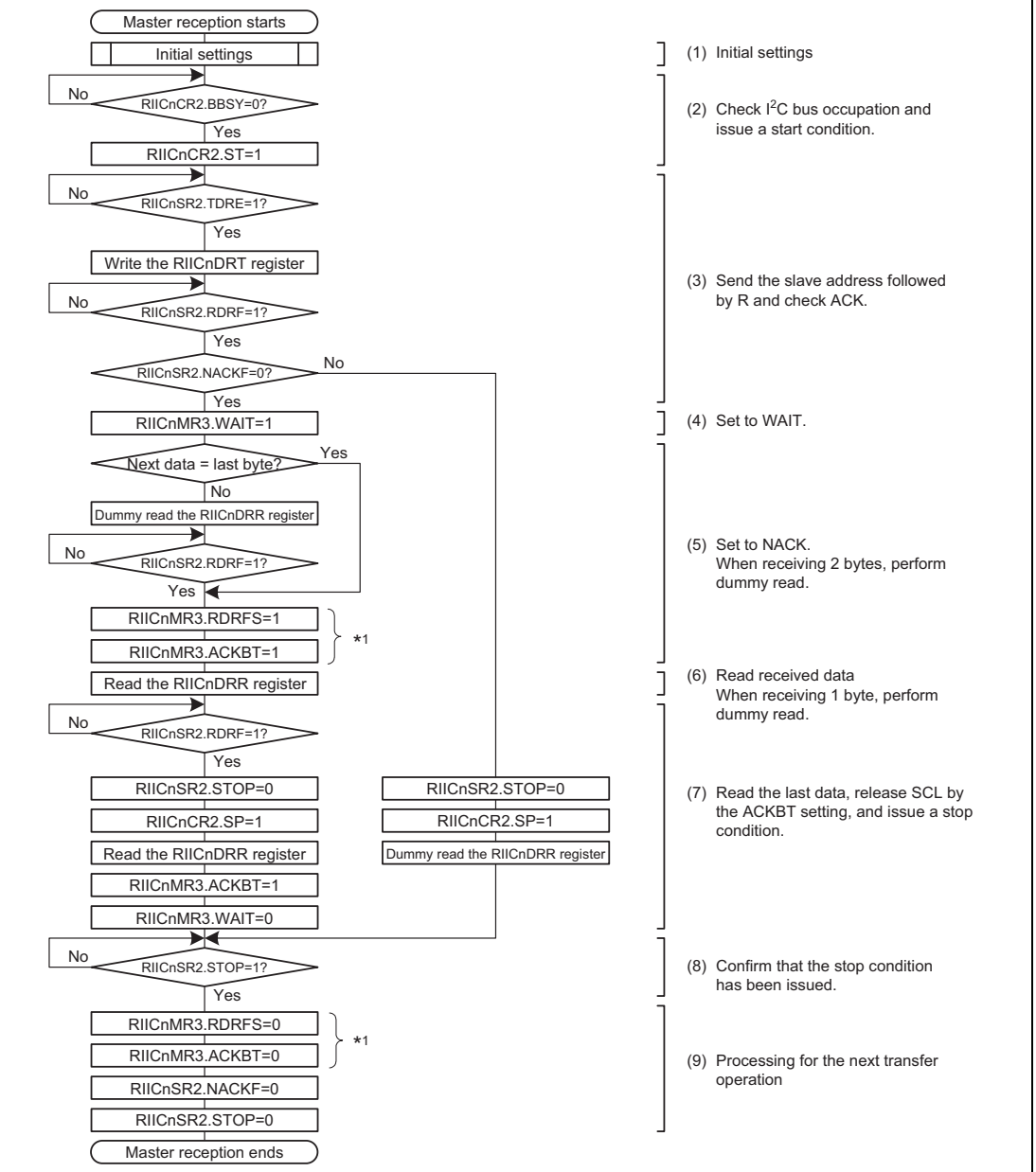
(5) After 1 byte of data has been received, the RIICnSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RIICnMR3.RDRFS bit. Reading out RIICnDRR at this time will produce the received data, and the RDRF flag is automatically cleared 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.



Note 1. These bits can be set at the same time.

Figure 18.10 Example of Master Reception Flowchart (7-Bit Address Format, 1 or 2 Bytes)

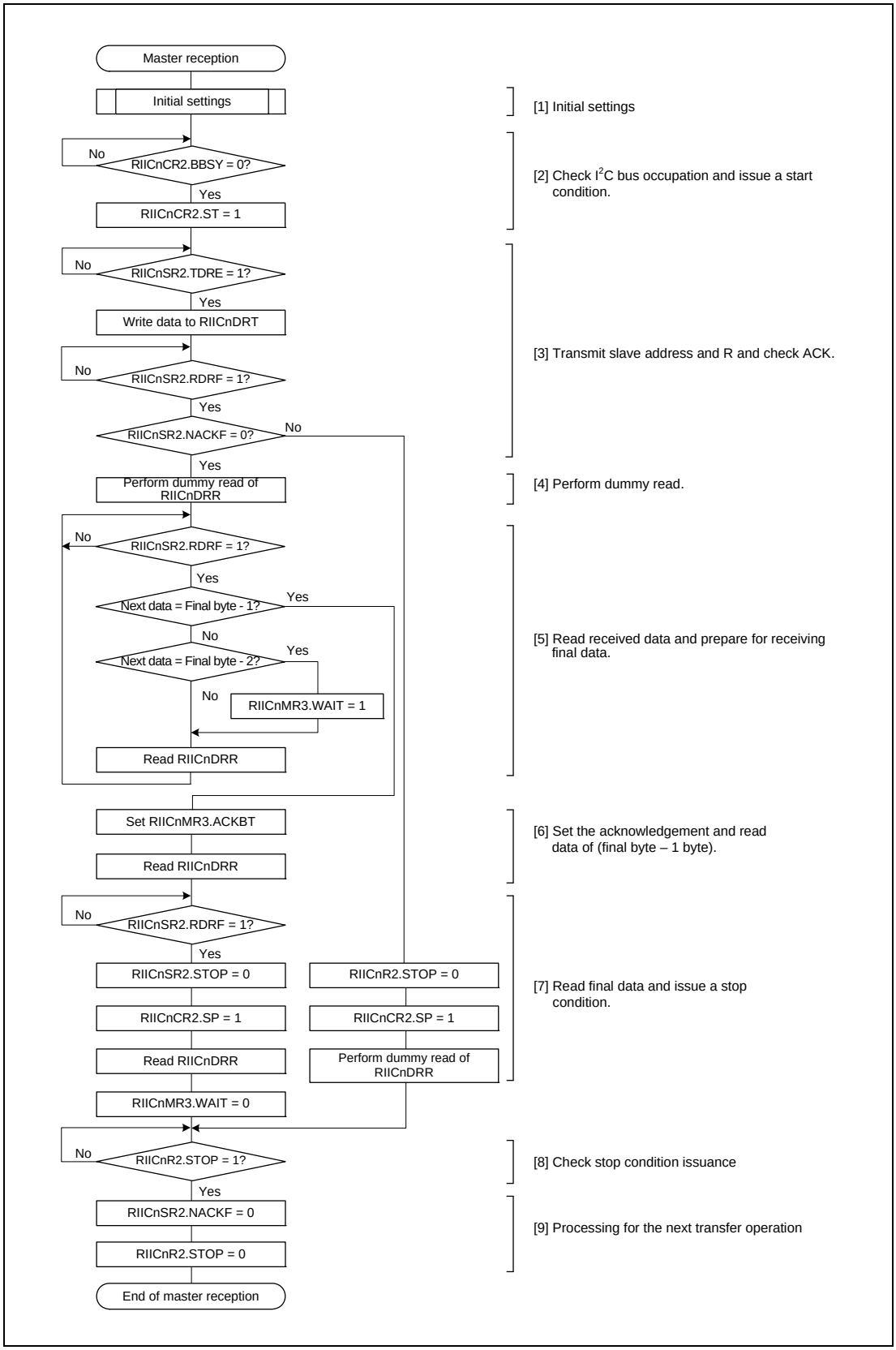


Figure 18.11 Example of Master Reception Flowchart (7-Bit Address Format, 3 Bytes or More)

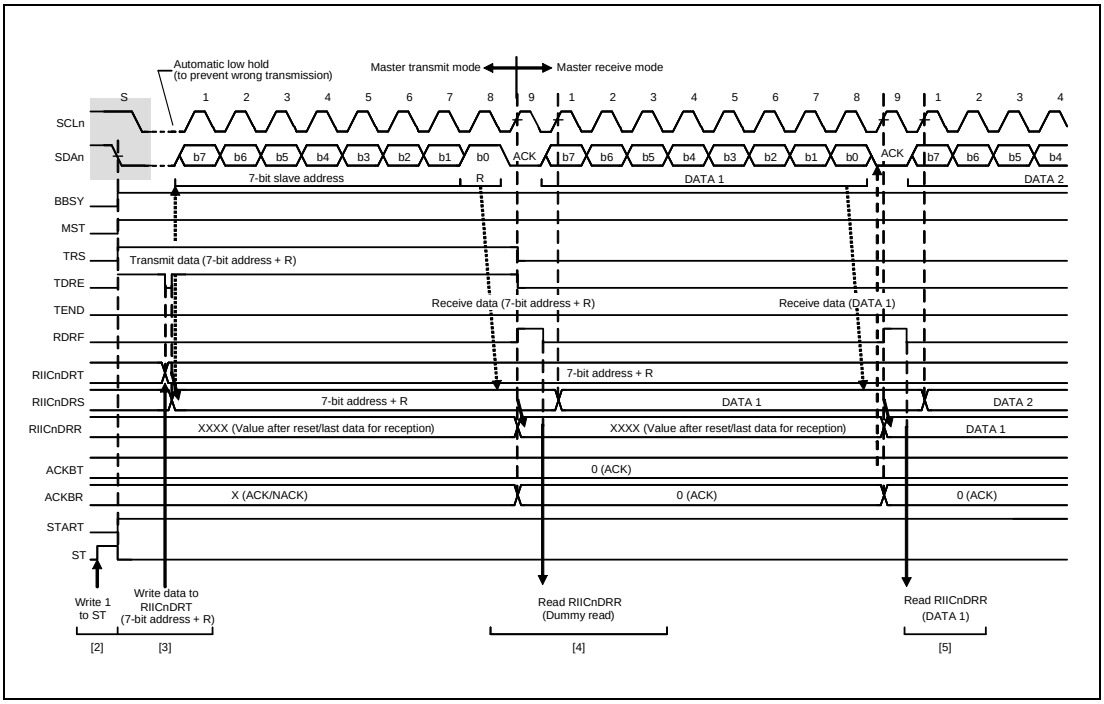


Figure 18.12 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

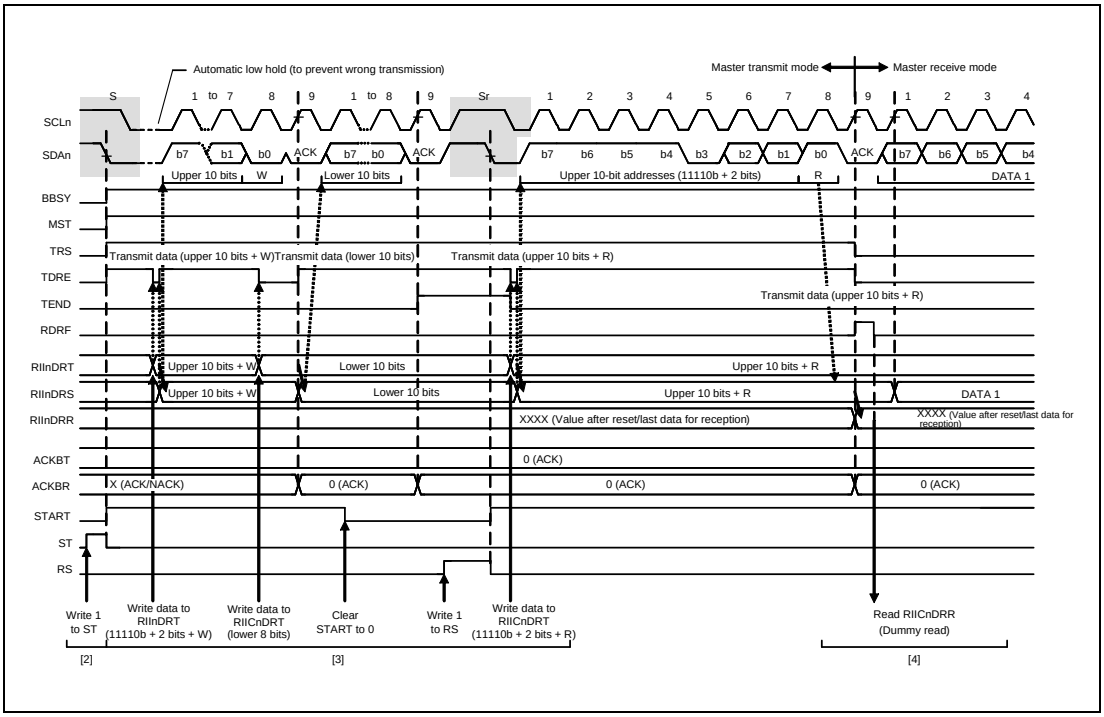


Figure 18.13 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)

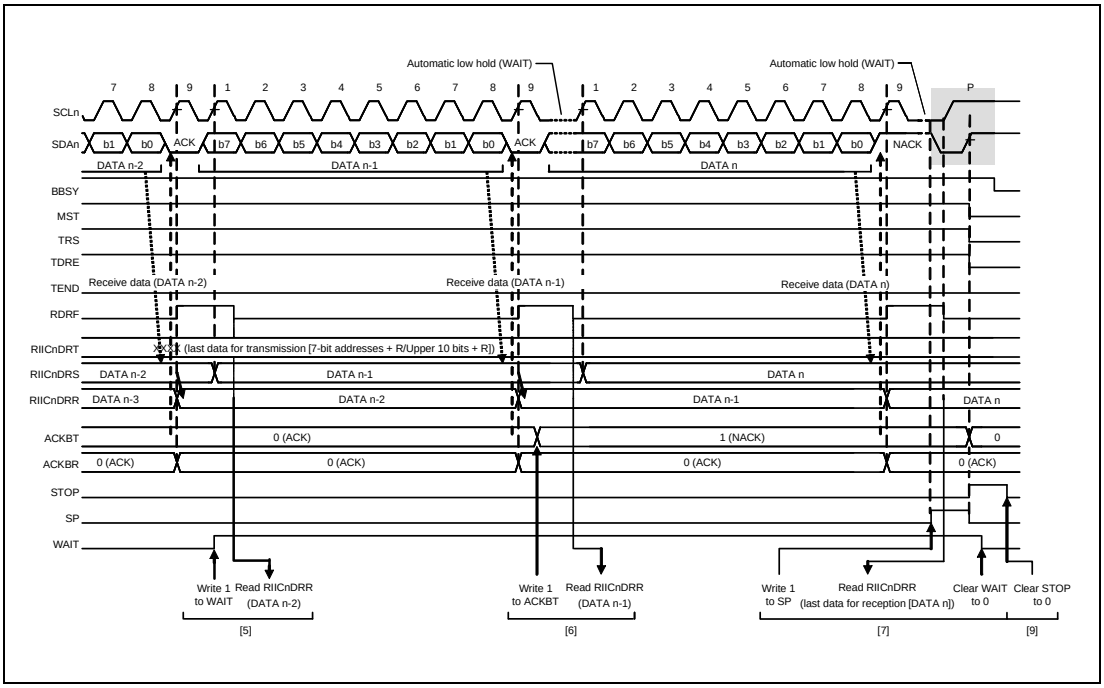


Figure 18.14 Master Receive Operation Timing (3) (when RDRFS = 0)

18.5.5 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 18.5 shows an example of usage of slave transmission and **Figure 18.16** and **Figure 18.17** show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Follow the procedure in **Figure 18.5** 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 will stay in the standby state until it receives a slave address that it matches.

(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.

(7) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

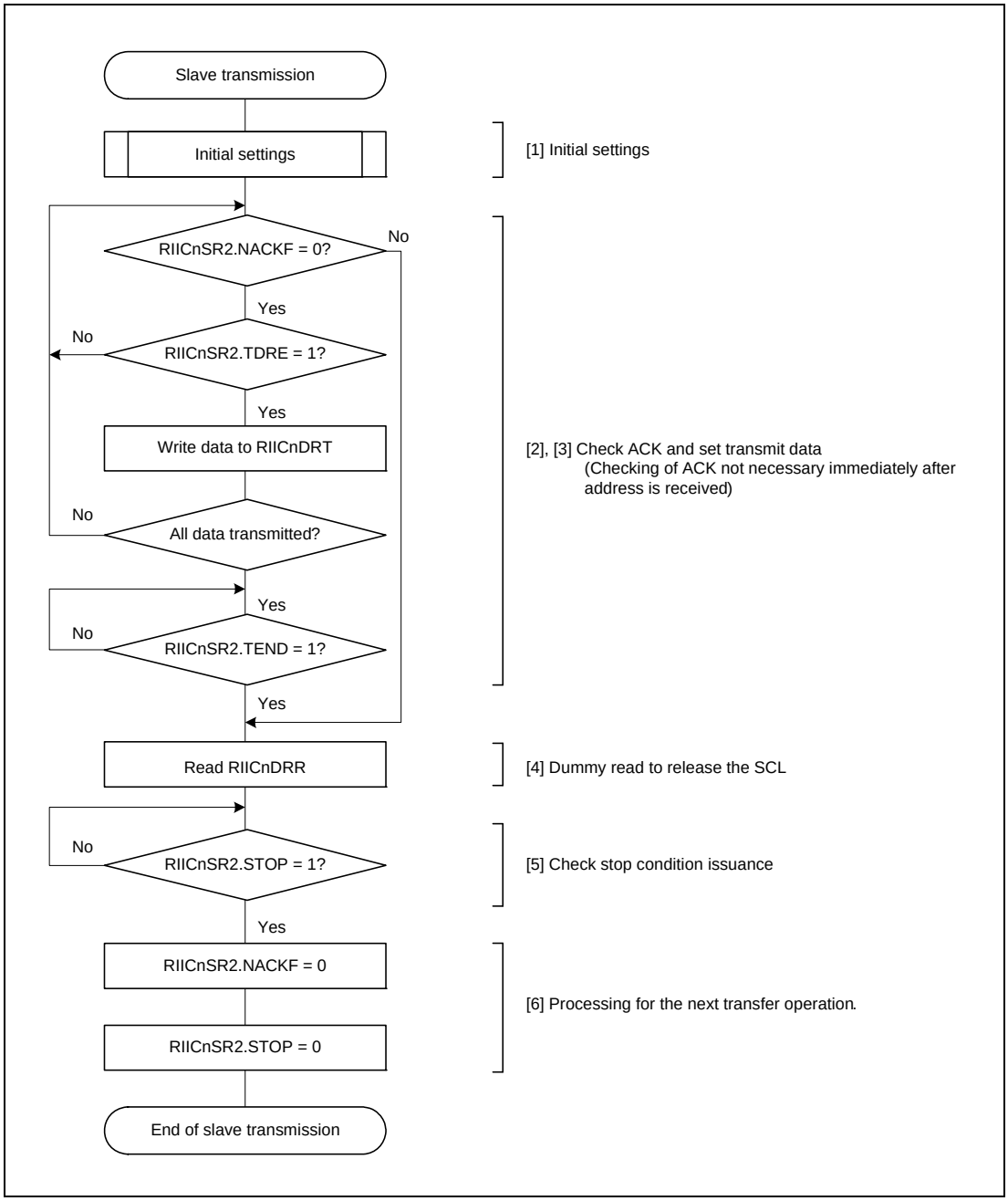


Figure 18.15 Example of Slave Transmission Flowchart

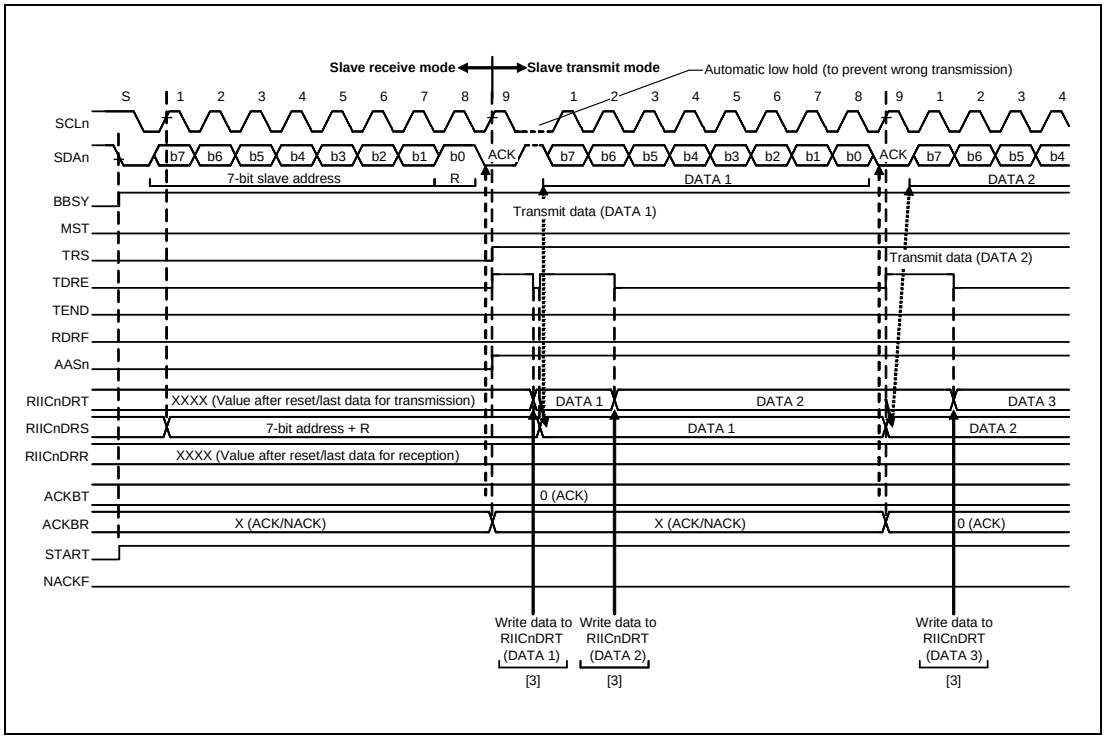


Figure 18.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

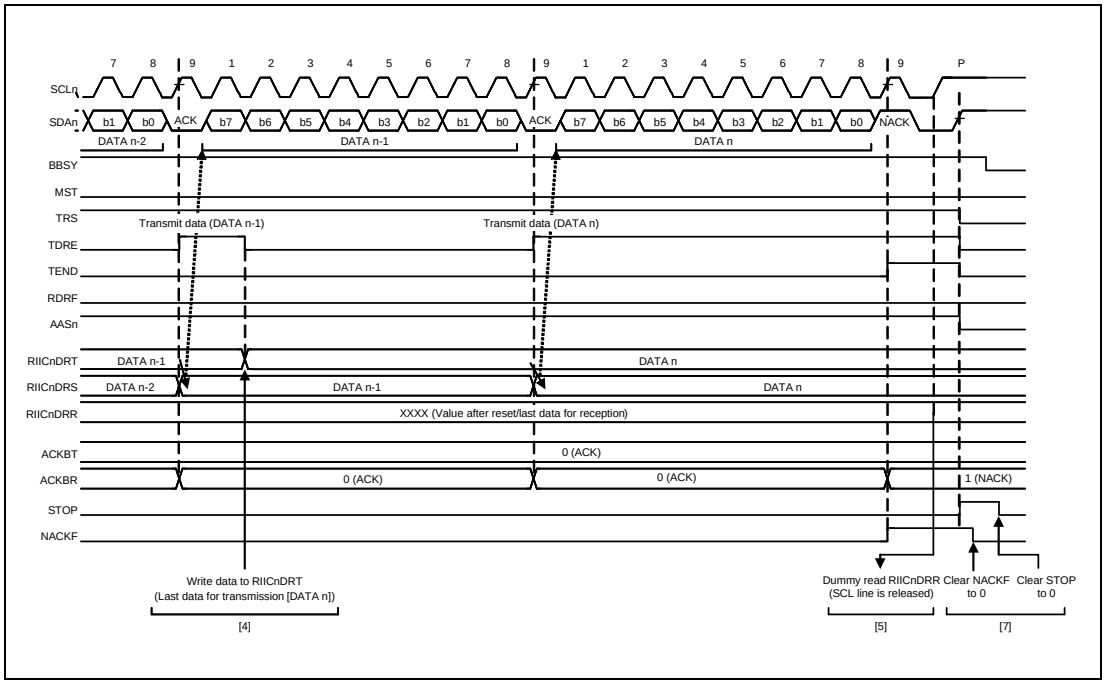


Figure 18.17 Slave Transmit Operation Timing (2)

18.5.6 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 18.18 shows an example of usage of slave reception and **Figure 18.19** and **Figure 18.20** show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Follow the procedure in **Figure 18.5** 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 will stay in the standby state until it receives a slave address that it matches.

(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.

(6) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.STOP flag to 0 for the next transfer operation.

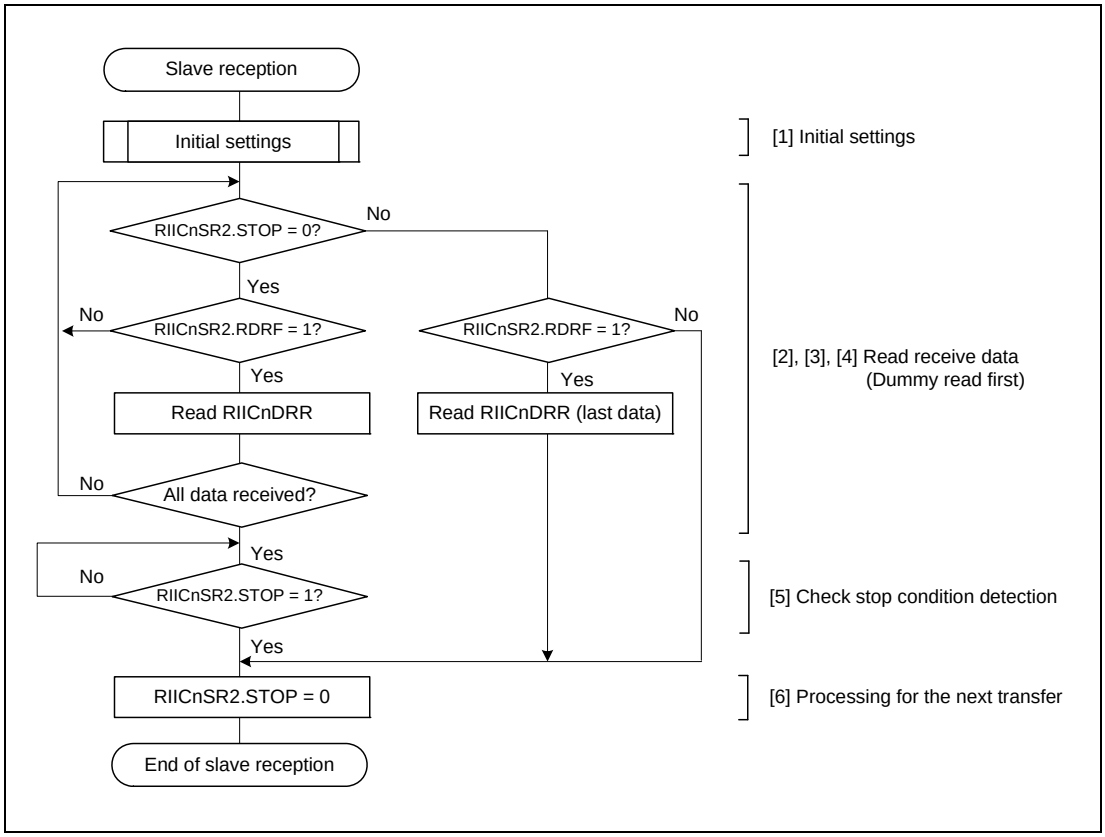


Figure 18.18 Example of Slave Reception Flowchart

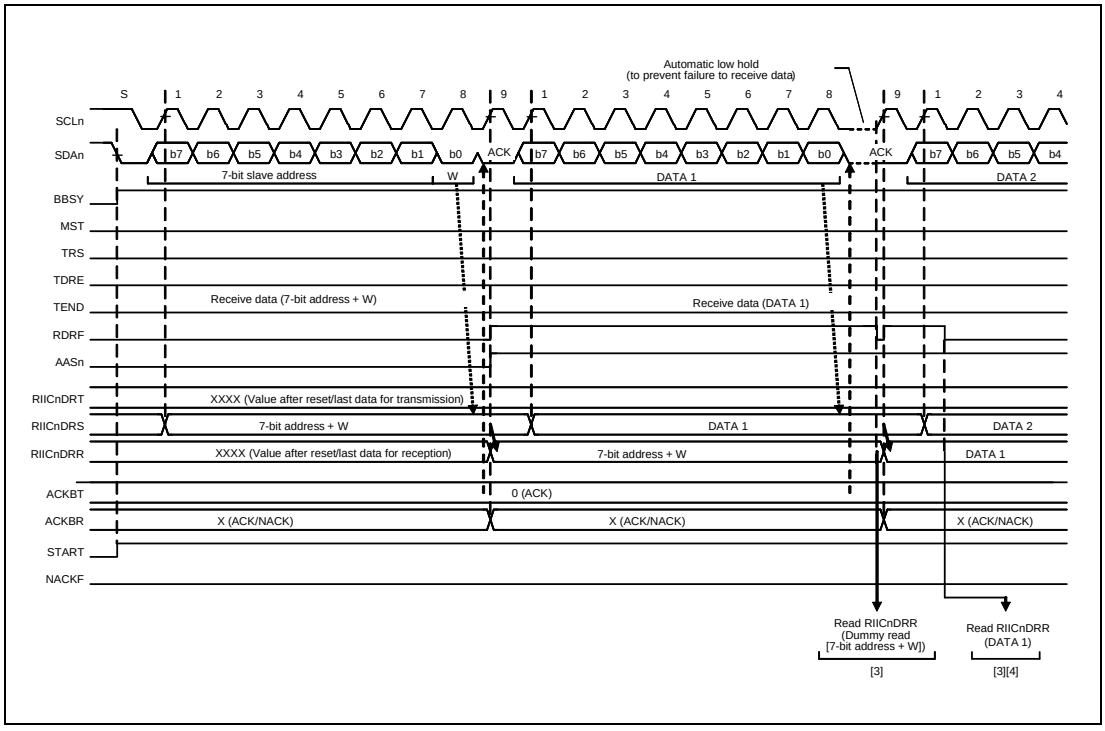


Figure 18.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

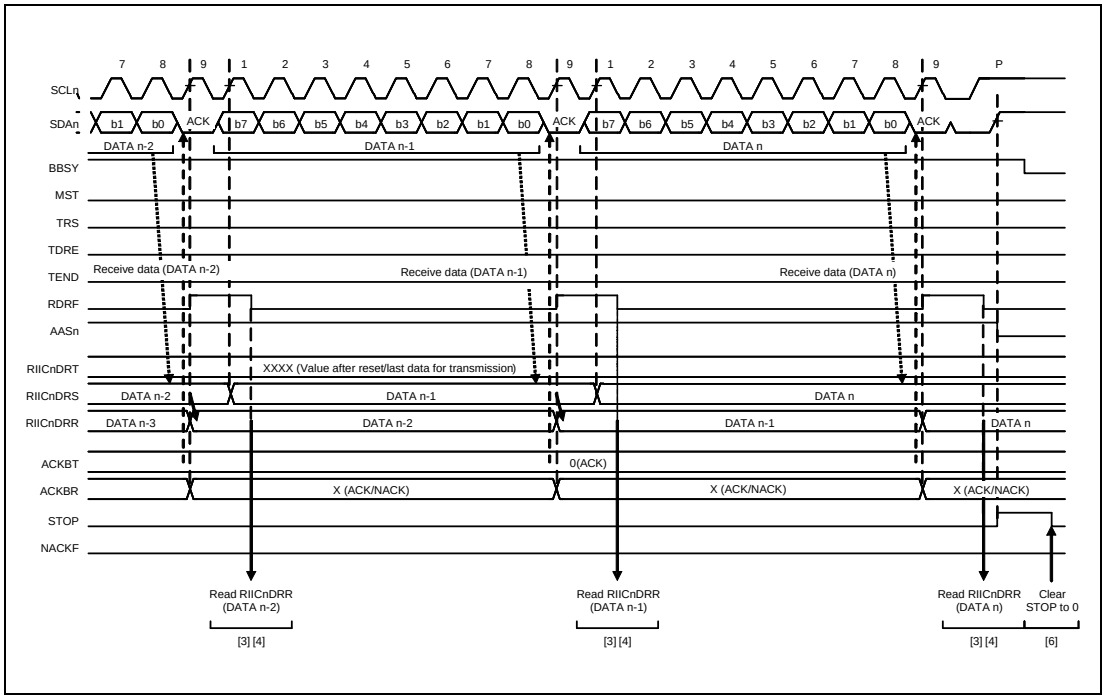


Figure 18.20 Slave Receive Operation Timing (2) (when RDRFS = 0)

18.6 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.

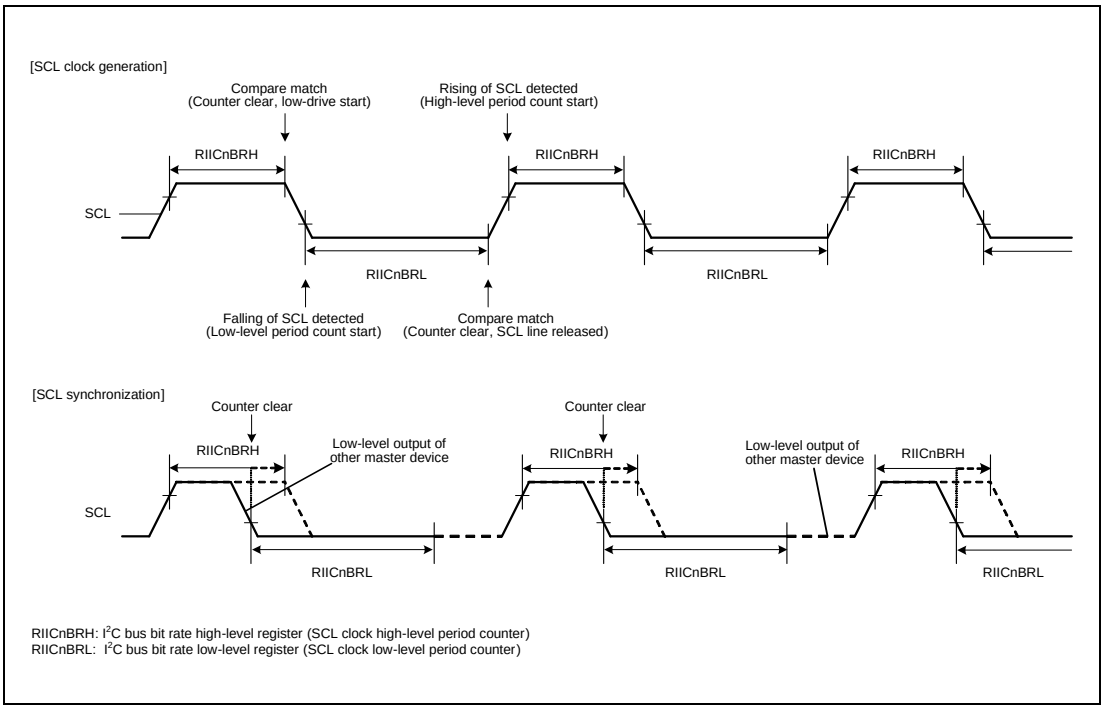


Figure 18.21 Generation and Synchronization of the SCL Signal from the RIIC

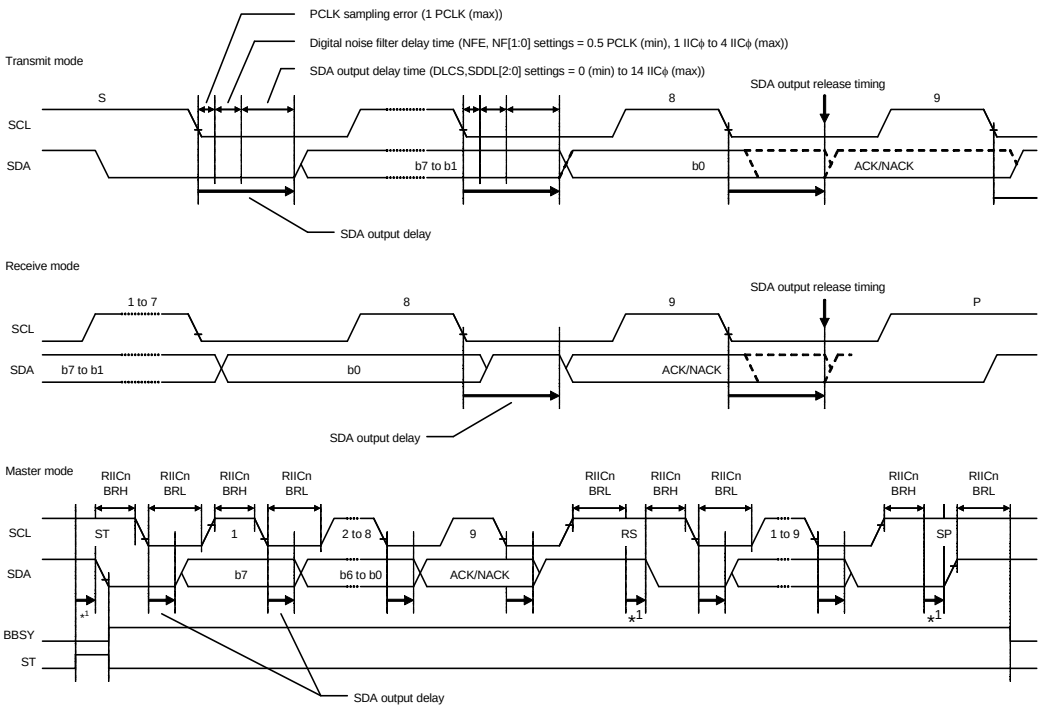
18.7 Facility for Delaying SDA Output

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 DLCS, and SDDL[2:0] bits when a start (ST), restart (RS), or stop (SP) condition is issued.

Figure 18.22 SDA Output Delay Facility

18.8 Digital Noise-Filter Circuits

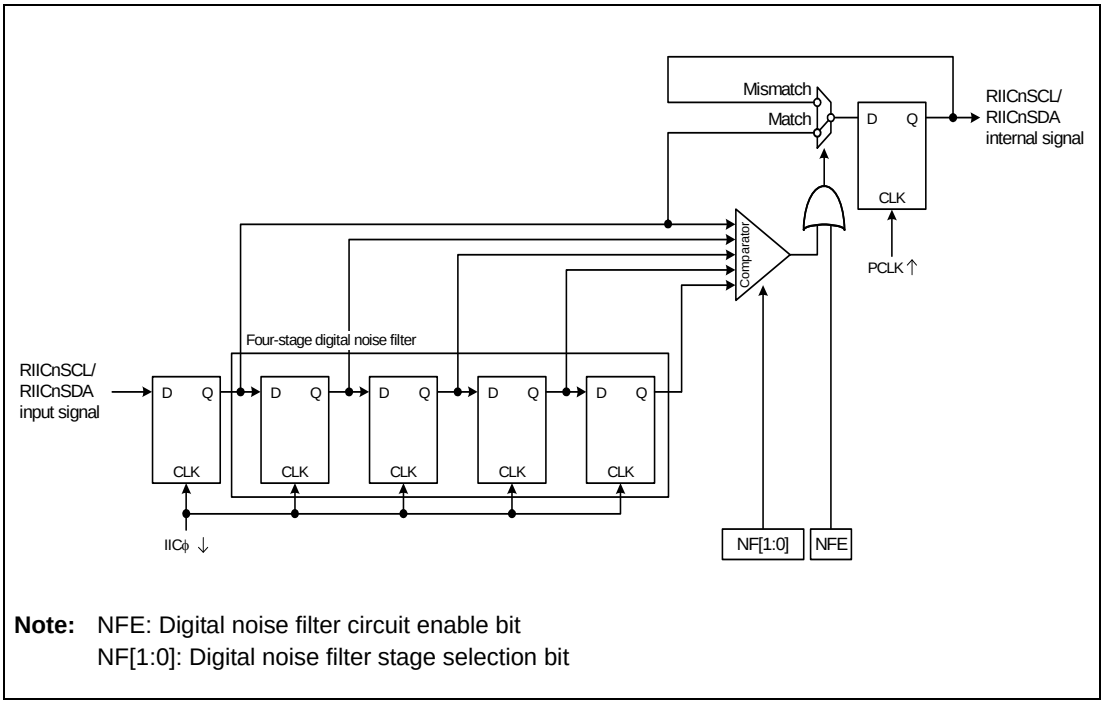
The states of the RIICnSCL and RIICnSDA pins are conveyed to the internal circuitry through the digital noise-filter circuit. **Figure 18.23** is a block diagram of the digital noise-filter circuit.

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 18.23 Block Diagram of Digital Noise Filter Circuit

18.9 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.

18.9.1 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 falling edge of the ninth SCL clock cycle, and 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 18.24 to **Figure 18.26** show the AASy flag set timing in three cases.

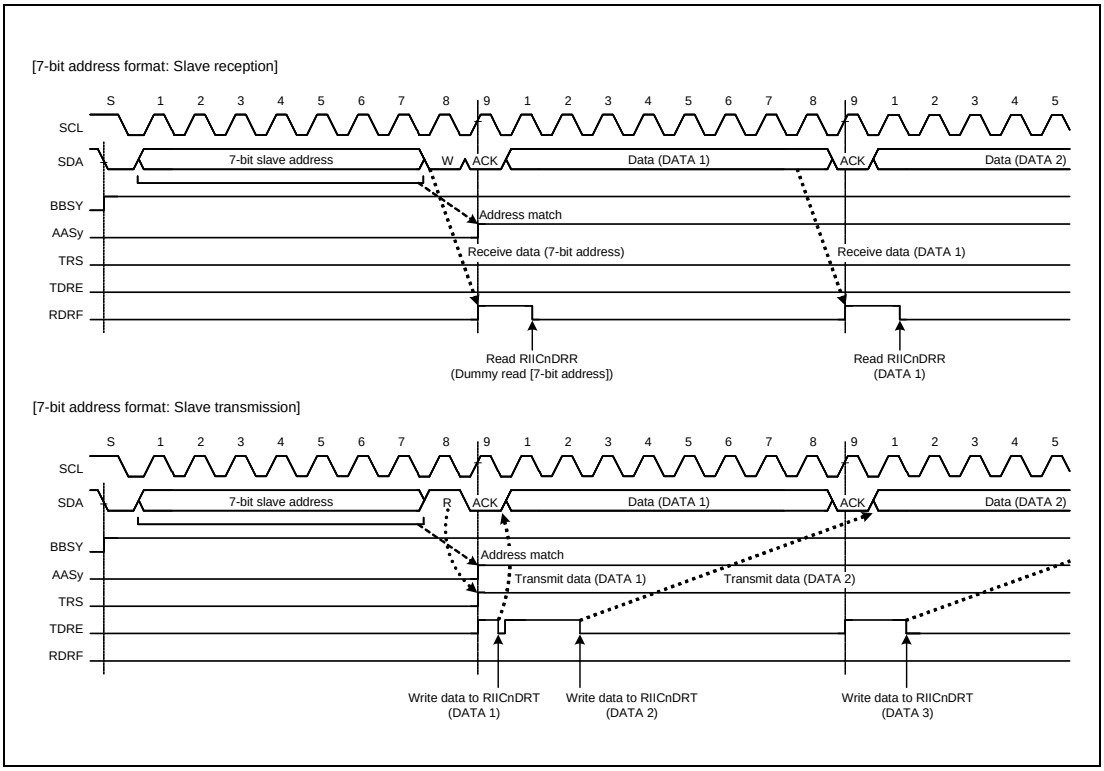


Figure 18.24 AASy Flag Set Timing with 7-Bit Address Format Selected

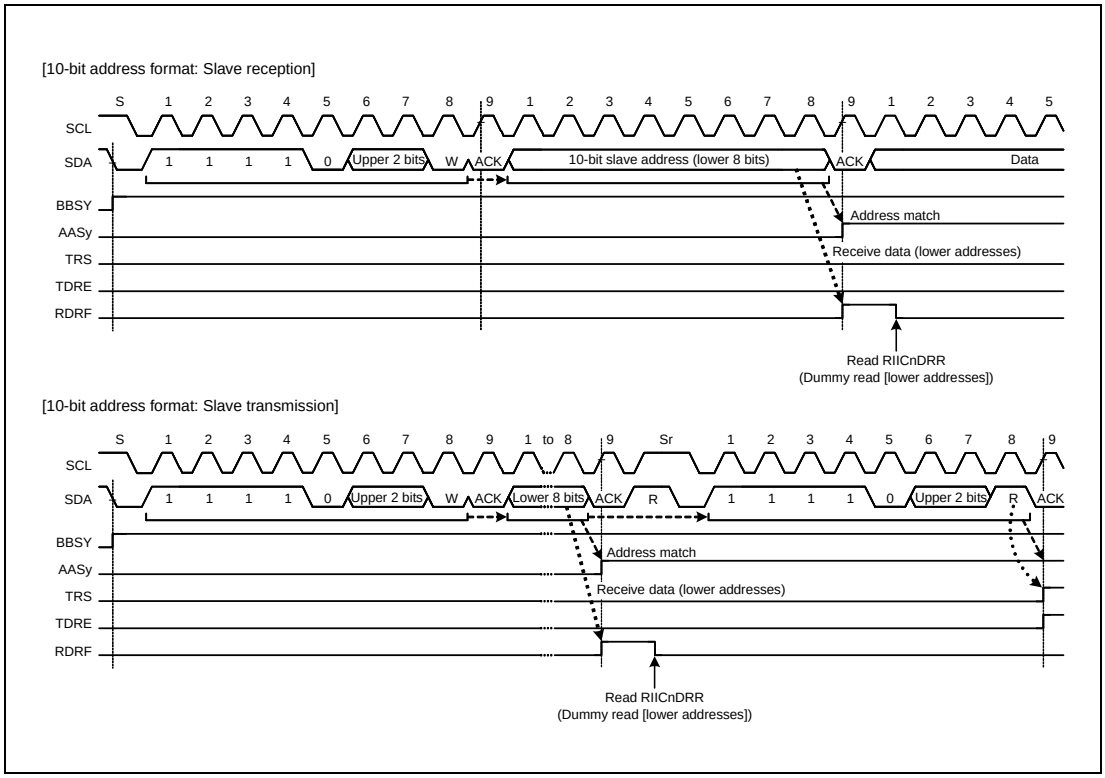


Figure 18.25 AASy Flag Set Timing with 10-Bit Address Format Selected

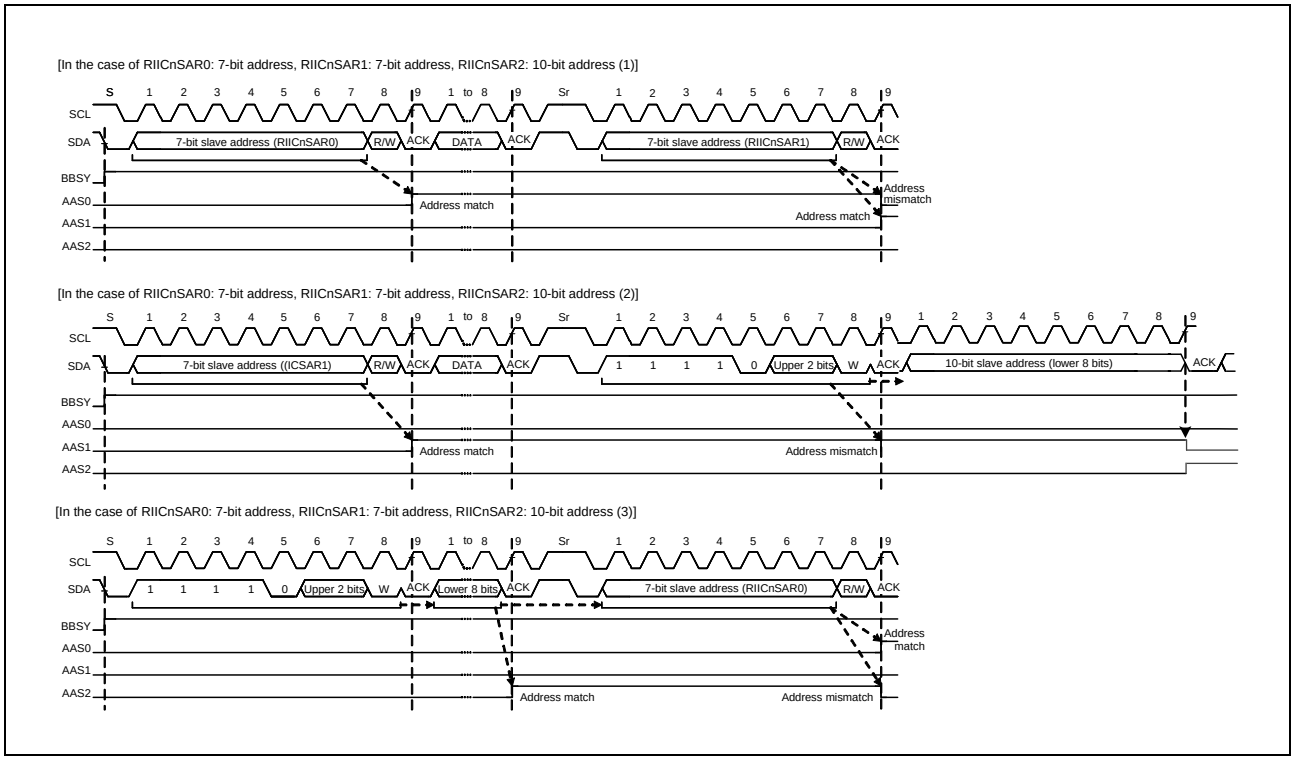


Figure 18.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

18.9.2 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.

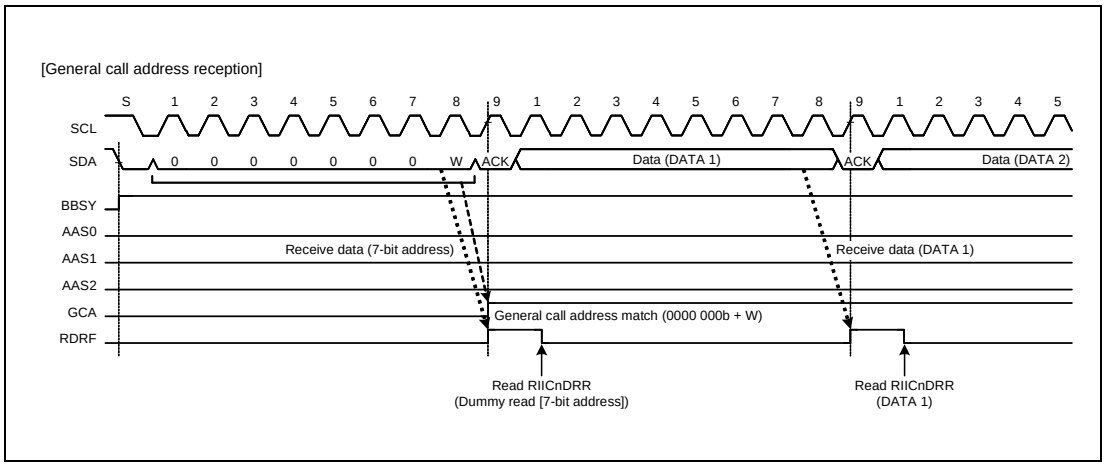


Figure 18.27 Timing of GCA Flag Setting during Reception of General Call Address

18.9.3 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.

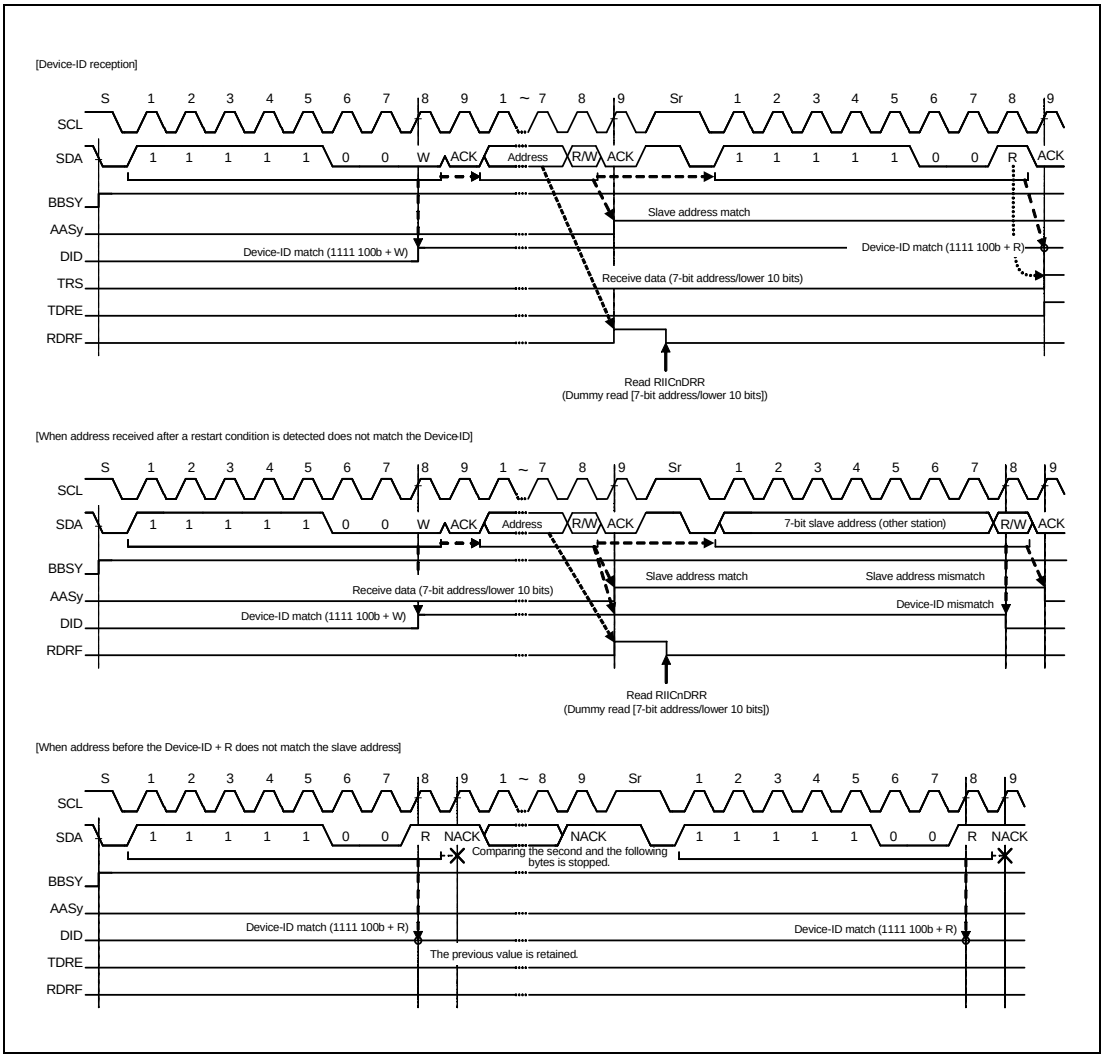


Figure 18.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

18.10 Automatically Low-Hold Function for SCL

18.10.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (RIICnDRS) is empty when data have not been written to the transmit data register (RIICnDRT) with the RIIC in transmission mode (RIICnCR2.TRS bit = 1), the SCL signal is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

<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

<Slave transmit mode>

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

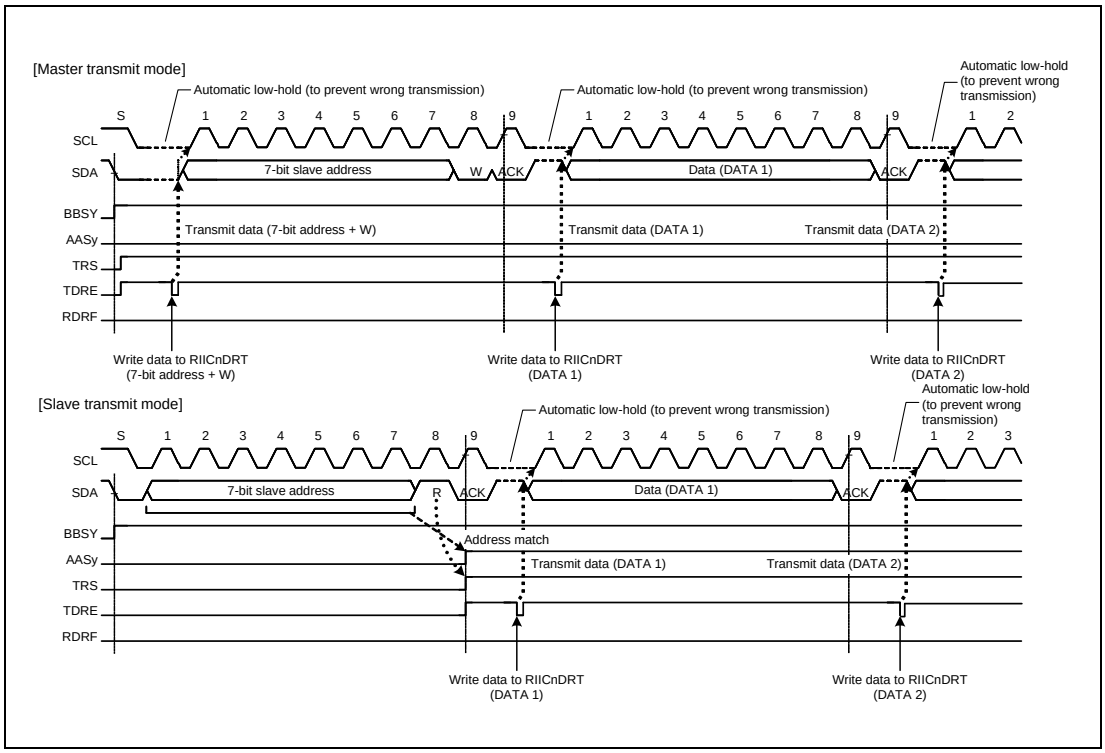


Figure 18.29 Automatic Low-Hold Operation in Transmit Mode

18.10.2 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.

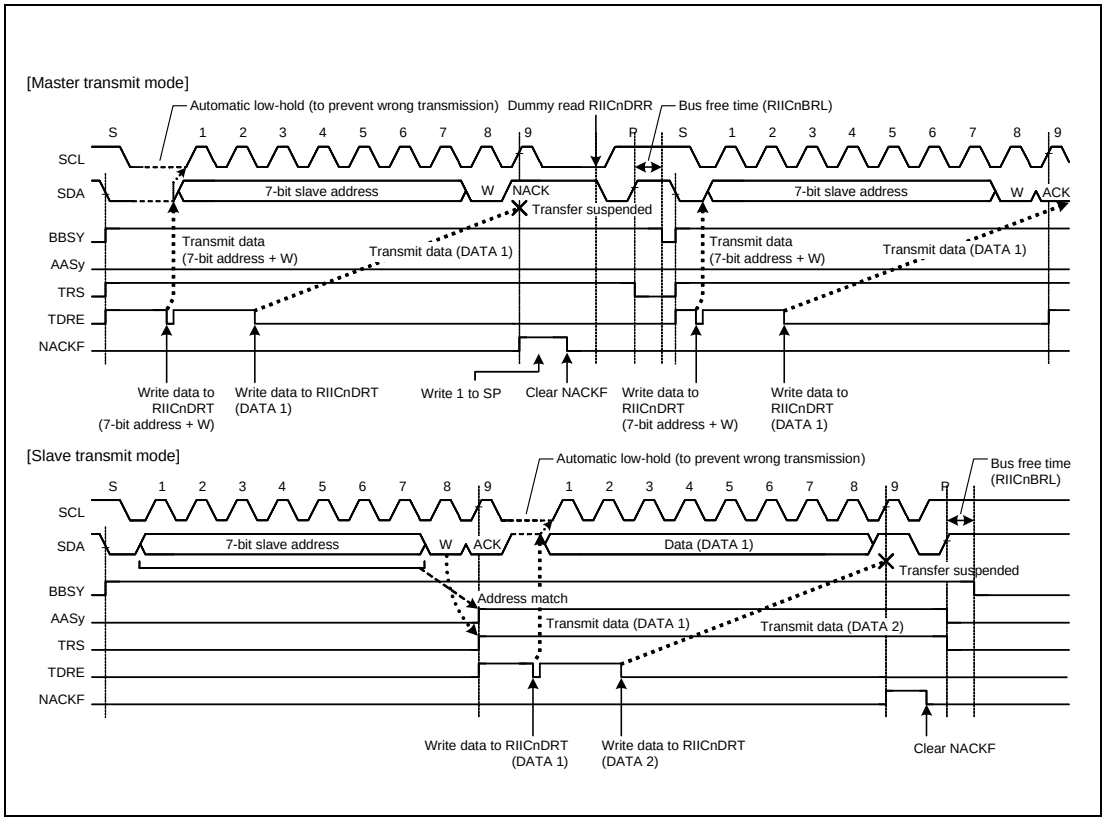


Figure 18.30 Suspension of Data Transfer when NACK is Received (NACKE = 1)

18.10.3 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.

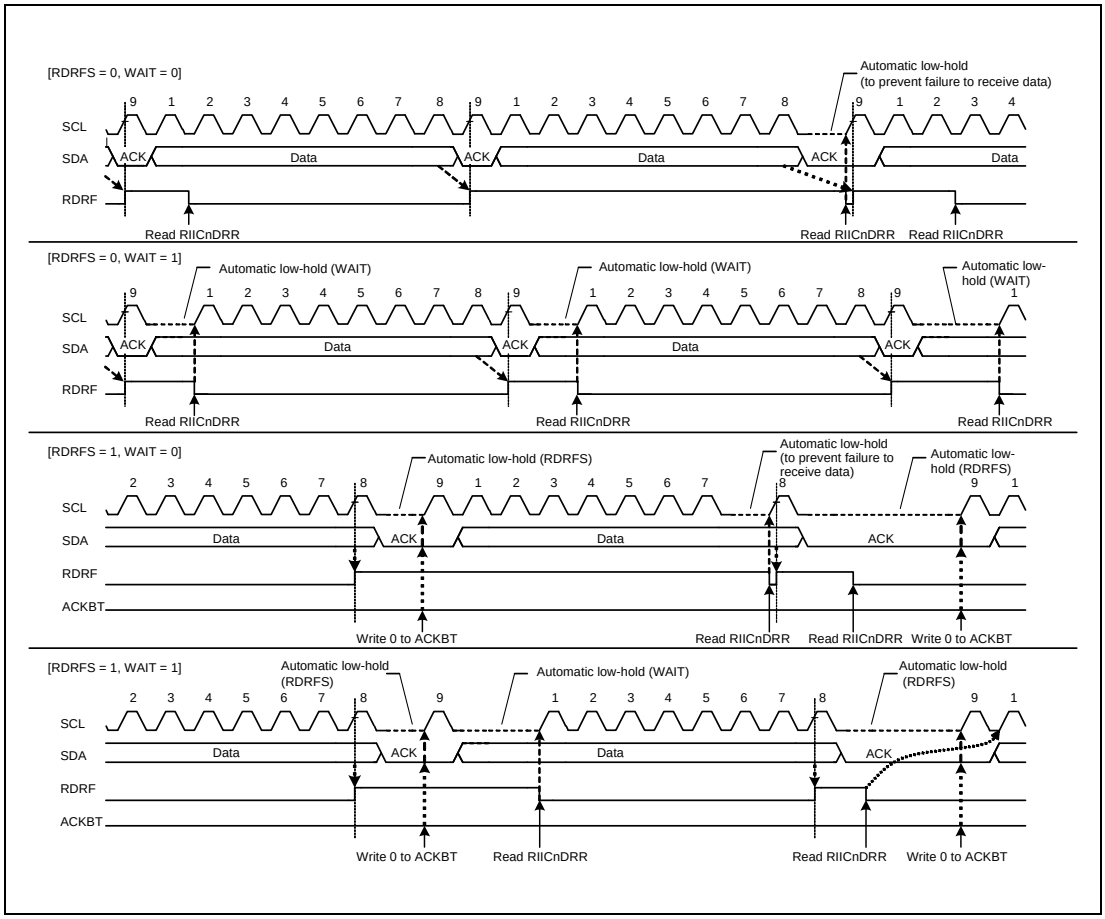


Figure 18.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

18.11 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.

18.11.1 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)

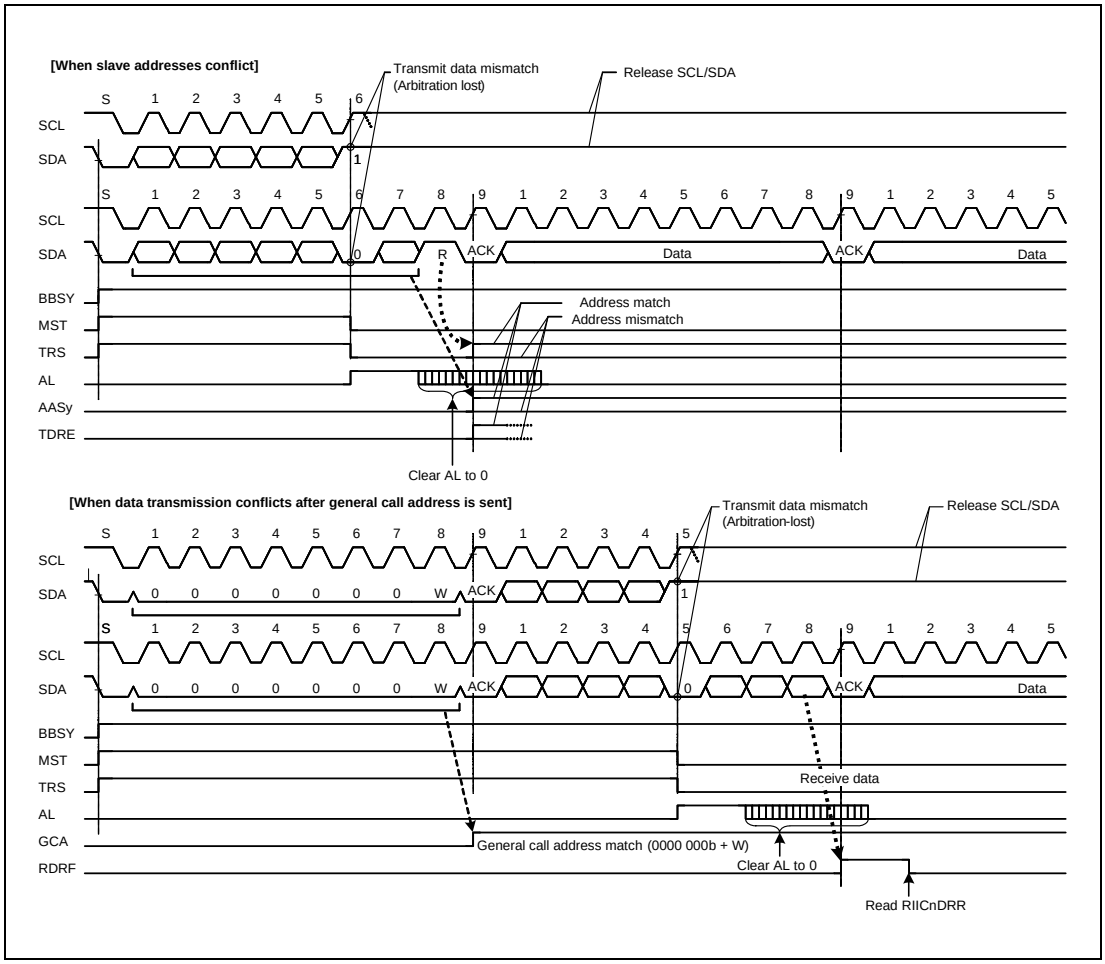


Figure 18.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

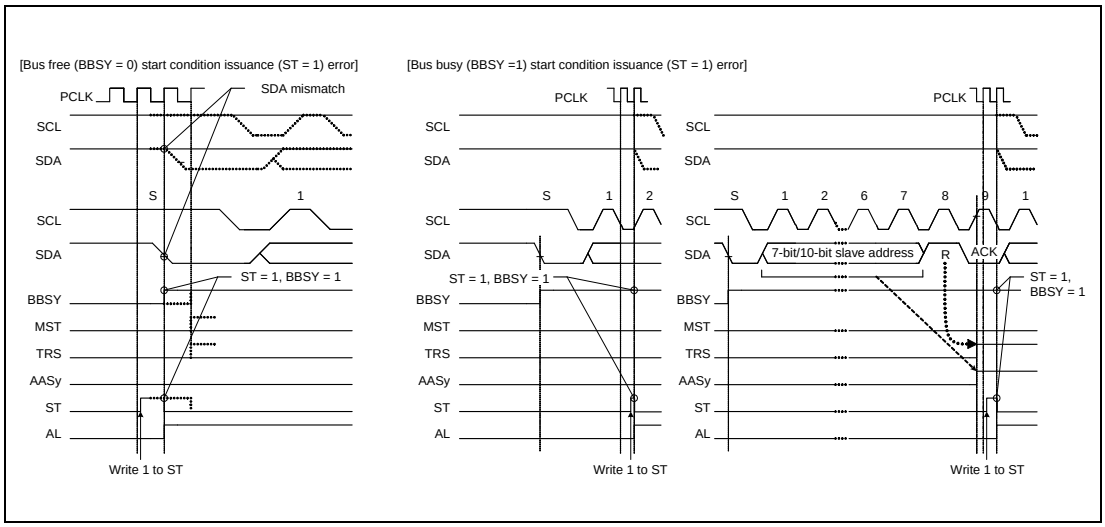


Figure 18.33 Arbitration-Lost when a Start Condition is Issued (MALE = 1)

18.11.2 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 18.34** shows an example of arbitration-lost detection during transmission of NACK.

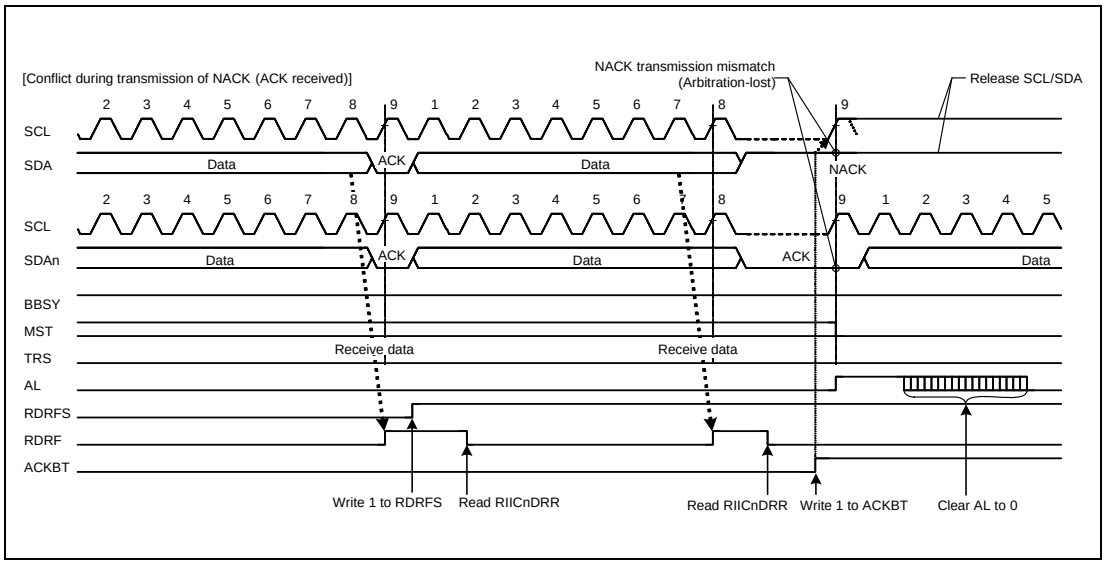


Figure 18.34 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

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 immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

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)

18.11.3 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 is immediately released from the slave-matched state and 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)

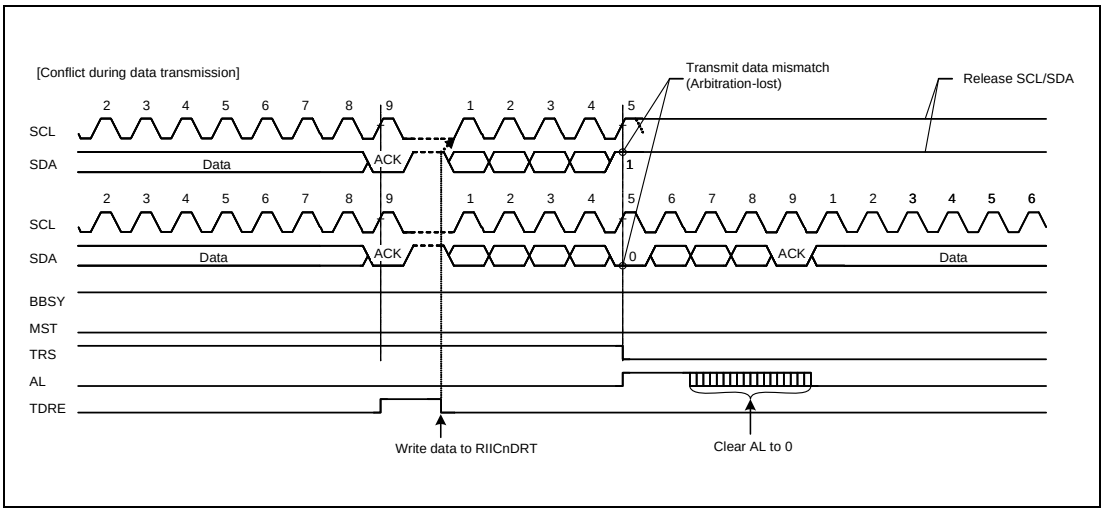


Figure 18.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

18.12 Start Condition/Restart Condition/Stop Condition Issuing Function

18.12.1 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.

18.12.2 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 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).

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.

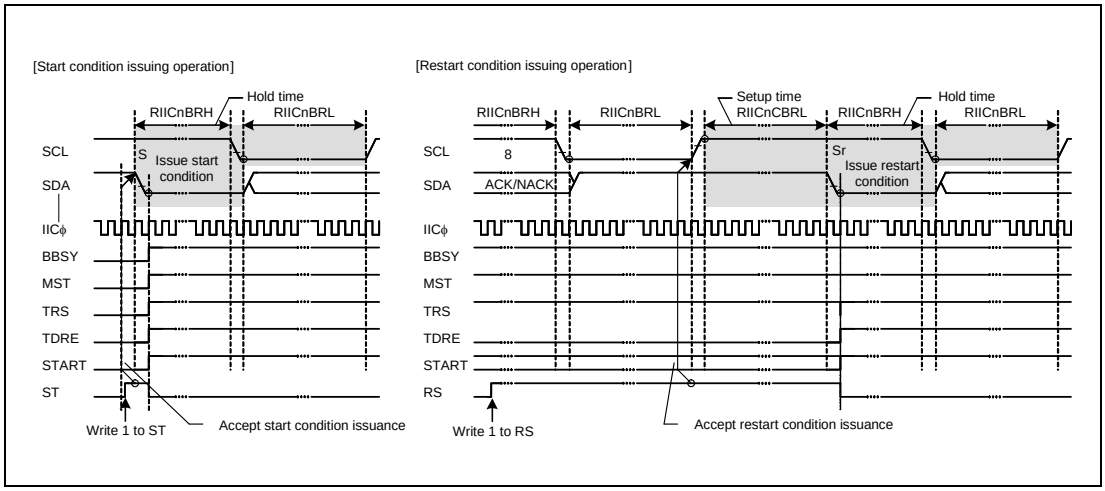


Figure 18.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

18.12.3 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).

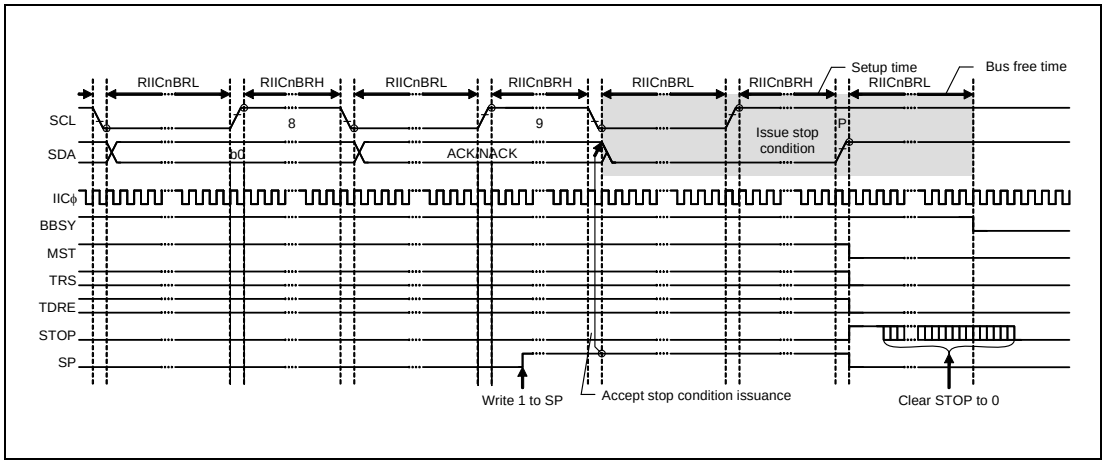


Figure 18.37 Stop Condition Issue Timing (SP Bit)

18.13 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.

18.13.1 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.

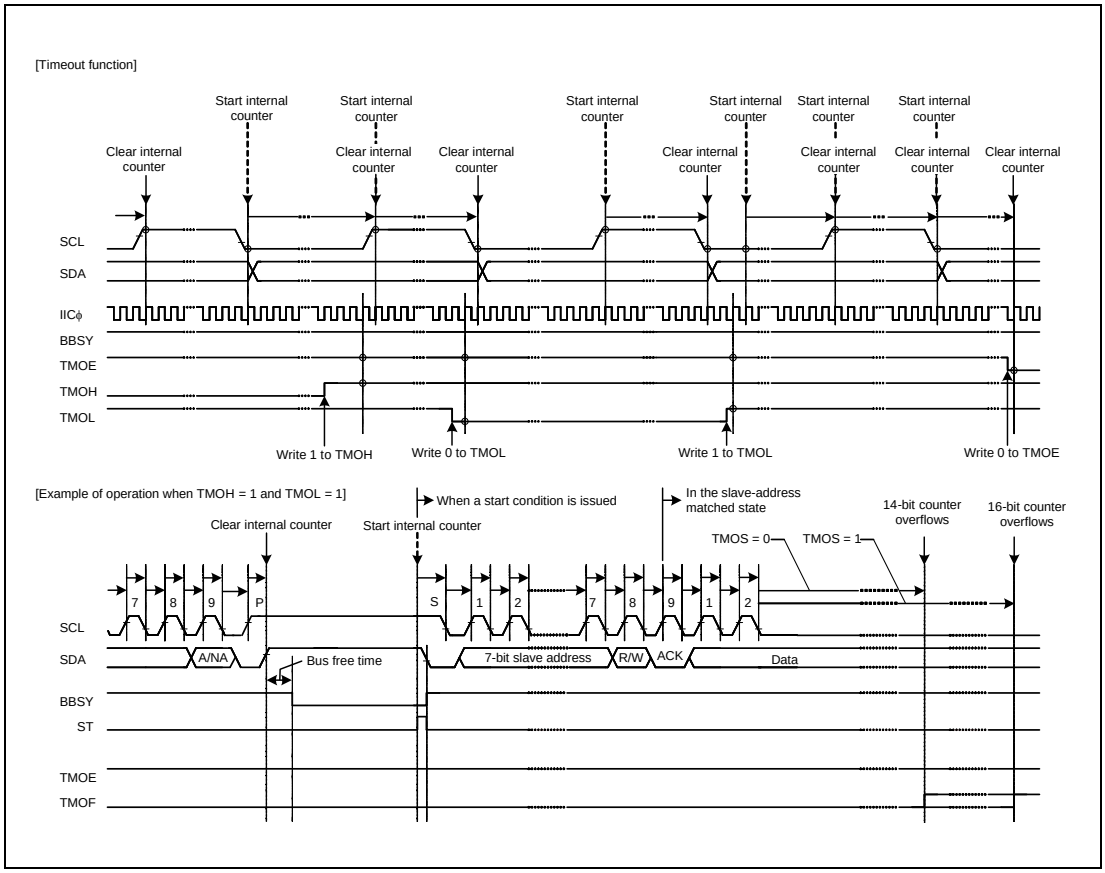


Figure 18.38 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)

18.13.2 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.

[Output conditions for using the RIICnCR1.CLO bit]

- When the bus is free (RIICnCR2.BBSY flag = 0) or in master mode (RIICnCR2.MST bit = 1 and BBSY flag = 1)
- When the communication device does not hold the SCL line low

Figure 18.39 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

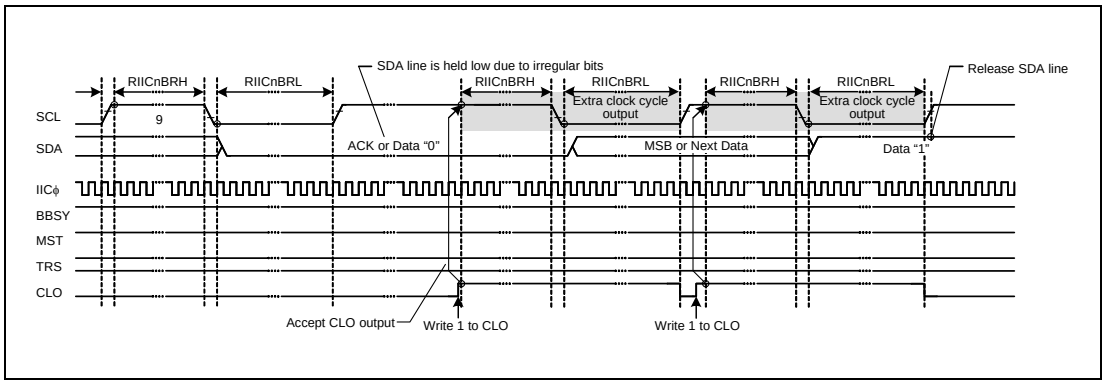


Figure 18.39 Extra SCL Clock Cycle Output Function (CLO Bit)

18.13.3 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 18.14, Reset Function of RIIC**.

18.14 Reset Function of RIIC

The RIIC has RIIC reset, and internal reset functions. In addition RIIC is cleared by ISORES. **Table 18.26** lists the scope of each reset and reset conditions.

Table 18.26 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
	CKS2-0	Initialized	Initialized	Retained	Retained	Retained
	BCWP	Initialized	Initialized	Retained	Retained	Retained
	BC2-0	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
	NF1-0	Initialized	Initialized	Retained	Retained	Retained
RIICnFER		Initialized	Initialized	Retained	Retained	Retained
RIICnSER		Initialized	Initialized	Retained	Retained	Retained
RIICnIER		Initialized	Initialized	Retained	Retained	Retained
RIICnSR1	RIICnDID	Initialized	Initialized	Initialized	Retained	Initialized
	RIICnGCA	Initialized	Initialized	Initialized	Retained	Initialized
	RIICnAAS2	Initialized	Initialized	Initialized	Retained	Initialized
	RIICnAAS1	Initialized	Initialized	Initialized	Retained	Initialized
	RIICnAAS0	Initialized	Initialized	Initialized	Retained	Initialized
RIICnSR2	RIICnTDRE	Initialized	Initialized	Initialized	Retained	Initialized
	RIICnTEND	Initialized	Initialized	Initialized	Retained	Initialized
	RIICnRDRF	Initialized	Initialized	Initialized	Retained	Retained
	RIICnNACKF	Initialized	Initialized	Initialized	Retained	Retained
	RIICnSTOP	Initialized	Initialized	Initialized	Retained	Operation
	RIICnSTART	Initialized	Initialized	Initialized	Operation	Initialized
	RIICnAL	Initialized	Initialized	Initialized	Retained	Retained
	RIICnTMOF	Initialized	Initialized	Initialized	Retained	Retained
RIICnSAR0, 1, 2		Initialized	Initialized	Retained	Retained	Retained

Table 18.26 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. BBSY can only be cleared after IICRST falls while the bus is free following the detection of SCL=H or SDAD (after a STOP condition) and the subsequent application of an ICE=IICRST internal reset.

Section 19 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/F1L 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.

19.1 Features of RH850/F1L RS-CAN

19.1.1 Number of Units and Channels

This microcontroller has the following number of RS-CAN units.

Table 19.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1					
Name	RSCANn (n = 0)					

The individual products have the CAN Interface Channel listed below.

Table 19.2 Unit Configurations and Channels

Unit Name	Channel Name	RH850/F1L 48 pins (1 ch)	RH850/F1L 64 pins (3 ch)	RH850/F1L 80 pins (3 ch)	RH850/F1L 100 pins (3 ch/6 ch*2)	RH850/F1L 144 pins (4 ch/6 ch*1)	RH850/F1L 176 pins (6 ch)
RSCAN0	CAN0	√	√	√	√	√	√
	CAN1		√	√	√	√	√
	CAN2		√	√	√	√	√
	CAN3				*2	√	√
	CAN4				*2	*1	√
	CAN5				*2	*1	√

Note 1. Code Flash 1.5MB and 2MB embedded.

Note 2. Only F1L for Gateway is supported.

Table 19.3 Index (1/2)

Index	Meaning
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 are 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 in the RS-CAN units are identified by the index “x” (x = 0 to 7): for example, RSCAN0RFSTSx is the receive FIFO buffer status register in the RSCAN0 unit.

Table 19.3 Index (2/2)

Index	Meaning
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.
p	The individual transmit buffers are generically indicated by the index “p” (q = 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 letter “y” (y = 0 to 2); for example, RSCAN0RMNDy is a receive buffer new data register.

Note: The functions and descriptions of registers in this section are for the RS-CANs that has 6 channels (m = 0 to 5). When referring to information with indexes, regard the index values as the ones corresponding to your target product. Also, note that, if the value of an index exceeds the range described in this section due to your target product, write the value after reset when writing to bits outside the index range.

The following table lists the values of indexes for individual products.

Table 19.4 Indexes for Individual Products

Indexes for Individual Products						
48 pins	64 pins, 80 pins	100 pins	100 pins (F1L for Gateway Only)	144 pins (Except for 1.5 MB, 2 MB embedded)	144 pins (1.5 MB, 2 MB embedded)	176 pins
j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15	j = 0 to 15
k = 0 to 2	k = 0 to 8	k = 0 to 8	k = 0 to 17	k = 0 to 11	k = 0 to 17	k = 0 to 17
x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7	x = 0 to 7
q = 0 to 15	q = 0 to 47	q = 0 to 47	q = 0 to 95	q = 0 to 63	q = 0 to 95	q = 0 to 95
p = 0 to 15	p = 0 to 47	p = 0 to 47	p = 0 to 95	p = 0 to 63	p = 0 to 95	p = 0 to 95
r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63	r = 0 to 63
y = 0	y = 0, 1	y = 0, 1	y = 0 to 2	y = 0, 1	y = 0 to 2	y = 0 to 2

19.1.2 Register Base Address

RSCAN0 base addresses are listed in the following table.

RSCAN0 register addresses are given as offsets from the base addresses in general.

Table 19.5 Register Base Address

Base Address Name	Base Address
<RSCAN0_base>	FFD0 0000_H

19.1.3 Clock Supply

The RSCAN0 clock supply is shown in the following table.

Table 19.6 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
RSCAN0	clk_xincan	CKSCLK_ICANOSC
	clkc	PPLLCLK2
	pclk	CKSCLK_ICAN

The operating frequency of the RSCAN0 depends on the transfer rate and the number of channels in use. **Table 19.7** shows the range of the frequency.

Table 19.7 Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1L

Condition		Range of Operating Frequency		
Transfer Rate	No. of Channels in Use	pclk	clk_xincan*1	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. Setting the DCS bit in RSCAN0GCFG enables to select either clk_xincan or clkc. Set clocks less than or equal to pclk/2.

Note 2. Select clk_xincan when pclk < 25 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 10.4.3.10, RS-CAN Clock Domains C_ISO_CAN and C_ISO_CANOSC.

19.1.4 Interrupt Request

RSCAN0 interrupt requests are listed in the following table.

Table 19.8 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
RSCAN0			
INTRCANGERR	CAN global error interrupt	14	—
INTRCANGRECC	CAN receive FIFO interrupt	15	—
CAN0			
INTRCANmERR (m = 0)	CAN0 error interrupt	16	—
INTRCANmREC (m = 0)	CAN0 transmit/receive FIFO receive completion interrupt	17	—
INTRCANmTRX (m = 0)	CAN0 transmit interrupt	18	—
CAN1			
INTRCANmERR (m = 1)	CAN1 error interrupt	105	—
INTRCANmREC (m = 1)	CAN1 transmit/receive FIFO receive completion interrupt	106	—
INTRCANmTRX (m = 1)	CAN1 transmit interrupt	107	—
CAN2			
INTRCANmERR (m = 2)	CAN2 error interrupt	209	—
INTRCANmREC (m = 2)	CAN2 transmit/receive FIFO receive completion interrupt	210	—
INTRCANmTRX (m = 2)	CAN2 transmit interrupt	211	—
CAN3			
INTRCANmERR (m = 3)	CAN3 error interrupt	212	—
INTRCANmREC (m = 3)	CAN3 transmit/receive FIFO receive completion interrupt	213	—
INTRCANmTRX (m = 3)	CAN3 transmit interrupt	214	—
CAN4			
INTRCANmERR (m = 4)	CAN4 error interrupt	264	—
INTRCANmREC (m = 4)	CAN4 transmit/receive FIFO receive completion interrupt	265	—
INTRCANmTRX (m = 4)	CAN4 transmit interrupt	266	—
CAN5			
INTRCANmERR (m = 5)	CAN5 error interrupt	279	—
INTRCANmREC (m = 5)	CAN5 transmit/receive FIFO receive completion interrupt	280	—
INTRCANmTRX (m = 5)	CAN5 transmit interrupt	281	—

NOTE

For the wake-up factors from standby mode, see **Section 11.1.2.1, Wake-Up Factors from Stand-By Mode**.

19.1.5 Reset Sources

RSCAN0 reset sources are listed in the following table. RSCAN0 is initialized by these reset sources.

Table 19.9 Reset Sources

Unit Name	Reset Source
RSCAN0	All reset sources (ISORES)

19.1.6 External Input/Output Signals

External input/output signals of RSCAN0 are listed below.

Table 19.10 External Input/Output Signals

Unit Signal Name	Outline	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		
CANnRX (m = 5)	CAN5 receive data input	CAN5RX
CANmTX (m = 5)	CAN5 transmit data output	CAN5TX

19.2 Overview

19.2.1 Functional Overview

The RH850/F1L incorporates one unit of the CAN interface (RS-CAN) which consists of six channels (CAN0 to CAN5) of the CAN controller conforming to the ISO11898-1 specifications. **Table 19.11** shows the RSCAN module specifications. **Figure 19.1** shows the RS-CAN module block diagram.

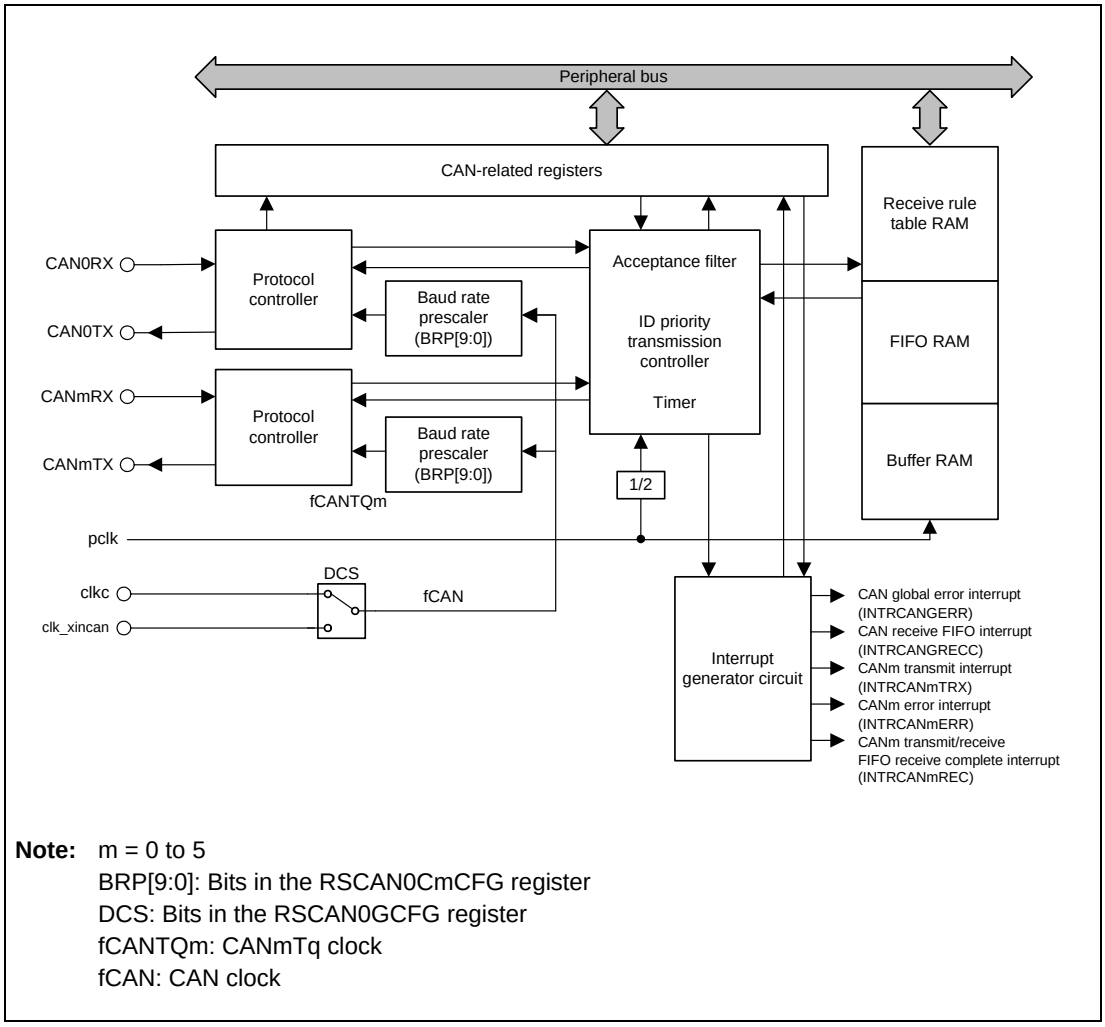
Table 19.11 RS-CAN Module Specifications (1/2)

Item	Specification
Number of channels	6
Protocol	ISO11898-1 compliant
Communication speed	• Maximum 1 Mbps Commnucation 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 • Individual buffers: 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 buffers: 384 buffers for all channels Receive buffer: 0 to 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 • Label addition 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. • Transmit request can be aborted (possible to confirm with a flag) • One-shot transmission function
Interval transmission function	Transmit messages at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)

Table 19.11 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 standby 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. As for the range of operating frequency, see **Table 19.7**.
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

19.2.2 Block Diagram



Note: 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 19.1 RS-CAN Module Block Diagram

19.3 Registers

19.3.1 List of Registers

RS-CAN registers are listed in the following table.

For details about <RSCAN0_base>, see **Section 19.1.2, Register Base Address**.

Table 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 register16	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	RSCAN0MDF019	<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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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> +<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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.12 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 19.13 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 19.14 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 19.15 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 19.16 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

19.3.2 RSCAN0CmCFG — Channel Configuration Register (m = 0 to 5)

Access: RSCAN0CmCFG register can be read/written in 32-bit units
RSCAN0CmCFGL, RSCAN0CmCFGH registers can be read/written in 16-bit units
RSCAN0CmCFGLL, RSCAN0CmCFGLH, RSCAN0CmCFGHL, RSCAN0CmCFGHH registers can be read/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 19.17 RSCAN0CmCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	These bits are read as the value after reset. The write value should be 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	These bits are read as the value after reset. The write value should be 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 19.17 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	These bits are read as the value after reset. The write value should be the value after reset.
9 to 0	BRP[9:0]	Prescaler Division Ratio Set 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 shifting to channel communication mode or channel wait mode. For a description of the bit timing parameters and settings, see **Section 19.10.1, Initial Settings**.

SJW[1:0] Bits

These bits are used to specify a Tq value for the resynchronization jump width. Allowed values are 1 Tq to 4 Tq, inclusive.
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).
Allowed values are 2 Tq to 8 Tq, inclusive.
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).
Allowed values are 4 Tq to 16 Tq, inclusive.

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).

19.3.3 RSCAN0CmCTR — Channel Control Register (m = 0 to 5)

Access: RSCAN0CmCTR register can be read/written in 32-bit units
RSCAN0CmCTRL, RSCAN0CmCTRH registers can be read/written in 16-bit units
RSCAN0CmCTRLL, RSCAN0CmCTRLH, RSCAN0CmCTRHL, RSCAN0CmCTRHH registers can be read/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 19.18 RSCAN0CmCTR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	These bits are read as the value after reset. The write value should be 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	These bits are read as the value after reset. The write value should be 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 19.18 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	These bits are read as the value after reset. The write value should be 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 these bits 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 clear 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 RSCAN 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 wait 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 19.5.2, 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.

19.3.4 RSCAN0CmSTS — Channel Status Register (m = 0 to 5)

Access: RSCAN0CmSTS register can be read only in 32-bit units
RSCAN0CmSTSL, RSCAN0CmSTSH registers can be read only in 16-bit units
RSCAN0CmSTSLL, RSCAN0CmSTSLH, RSCAN0CmSTSHL, RSCAN0CmSTSHH registers can be read only 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),
RSCAN0CmSTSLH: <RSCAN0_base> + 0009_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 19.19 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	These bits are read as the value after reset.
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.

19.3.5 RSCAN0CmERFL — Channel Error Flag Register (m = 0 to 5)

Access: RSCAN0CmERFL register can be read/written in 32-bit units
RSCAN0CmERFLL, RSCAN0CmERFLH registers can be read/written in 16-bit units
RSCAN0CmERFLLL, RSCAN0CmERFLLH, RSCAN0CmERFLHL, RSCAN0CmERFLHH registers can be read/written 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 19.20 RSCAN0CmERFL Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, 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 19.20 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 0 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.

19.3.6 RSCAN0GCFG — Global Configuration Register

Access: RSCAN0GCFG register can be read/written in 32-bit units
RSCAN0GCFGL, RSCAN0GCFGH registers can be read/written in 16-bit units
RSCAN0GCFGLL, RSCAN0GCFGLH, RSCAN0GCFGHL, RSCAN0GCFGHH registers can be read/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															
Value after 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 19.21 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 19.21 RSCAN0GCFG Register Contents (2/2)

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing to these bits, 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 19.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1L**.

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 19.7.3.1, 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 19.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1L**.

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.

19.3.7 RSCAN0GCTR — Global Control Register

Access: RSCAN0GCTR register can be read/written in 32-bit units
RSCAN0GCTRL, RSCAN0GCTRH registers can be read/written in 16-bit units
RSCAN0GCTRLL, RSCAN0GCTRLH, RSCAN0GCTRHL, RSCAN0GCTRHH registers can be read/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> + $008A_H$, RSCAN0GCTRHH: <RSCAN0_base> + $008B_H$

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 19.22 RSCAN0GCTR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, 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 to these bits, 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 RSCAN module into global stop mode.
Clearing this bit to 0 makes the RSCAN 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 19.5.1, Global Modes**. Setting the GSLPR bit to 1 when in global reset mode places the RS-CAN module into global stop mode.

19.3.8 RSCAN0GSTS — Global Status Register

Access: RSCAN0GSTS register can be read only in 32-bit units
RSCAN0GSTSL, RSCAN0GSTSH registers can be read only in 16-bit units
RSCAN0GSTSLL, RSCAN0GSTSLH, RSCAN0GSTSHL, RSCAN0GSTSHH registers can be read only in 8-bit units

Address: RSCAN0GSTS: <RSCAN0_base> + $008C_H$

RSCAN0GSTSL: <RSCAN0_base> + $008C_H$, RSCAN0GSTSH: <RSCAN0_base> + $008E_H$

RSCAN0GSTSLL: <RSCAN0_base> + $008C_H$, RSCAN0GSTSLH: <RSCAN0_base> + $008D_H$,
RSCAN0GSTSHL: <RSCAN0_base> + $008E_H$, RSCAN0GSTSHH: <RSCAN0_base> + $008F_H$

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 19.23 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 initialisation 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.

19.3.9 RSCAN0GERFL — Global Error Flag Register

Access: RSCAN0GERFL register can be read/written in 32-bit units
RSCAN0GERFLL, RSCAN0GERFLH registers can be read/written in 16-bit units
RSCAN0GERFLLL, RSCAN0GERFLLH, RSCAN0GERFLHL, RSCAN0GERFLHH registers can be read/written in 8-bit units

Address: RSCAN0GERFL: <RSCAN0_base> + 0090_H

RSCAN0GERFLL: <RSCAN0_base> + 0090_H, RSCAN0GERFLH: <RSCAN0_base> + 0092_H

RSCAN0GERFLLL: <RSCAN0_base> + 0090_H, RSCAN0GERFLLH: <RSCAN0_base> + 0091_H,
RSCAN0GERFLHL: <RSCAN0_base> + 0092_H, RSCAN0GERFLHH: <RSCAN0_base> + 0093_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 19.24 RSCAN0GERFL Register Contents

Bit Position	Bit Name	Function
31 to 14, 7, 6, 3	Reserved	When read, the value after reset is returned. When writing these bits, write the value after reset.
13 to 8, 5, 4	Reserved	When read, the value after reset is returned. When writing these bits, 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.

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”.

19.3.10 RSCAN0GTINTSTS0 — Global TX Interrupt Status Register 0

Access: RSCAN0GTINTSTS0 register can be read only in 32-bit units
RSCAN0GTINTSTS0L, RSCAN0GTINTSTS0H registers can be read only in 16-bit units
RSCAN0GTINTSTS0LL, RSCAN0GTINTSTS0LH, RSCAN0GTINTSTS0HL, RSCAN0GTINTSTS0HH registers can be read only 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 19.25 RSCAN0GTINTSTS0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
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	These bits are read as the value after reset. When writing to these bits, write the value after reset.
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.

Table 19.25 RSCAN0GTINTSTS0 Register Contents (2/2)

Bit Position	Bit Name	Function
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.
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	These bits are read as the value after reset.
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	These bits are read as the value after reset.
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 TMIE bit in the RSCAN0TMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RCAN0TMSTSp 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 RCAN0TMSTSp 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.

19.3.11 RSCAN0GTINTSTS1 — Global TX Interrupt Status Register 1

Access: RSCAN0GTINTSTS1 register can be read only in 32-bit units
RSCAN0GTINTSTS1L, RSCAN0GTINTSTS1H registers can be read only in 16-bit units
RSCAN0GTINTSTS1LL, RSCAN0GTINTSTS1LH, RSCAN0GTINTSTS1HL, RSCAN0GTINTSTS1HH registers can be read only in 8-bit units

Address: RSCAN0GTINTSTS1: <RSCAN0_base> + 0464_H

RSCAN0GTINTSTS1L: <RSCAN0_base> + 0464_H, RSCAN0GTINTSTS1H: <RSCAN0_base> + 0466_H

RSCAN0GTINTSTS1LL: <RSCAN0_base> + 0464_H, RSCAN0GTINTSTS1LH: <RSCAN0_base> + 0465_H,
RSCAN0GTINTSTS1HL: <RSCAN0_base> + 0466_H, RSCAN0GTINTSTS1HH: <RSCAN0_base> + 0467_H

Value after reset: $0000\ 0000_H$ This register is initialized by any reset.

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	R	R	R	R	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 19.26 RSCAN0GTINTSTS1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 13	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
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 FIFIO Transmit Interrupt Status Flag 0: Transmit/receive FIFIO transmit interrupt is not requested. 1: Transmit/receive FIFIO 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	These bits are read as the value after reset. When writing to these bits, write the value after reset.
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 FIFIO Transmit Interrupt Status Flag 0: Transmit/receive FIFIO transmit interrupt is not requested. 1: Transmit/receive FIFIO 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.

Table 19.26 RSCAN0GTINTSTS1 Register Contents (2/2)

Bit Position	Bit Name	Function
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.
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 TMIE bit in the RSCAN0TMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RCAN0TMSTSp 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.

19.3.12 RSCAN0GTSC — Global Timestamp Counter Register

Access: RSCAN0GTSC register can be read only in 32-bit units.
RSCAN0GTSCL, RSCAN0GTSCH registers can be read only in 16-bit units.

Address: RSCAN0GTSC: <RSCAN0_base> + 0094_H

RSCAN0GTSCL: <RSCAN0_base> + 0094_H, RSCAN0GTSCH: <RSCAN0_base> + 0096_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 19.27 RSCAN0GTSC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset.
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] 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 RSCAN module has transitioned to global operating mode.
  This counter stops counting when the RSCAN 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.

19.3.13 RSCAN0GAFLECTR — Receive Rule Entry Control Register

Access: RSCAN0GAFLECTR register can be read/written in 32-bit units
RSCAN0GAFLECTRL, RSCAN0GAFLECTRH registers can be read/written in 16-bit units
RSCAN0GAFLECTRLL, RSCAN0GAFLECTRLH, RSCAN0GAFLECTRHL, RSCAN0GAFLECTRHH registers can be read/written in 8-bit units

Address: RSCAN0GAFLECTR: <RSCAN0_base> + 0098_H

RSCAN0GAFLECTRL: <RSCAN0_base> + 0098_H, RSCAN0GAFLECTRH: <RSCAN0_base> + $009A_H$

RSCAN0GAFLECTRLL: <RSCAN0_base> + 0098_H, RSCAN0GAFLECTRLH: <RSCAN0_base> + 0099_H,
RSCAN0GAFLECTRHL: <RSCAN0_base> + $009A_H$, RSCAN0GAFLECTRHH: <RSCAN0_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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	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 19.28 RSCAN0GAFLECTR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are read as the value after reset. The write value should be 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	These bits are read as the value after reset. The write value should be 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 (10011_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.

19.3.14 RSCAN0GAFLCFG0 — Receive Rule Configuration Register 0

Access: RSCAN0GAFLCFG0 register can be read/written in 32-bit units
RSCAN0GAFLCFG0L, RSCAN0GAFLCFG0H registers can be read/written in 16-bit units
RSCAN0GAFLCFG0LL, RSCAN0GAFLCFG0LH, RSCAN0GAFLCFG0HL, RSCAN0GAFLCFG0HH registers can be read/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 19.29 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.

19.3.15 RSCAN0GAFLCFG1 — Receive Rule Configuration Register 1

Access: RSCAN0GAFLCFG1 register can be read/written in 32-bit units
RSCAN0GAFLCFG1L, RSCAN0GAFLCFG1H registers can be read/written in 16-bit units
RSCAN0GAFLCFG1LL, RSCAN0GAFLCFG1LH, RSCAN0GAFLCFG1HL, RSCAN0GAFLCFG1HH registers can be read/written in 8-bit units

Address: RSCAN0GAFLCFG1: <RSCAN0_base> + $00A0_H$

RSCAN0GAFLCFG1L: <RSCAN0_base> + $00A0_H$, RSCAN0GAFLCFG1H: <RSCAN0_base> + $00A2_H$

RSCAN0GAFLCFG1LL: <RSCAN0_base> + $00A0_H$, RSCAN0GAFLCFG1LH: <RSCAN0_base> + $00A1_H$,
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 19.30 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	These bits are read as the value after reset. The write value should be 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.

19.3.16 RSCAN0GAFLIDj — Receive Rule ID Register (j = 0 to 15)

Access: RSCAN0GAFLIDj register can be read/written in 32-bit units
RSCAN0GAFLIDjL, RSCAN0GAFLIDjH registers can be read/written in 16-bit units
RSCAN0GAFLIDjLL, RSCAN0GAFLIDjLH, RSCAN0GAFLIDjHL, RSCAN0GAFLIDjHH registers can be read/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 19.31 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.

19.3.17 RSCAN0GAFLMj — Receive Rule Mask Register (j = 0 to 15)

Access: RSCAN0GAFLMj register can be read/written in 32-bit units
RSCAN0GAFLMjL, RSCAN0GAFLMjH registers can be read/written in 16-bit units
RSCAN0GAFLMjLL, RSCAN0GAFLMjLH, RSCAN0GAFLMjHL, RSCAN0GAFLMjHH registers can be read/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 19.32 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	These bits are read as the value after reset. The write value should be 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.

19.3.18 RSCAN0GAFLP0j — Receive Rule Pointer 0 Register (j = 0 to 15)

Access: RSCAN0GAFLP0j register can be read/written in 32-bit units
RSCAN0GAFLP0jL, RSCAN0GAFLP0jH registers can be read/written in 16-bit units
RSCAN0GAFLP0jLL, RSCAN0GAFLP0jLH, RSCAN0GAFLP0jHL, RSCAN0GAFLP0jHH registers can be read/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)

RSCAN0GAFLP0jLL: <RSCAN0_base> + 0508_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
	GAFLRVM	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 19.33 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	These bits are read as the value after reset. The write value should be 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.

19.3.19 RSCAN0GAFLP1j — Receive Rule Pointer 1 Register (j = 0 to 15)

Access: RSCAN0GAFLP1j register can be read/written in 32-bit units
RSCAN0GAFLP1jL, RSCAN0GAFLP1jH registers can be read/written in 16-bit units
RSCAN0GAFLP1jLL, RSCAN0GAFLP1jLH, RSCAN0GAFLP1jHL, RSCAN0GAFLP1jHH registers can be read/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 19.34 RSCAN0GAFLP1j Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	These bits are read as the value after reset. The write value should be 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.

19.3.20 RSCAN0RMNB — Receive Buffer Number Register

Access: RSCAN0RMNB register can be read/written in 32-bit units
RSCAN0RMNBL, RSCAN0RMNBH registers can be read/written in 16-bit units
RSCAN0RMNBLL, RSCAN0RMNBLH, RSCAN0RMNBHL, RSCAN0RMNBHH registers can be read/written in 8-bit units

Address: RSCAN0RMNB: <RSCAN0_base> + $00A4_H$

RSCAN0RMNBL: <RSCAN0_base> + $00A4_H$, RSCAN0RMNBH: <RSCAN0_base> + $00A6_H$

RSCAN0RMNBLL: <RSCAN0_base> + $00A4_H$, RSCAN0RMNBLH: <RSCAN0_base> + $00A5_H$,
RSCAN0RMNBHL: <RSCAN0_base> + $00A6_H$, RSCAN0RMNBHH: <RSCAN0_base> + $00A7_H$

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 19.35 RSCAN0RMNB Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be 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.

19.3.21 RSCAN0RMNDy — Receive Buffer New Data Register (y = 0 to 2)

Access: RSCAN0RMNDy register can be read/written in 32-bit units
RSCAN0RMNDyL, RSCAN0RMNDyH registers can be read/written in 16-bit units
RSCAN0RMNDyLL, RSCAN0RMNDyLH, RSCAN0RMNDyHL, RSCAN0RMNDyHH registers can be read/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 19.36 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.

19.3.22 RSCAN0RMIDq — Receive Buffer ID Register (q = 0 to 95)

Access: RSCAN0RMIDq register can be read only in 32-bit units
RSCAN0RMIDqL, RSCAN0RMIDqH registers can be read only in 16-bit units
RSCAN0RMIDqLL, RSCAN0RMIDqLH, RSCAN0RMIDqHL, RSCAN0RMIDqHH registers can be read only 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 19.37 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	These bits are read as the value after reset.
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.

19.3.23 RSCAN0RMPTRq — Receive Buffer Pointer Register (q = 0 to 95)

Access: RSCAN0RMPTRq register can be read only in 32-bit units
RSCAN0RMPTRqL, RSCAN0RMPTRqH registers can be read only in 16-bit units
RSCAN0RMPTRqLL, RSCAN0RMPTRqLH, RSCAN0RMPTRqHL, RSCAN0RMPTRqHH registers can be read only 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 19.38 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.

19.3.24 RSCAN0RMDF0q — Receive Buffer Data Field 0 Register (q = 0 to 95)

Access: RSCAN0RMDF0q register can be read only in 32-bit units
RSCAN0RMDF0qL, RSCAN0RMDF0qH registers can be read only in 16-bit units
RSCAN0RMDF0qLL, RSCAN0RMDF0qLH, RSCAN0RMDF0qHL, RSCAN0RMDF0qHH registers can be read only 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 19.39 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] value in the RSCAN0RMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

19.3.25 RSCAN0RMDF1q — Receive Buffer Data Field 1 Register (q = 0 to 95)

Access: RSCAN0RMDF1q register can be read only in 32-bit units
RSCAN0RMDF1qL, RSCAN0RMDF1qH register can be read only in 16-bit units
RSCAN0RMDF1qLL, RSCAN0RMDF1qLH, RSCAN0RMDF1qHL, RSCAN0RMDF1qHH registers can be read only 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 19.40 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] value in the RSCAN0RMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

19.3.26 RSCAN0RFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7)

Access: RSCAN0RFCCx register can be read/written in 32-bit units
RSCAN0RFCCxL, RSCAN0RFCCxH registers can be read/written in 16-bit units
RSCAN0RFCCxLL, RSCAN0RFCCxLH, RSCAN0RFCCxHL, RSCAN0RFCCxHH registers can be read/written in 8-bit units

Address: RSCAN0RFCCx: <RSCAN0_base> + $00B8_H$ + (04_H × x)

RSCAN0RFCCxL: <RSCAN0_base> + $00B8_H$ + (04_H × x),
RSCAN0RFCCxH: <RSCAN0_base> + $00BA_H$ + (04_H × x)

RSCAN0RFCCxLL: <RSCAN0_base> + $00B8_H$ + (04_H × x),
RSCAN0RFCCxLH: <RSCAN0_base> + $00B9_H$ + (04_H × x),
RSCAN0RFCCxHL: <RSCAN0_base> + $00BA_H$ + (04_H × x),
RSCAN0RFCCxHH: <RSCAN0_base> + $00BB_H$ + (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 19.41 RSCAN0RFCCx Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. The write value should be 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	These bits are read as the value after reset. The write value should be 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	These bits are read as the value after reset. The write value should be the value after reset.

Table 19.41 RSCAN0RFCCx Register Contents (2/2)

Bit Position	Bit Name	Function
1	RFIE	Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled.
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.

19.3.27 RSCAN0RFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7)

Access: RSCAN0RFSTSx register can be read/written in 32-bit units
RSCAN0RFSTSxL, RSCAN0RFSTSxH registers can be read/written in 16-bit units
RSCAN0RFSTSxLL, RSCAN0RFSTSxLH, RSCAN0RFSTSxHL, RSCAN0RFSTSxHH registers can be read/written in 8-bit units

Address: RSCAN0RFSTSx: <RSCAN0_base> + $00D8_H$ + (04_H × x)

RSCAN0RFSTSxL: <RSCAN0_base> + $00D8_H$ + (04_H × x),
RSCAN0RFSTSxH: <RSCAN0_base> + $00DA_H$ + (04_H × x)

RSCAN0RFSTSxLL: <RSCAN0_base> + $00D8_H$ + (04_H × x),
RSCAN0RFSTSxLH: <RSCAN0_base> + $00D9_H$ + (04_H × x),
RSCAN0RFSTSxHL: <RSCAN0_base> + $00DA_H$ + (04_H × x),
RSCAN0RFSTSxHH: <RSCAN0_base> + $00DB_H$ + (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 19.42 RSCAN0RFSTSx Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. When writing to these bits, 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	These bits are read as the value after reset. When writing to these bits, 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

This flag indicates the number of unread messages in the receive FIFO buffer. This flag becomes 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.

19.3.28 RSCAN0RFPCTRx — Receive FIFO Buffer Pointer Control Register (x = 0 to 7)

Access: RSCAN0RFPCTRx register can only be written in 32-bit units
RSCAN0RFPCTRxL, RSCAN0RFPCTRxH registers can only be written in 16-bit units
RSCAN0RFPCTRxLL, RSCAN0RFPCTRxLH, RSCAN0RFPCTRxHL, RSCAN0RFPCTRxHH registers can only be written in 8-bit units

Address: RSCAN0RFPCTRx: <RSCAN0_base> + $00F8_H$ + (04_H × x)

RSCAN0RFPCTRxL: <RSCAN0_base> + $00F8_H$ + (04_H × x),
RSCAN0RFPCTRxH: <RSCAN0_base> + $00FA_H$ + (04_H × x)

RSCAN0RFPCTRxLL: <RSCAN0_base> + $00F8_H$ + (04_H × x),
RSCAN0RFPCTRxLH: <RSCAN0_base> + $00F9_H$ + (04_H × x),
RSCAN0RFPCTRxHL: <RSCAN0_base> + $00FA_H$ + (04_H × x),
RSCAN0RFPCTRxHH: <RSCAN0_base> + $00FB_H$ + (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 19.43 RSCAN0RFPCTRx Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be 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 RSCAN0RFID, RSCAN0RFPTR, RSCAN0RFDF0, and RSCAN0RFDF1 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).

19.3.29 RSCAN0RFIDx — Receive FIFO Buffer Access ID Register (x = 0 to 7)

Access: RSCAN0RFIDx register can be read only in 32-bit units
RSCAN0RFIDxL, RSCAN0RFIDxH registers can be read only in 16-bit units
RSCAN0RFIDxLL, RSCAN0RFIDxLH, RSCAN0RFIDxHL, RSCAN0RFIDxHH registers can be read only 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 19.44 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	These bits are read as the value after reset.
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.

19.3.30 RSCAN0RFPTRx — Receive FIFO Buffer Access Pointer Register (x = 0 to 7)

Access: RSCAN0RFPTRx register can be read only in 32-bit units
RSCAN0RFPTRxL, RSCAN0RFPTRxH registers can be read only in 16-bit units
RSCAN0RFPTRxLL, RSCAN0RFPTRxLH, RSCAN0RFPTRxHL, RSCAN0RFPTRxHH registers can be read only 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 19.45 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.

19.3.31 RSCAN0RFDF0x — Receive FIFO Buffer Access Data Field 0 Register (x = 0 to 7)

Access: RSCAN0RFDF0x register can be read-only in 32-bit units
RSCAN0RFDF0xL, RSCAN0RFDF0xH registers can be read only in 16-bit units
RSCAN0RFDF0xLL, RSCAN0RFDF0xLH, RSCAN0RFDF0xHL, RSCAN0RFDF0xHH registers can be read only 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 19.46 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] value in the RSCAN0RFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

19.3.32 RSCAN0RFDF1x — Receive FIFO Buffer Access Data Field 1 Register (x = 0 to 7)

Access: RSCAN0RFDF1x register can be read only in 32-bit units
RSCAN0RFDF1xL, RSCAN0RFDF1xH registers can be read only in 16-bit units
RSCAN0RFDF1xLL, RSCAN0RFDF1xLH, RSCAN0RFDF1xHL, RSCAN0RFDF1xHH registers can be read only 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 19.47 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] value in the RSCAN0RFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

19.3.33 RSCAN0CFCCk — Transmit/receive FIFO Buffer Configuration and Control Register k (k = 0 to 17)

Access: RSCAN0CFCCk register can be read/written in 32-bit units
RSCAN0CFCCkL, RSCAN0CFCCkH registers can be read/written in 16-bit units
RSCAN0CFCCkLL, RSCAN0CFCCkLH, RSCAN0CFCCkHL, RSCAN0CFCCkHH registers can be read/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 19.48 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 by (ITRCP [15:0] bits) 1: Clock dividing pclk 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 19.48 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	These bits are read as the value after reset. The write value should be 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	These bits are read as the value after reset. The write value should be 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 n of FIFO buffer k is calculated as n = k/3 (integer division). The actual assigned transmit buffer number p linked to FIFO buffer k will be ((16 × n) + CFTML[3:0]).

See **Table 19.14** and **Table 19.15**, 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 FIFIO 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

19.3.34 RSCAN0CFSTSk — Transmit/receive FIFO Buffer Status Register (k = 0 to 17)

Access: RSCAN0CFSTSk register can be read/written in 32-bit units
RSCAN0CFSTSkL, RSCAN0CFSTSkH registers can be read/written in 16-bit units
RSCAN0CFSTSkLL, RSCAN0CFSTSkLH, RSCAN0CFSTSkHL, RSCAN0CFSTSkHH registers can be read/written in 8-bit units

Address: RSCAN0CFSTSk: <RSCAN0_base> + 0178_H + (04_H × k)

RSCAN0CFSTSkL: <RSCAN0_base> + 0178_H + (04_H × k),
RSCAN0CFSTSkH: <RSCAN0_base> + $017A_H$ + (04_H × k)

RSCAN0CFSTSkLL: <RSCAN0_base> + 0178_H + (04_H × k),
RSCAN0CFSTSkLH: <RSCAN0_base> + 0179_H + (04_H × k),
RSCAN0CFSTSkHL: <RSCAN0_base> + $017A_H$ + (04_H × k),
RSCAN0CFSTSkHH: <RSCAN0_base> + $017B_H$ + (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 19.49 RSCAN0CFSTSk Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. When writing to these bits, 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	These bits are read as the value after reset. When writing to these bits, 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] value is 01_B (transmit mode): Number of untransmitted messages in the buffer
- When CFM[1:0] value is 00_B (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] value is 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] value is 00_B: In global reset mode
- When CFM[1:0] value is 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 CFMLT 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.

19.3.35 RSCAN0CFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register (k = 0 to 17)

Access: RSCAN0CFPCTRk register can only be written in 32-bit units
RSCAN0CFPCTRkL, RSCAN0CFPCTRkH registers can only be written in 16-bit units
RSCAN0CFPCTRkLL, RSCAN0CFPCTRkLH, RSCAN0CFPCTRkHL, RSCAN0CFPCTRkHH registers can only be written in 8-bit units

Address: RSCAN0CFPCTRk: <RSCAN0_base> + $01D8_H$ + (04_H × k)

RSCAN0CFPCTRkL: <RSCAN0_base> + $01D8_H$ + (04_H × k),
RSCAN0CFPCTRkH: <RSCAN0_base> + $01DA_H$ + (04_H × k)

RSCAN0CFPCTRkLL: <RSCAN0_base> + $01D8_H$ + (04_H × k),
RSCAN0CFPCTRkLH: <RSCAN0_base> + $01D9_H$ + (04_H × k),
RSCAN0CFPCTRkHL: <RSCAN0_base> + $01DA_H$ + (04_H × k),
RSCAN0CFPCTRkHH: <RSCAN0_base> + $01DB_H$ + (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 19.50 RSCAN0CFPCTRk Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be 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] value in the RSCAN0CFCCk register is 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] 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] value in the RSCAN0CFCCk register is 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] 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] value in the RSCAN0CFCCk register is 10_B):
  Setting prohibited

19.3.36 RSCAN0CFIDk — Transmit/receive FIFO Buffer Access ID Register (k = 0 to 17)

Access: RSCAN0CFIDk register can be read/written in 32-bit units
RSCAN0CFIDkL, RSCAN0CFIDkH registers can be read/written in 16-bit units
RSCAN0CFIDkLL, RSCAN0CFIDkLH, RSCAN0CFIDkHL, RSCAN0CFIDkHH registers can be read/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 19.51 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] value is 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] value is 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] value is 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] value in the RSCAN0CFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This RSCAN0CFIDk register should not be read or written when the CFM[1:0] value is 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] value is 00_B. When the CFM[1:0] value is 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] value is 00_B. When the CFM[1:0] value is 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] value is 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] value is 00_B.

When the CFM[1:0] value is 01_B, this bit is used to set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

19.3.37 RSCAN0CFPTRk — Transmit/receive FIFO Buffer Access Pointer Register (k = 0 to 17)

Access: RSCAN0CFPTRk register can be read/written in 32-bit units
RSCAN0CFPTRkL, RSCAN0CFPTRkH registers can be read/written in 16-bit units
RSCAN0CFPTRkLL, RSCAN0CFPTRkLH, RSCAN0CFPTRkHL, RSCAN0CFPTRkHH registers can be read/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 19.52 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] value is 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] value is 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] value is 00_B (receive mode). The timestamp value of the received message can be read.

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 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] value is 00_B. When the CFM[1:0] value is 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] value is 00_B. When the CFM[1:0] value is 01_B, the CFPTR[7:0] 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] value is 00_B.

19.3.38 RSCAN0CFDF0k — Transmit/receive FIFO Buffer Access Data Field 0 Register (k = 0 to 17)

Access: RSCAN0CFDF0k register can be read/written in 32-bit units
RSCAN0CFDF0kL, RSCAN0CFDF0kH registers can be read/written in 16-bit units
RSCAN0CFDF0kLL, RSCAN0CFDF0kLH, RSCAN0CFDF0kHL, RSCAN0CFDF0kHH registers can be read/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 19.53 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] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.
23 to 16	CFDB2[7:0]	
15 to8	CFDB1[7:0]	
7 to 0	CFDB0[7:0]	

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01_B (transmit mode).

This register is readable only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] 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] value is 10_B (gateway mode).

19.3.39 RSCAN0CFDF1k — Transmit/receive FIFO Buffer Access Data Field 1 Register (k = 0 to 17)

Access: RSCAN0CFDF1k register can be read/written in 32-bit units
RSCAN0CFDF1kL, RSCAN0CFDF1kH registers can be read/written in 16-bit units
RSCAN0CFDF1kLL, RSCAN0CFDF1kLH, RSCAN0CFDF1kHL, RSCAN0CFDF1kHH registers can be read/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 19.54 RSCAN0CFDF1k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB7[7:0]	Transmit/Receive FIFO Buffer Data Byte 7
23 to 16	CFDB6[7:0]	Transmit/Receive FIFO Buffer Data Byte 6
15 to 8	CFDB5[7:0]	Transmit/Receive FIFO Buffer Data Byte 5
7 to 0	CFDB4[7:0]	Transmit/Receive FIFO Buffer Data Byte 4 • When CFM[1:0] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01_B (transmit mode).

This register is readable only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] 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] value is 10_B (gateway mode).

19.3.40 RSCAN0FESTS — FIFO Empty Status Register

Access: RSCAN0FESTSregisters can be read only in 32-bit units
RSCAN0FESTSL, RSCAN0FESTSH registers can be read only in 16-bit units
RSCAN0FESTSLL, RSCAN0FESTSLH, RSCAN0FESTSHL, RSCAN0FESTSHH registers can be read only 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 19.55 RSCAN0FESTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	These bits are read as the value after reset.
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 19.55 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.

19.3.41 RSCAN0FFSTS — FIFO Full Status Register

Access: RSCAN0FFSTS register can be read only in 32-bit units
RSCAN0FFSTSL, RSCAN0FFSTSH registers can be read only in 16-bit units
RSCAN0FFSTSLL, RSCAN0FFSTSLH, RSCAN0FFSTSHL, RSCAN0FFSTSHH registers can be read only 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 19.56 RSCAN0FFSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	These bits are read as the value after reset.
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 19.56 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.

19.3.42 RSCAN0FMSTS — FIFO Message Lost Status Register

Access: RSCAN0FMSTS register can be read only in 32-bit units
RSCAN0FMSTSL, RSCAN0FMSTSH registers can be read only in 16-bit units
RSCAN0FMSTSLL, RSCAN0FMSTSLH, RSCAN0FMSTSHL, RSCAN0FMSTSHH registers can be read only 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 19.57 RSCAN0FMSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	These bits are read as the value after reset.
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 19.57 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.

19.3.43 RSCAN0RFISTS — Receive FIFO Buffer Interrupt Flag Status Register

Access: RSCAN0RFISTS register can be read only in 32-bit units
RSCAN0RFISTSL, RSCAN0RFISTSH registers can be read only in 16-bit units
RSCAN0RFISTSLL, RSCAN0RFISTSLH, RSCAN0RFISTSHL, RSCAN0RFISTSHH registers can be read only in 8-bit units

Address: RSCAN0RFISTS: <RSCAN0_base> + 0244_H

RSCAN0RFISTSL: <RSCAN0_base> + 0244_H, RSCAN0RFISTSH: <RSCAN0_base> + 0246_H

RSCAN0RFISTSLL: <RSCAN0_base> + 0244_H, RSCAN0RFISTSLH: <RSCAN0_base> + 0245_H,
RSCAN0RFISTSHL: <RSCAN0_base> + 0246_H, RSCAN0RFISTSHH: <RSCAN0_base> + 0247_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 19.58 RSCAN0RFISTS Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset.
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.

19.3.44 RSCAN0CFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register

Access: RSCAN0CFRISTS register can be read only in 32-bit units
RSCAN0CFRISTSL, RSCAN0CFRISTSH registers can be read only in 16-bit units
RSCAN0CFRISTSLL, RSCAN0CFRISTSLH, RSCAN0CFRISTSHL, RSCAN0CFRISTSHH registers can be read only 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> + $024A_H$, RSCAN0CFRISTSHH: <RSCAN0_base> + $024B_H$

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 19.59 RSCAN0CFRISTS Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	These bits are read as the value after reset.
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.

19.3.45 RSCAN0CFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register

Access: RSCAN0CFTISTS register can be read only in 32-bit units
RSCAN0CFTISTSL, RSCAN0CFTISTSH registers can be read only in 16-bit units
RSCAN0CFTISTSLL, RSCAN0CFTISTSLH, RSCAN0CFTISTSHL, RSCAN0CFTISTSHH registers can be read only 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> + $024E_H$, RSCAN0CFTISTSHH: <RSCAN0_base> + $024F_H$

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 19.60 RSCAN0CFTISTS Register Contents

Bit Position	Bit Name	Function
31 to 18	Reserved	These bits are read as the value after reset.
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.

19.3.46 RSCAN0TMCp — Transmit Buffer Control Register (p = 0 to 95)

Access: RSCAN0TMCp register can be read/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 19.61 RSCAN0TMCp Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	These bits are read as the value after reset. The write value should be 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 loss 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] in the RSCAN0TMSTSp register is 00_B.

19.3.47 RSCAN0TMSTSp — Transmit Buffer Status Register (p = 0 to 95)

Access: RSCAN0TMSTSp register can be read/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 19.62 RSCAN0TMSTSp Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	These bits are read as the value after reset. The write value should be 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

This flag indicates 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] flag in channel communication mode or channel halt mode. Do not write any value other than 00_B to this flag.

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.

19.3.48 RSCAN0TMTRSTSy — Transmit Buffer Transmit Request Status Register (y = 0 to 2)

Access: RSCAN0TMTRSTSy register can be read only in 32-bit units
RSCAN0TMTRSTSyL, RSCAN0TMTRSTSyH registers can be read only in 16-bit units
RSCAN0TMTRSTSyLL, RSCAN0TMTRSTSyLH, RSCAN0TMTRSTSyHL, RSCAN0TMTRSTSyHH registers can be read only 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> + 0253_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 19.63 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 19.64 shows the bit assignment.

Table 19.64 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

19.3.49 RSCAN0TMTARSTSy — Transmit Buffer Transmit Abort Request Status Register (y = 0 to 2)

Access: RSCAN0TMTARSTSy register can be read only in 32-bit units
RSCAN0TMTARSTSyL, RSCAN0TMTARSTSyH registers can be read only in 16-bit units
RSCAN0TMTARSTSyLL, RSCAN0TMTARSTSyLH, RSCAN0TMTARSTSyHL, RSCAN0TMTARSTSyHH registers can be read only 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> + 0263_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 19.65 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 19.66 shows the bit assignment.

Table 19.66 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

19.3.50 RSCAN0TMTCSTSy — Transmit Buffer Transmit Complete Status Register (y = 0 to 2)

Access: RSCAN0TMTCSTSy register can be read only in 32-bit units
RSCAN0TMTCSTSyL, RSCAN0TMTCSTSyH registers can be read only in 16-bit units
RSCAN0TMTCSTSyLL, RSCAN0TMTCSTSyLH, RSCAN0TMTCSTSyHL, RSCAN0TMTCSTSyHH registers can be read only 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> + 0273_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 19.67 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] flag in the RSCAN0TMSTSp register is 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.

A TMTCSTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 19.68 shows the bit assignment.

Table 19.68 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

19.3.51 RSCAN0TMTASTSy — Transmit Buffer Transmit Abort Status Register (y = 0 to 2)

Access: RSCAN0TMTASTSy register can be read only in 32-bit units
RSCAN0TMTASTSyL, RSCAN0TMTASTSyH registers can be read only in 16-bit units
RSCAN0TMTASTSyLL, RSCAN0TMTASTSyLH, RSCAN0TMTASTSyHL, RSCAN0TMTASTSyHH registers can be read only 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 19.69 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] flag in the RSCAN0TMSTSp register is set to 01_B (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

A TMTASTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 19.70 shows the bit assignment.

Table 19.70 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

19.3.52 RSCAN0TMIECy — Transmit Buffer Interrupt Enable Configuration Register (y = 0 to 2)

Access: RSCAN0TMIECy register can be read/written in 32-bit units
RSCAN0TMIECyL, RSCAN0TMIECyH registers can be read/written in 16-bit units
RSCAN0TMIECyLL, RSCAN0TMIECyLH, RSCAN0TMIECyHL, RSCAN0TMIECyHH registers can be read/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 19.71 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 19.72 shows the bit assignment.

Table 19.72 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

19.3.53 RSCAN0TMIDp — Transmit Buffer ID Register (p = 0 to 95)

Access: RSCAN0TMIDp register can be read/written in 32-bit units
RSCAN0TMIDpL, RSCAN0TMIDpH registers can be read/written in 16-bit units
RSCAN0TMIDpLL, RSCAN0TMIDpLH, RSCAN0TMIDpHL, RSCAN0TMIDpHH registers can be read/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 19.73 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.

19.3.54 RSCAN0TMPTRp — Transmit Buffer Pointer Register (p= 0 to 95)

Access: RSCAN0TMPTRp register can be read/written in 32-bit units
RSCAN0TMPTRpL, RSCAN0TMPTRpH registers can be read/written in 16-bit units
RSCAN0TMPTRpLL, RSCAN0TMPTRpLH, RSCAN0TMPTRpHL, RSCAN0TMPTRpHH registers can be read/written in 8-bit units

Address: RSCAN0TMPTRp: <RSCAN0_base> + 1004_H + (10_H × p)

RSCAN0TMPTRpL: <RSCAN0_base> + 1004_H + (10_H × p),
RSCAN0TMPTRpH: <RSCAN0_base> + 1006_H + (10_H × p)

RSCAN0TMPTRpLL: <RSCAN0_base> + 1004_H + (10_H × p),
RSCAN0TMPTRpLH: <RSCAN0_base> + 1005_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 19.74 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 to these bits, 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 to these bits, 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] value is stored in the transmit history buffer.

19.3.55 RSCAN0TMDF0p — Transmit Buffer Data Field 0 Register (p = 0 to 95)

Access: RSCAN0TMDF0p register can be read/written in 32-bit units
RSCAN0TMDF0pL, RSCAN0TMDF0pH registers can be read/written in 16-bit units
RSCAN0TMDF0pLL, RSCAN0TMDF0pLH, RSCAN0TMDF0pHL, RSCAN0TMDF0pHH registers can be read/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 19.75 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.

19.3.56 RSCAN0TMDF1p — Transmit Buffer Data Field 1 Register (p = 0 to 95)

Access: RSCAN0TMDF1p register can be read/written in 32-bit units
RSCAN0TMDF1pL, RSCAN0TMDF1pH registers can be read/written in 16-bit units
RSCAN0TMDF1pLL, RSCAN0TMDF1pLH, RSCAN0TMDF1pHL, RSCAN0TMDF1pHH registers can be read/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 19.76 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.

19.3.57 RSCAN0TXQCCm — Transmit Queue Configuration and Control Register (m = 0 to 5)

Access: RSCAN0TXQCCm register can be read/written in 32-bit units
RSCAN0TXQCCmL, RSCAN0TXQCCmH registers can be read/written in 16-bit units
RSCAN0TXQCCmLL, RSCAN0TXQCCmLH, RSCAN0TXQCCmHL, RSCAN0TXQCCmHH registers can be read/written in 8-bit units

Address: RSCAN0TXQCCm: <RSCAN0_base> + $03A0_H$ + (04_H × m)

RSCAN0TXQCCmL: <RSCAN0_base> + $03A0_H$ + (04_H × m),
RSCAN0TXQCCmH: <RSCAN0_base> + $03A2_H$ + (04_H × m)

RSCAN0TXQCCmLL: <RSCAN0_base> + $03A0_H$ + (04_H × m),
RSCAN0TXQCCmLH: <RSCAN0_base> + $03A1_H$ + (04_H × m),
RSCAN0TXQCCmHL: <RSCAN0_base> + $03A2_H$ + (04_H × m),
RSCAN0TXQCCmHH: <RSCAN0_base> + $03A3_H$ + (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 19.77 RSCAN0TXQCCm Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, 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 19.9**. 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] bits to 0010_B or more.

19.3.58 RSCAN0TXQSTSm — Transmit Queue Status Register (m = 0 to 5)

Access: RSCAN0TXQSTSm register can be read/written in 32-bit units
RSCAN0TXQSTSmL, RSCAN0TXQSTSmH registers can be read/written in 16-bit units
RSCAN0TXQSTSmLL, RSCAN0TXQSTSmLH, RSCAN0TXQSTSmHL, RSCAN0TXQSTSmHH registers can be read/written in 8-bit units

Address: RSCAN0TXQSTSm: <RSCAN0_base> + $03C0_H$ + (04_H × m)

RSCAN0TXQSTSmL: <RSCAN0_base> + $03C0_H$ + (04_H × m),
RSCAN0TXQSTSmH: <RSCAN0_base> + $03C2_H$ + (04_H × m)

RSCAN0TXQSTSmLL: <RSCAN0_base> + $03C0_H$ + (04_H × m),
RSCAN0TXQSTSmLH: <RSCAN0_base> + $03C1_H$ + (04_H × m),
RSCAN0TXQSTSmHL: <RSCAN0_base> + $03C2_H$ + (04_H × m),
RSCAN0TXQSTSmHH: <RSCAN0_base> + $03C3_H$ + (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 19.78 RSCAN0TXQSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	Reserved	When read, the value after reset is returned. When writing to these bits, write “0”.
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, 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

19.3.59 RSCAN0TXQPCTRm — Transmit Queue Pointer Control Register (m = 0 to 5)

Access: RSCAN0TXQPCTRm registers can only be written in 32-bit units
RSCAN0TXQPCTRmL, RSCAN0TXQPCTRmH registers can only be written in 16-bit units
RSCAN0TXQPCTRmLL, RSCAN0TXQPCTRmLH, RSCAN0TXQPCTRmHL, RSCAN0TXQPCTRmHH registers can only be written in 8-bit units

Address: RSCAN0TXQPCTRm: <RSCAN0_base> + $03E0_H$ + (04_H × m)

RSCAN0TXQPCTRmL: <RSCAN0_base> + $03E0_H$ + (04_H × m),
RSCAN0TXQPCTRmH: <RSCAN0_base> + $03E2_H$ + (04_H × m)

RSCAN0TXQPCTRmLL: <RSCAN0_base> + $03E0_H$ + (04_H × m),
RSCAN0TXQPCTRmLH: <RSCAN0_base> + $03E1_H$ + (04_H × m),
RSCAN0TXQPCTRmHL: <RSCAN0_base> + $03E2_H$ + (04_H × m),
RSCAN0TXQPCTRmHH: <RSCAN0_base> + $03E3_H$ + (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 19.79 RSCAN0TXQPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be 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 (he transmit queue is not full).

19.3.60 RSCAN0THLCCm — Transmit History Configuration and Control Register (m = 0 to 5)

Access: RSCAN0THLCCm register can be read/written in 32-bit units
RSCAN0THLCCmL, RSCAN0THLCCmH registers can be read/written in 16-bit units
RSCAN0THLCCmLL, RSCAN0THLCCmLH, RSCAN0THLCCmHL, RSCAN0THLCCmHH registers can be read/written in 8-bit units

Address: RSCAN0THLCCm: <RSCAN0_base> + 0400_H + (04_H × m)

RSCAN0THLCCmL: <RSCAN0_base> + 0400_H + (04_H × m),
RSCAN0THLCCmH: <RSCAN0_base> + 0402_H + (04_H × m)

RSCAN0THLCCmLL: <RSCAN0_base> + 0400_H + (04_H × m),
RSCAN0THLCCmLH: <RSCAN0_base> + 0401_H + (04_H × m),
RSCAN0THLCCmHL: <RSCAN0_base> + 0402_H + (04_H × m),
RSCAN0THLCCmHH: <RSCAN0_base> + 0403_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 19.80 RSCAN0THLCCm Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, 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.

19.3.61 RSCAN0THLSTSm — Transmit History Status Register (m = 0 to 5)

Access: RSCAN0THLSTSm register can be read/written in 32-bit units
RSCAN0THLSTSmL, RSCAN0THLSTSmH register can be read/written in 16-bit units
RSCAN0THLSTSmLL, RSCAN0THLSTSmLH, RSCAN0THLSTSmHL, RSCAN0THLSTSmHH registers can be read/written in 8-bit units

Address: RSCAN0THLSTSm: <RSCAN0_base> + 0420_H + (04_H × m)

RSCAN0THLSTSmL: <RSCAN0_base> + 0420_H + (04_H × m),
RSCAN0THLSTSmH: <RSCAN0_base> + 0422_H + (04_H × m)

RSCAN0THLSTSmLL: <RSCAN0_base> + 0420_H + (04_H × m),
RSCAN0THLSTSmLH: <RSCAN0_base> + 0421_H + (04_H × m),
RSCAN0THLSTSmHL: <RSCAN0_base> + 0422_H + (04_H × m),
RSCAN0THLSTSmHH: <RSCAN0_base> + 0423_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 19.81 RSCAN0THLSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, 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.

19.3.62 RSCAN0THLACCm — Transmit History Access Register (m = 0 to 5)

Access: RSCAN0THLACCm register can be read only in 32-bit units
RSCAN0THLACCmL, RSCAN0THLACCmH registers can be read only in 16-bit units
RSCAN0THLACCmLL, RSCAN0THLACCmLH, RSCAN0THLACCmHL, RSCAN0THLACCmHH registers can be read only in 8-bit units

Address: RSCAN0THLACCm: <RSCAN0_base> + 1800_H + (04_H × m)

RSCAN0THLACCmL: <RSCAN0_base> + 1800_H + (04_H × m),
RSCAN0THLACCmH: <RSCAN0_base> + 1802_H + (04_H × m)

RSCAN0THLACCmLL: <RSCAN0_base> + 1800_H + (04_H × m),
RSCAN0THLACCmLH: <RSCAN0_base> + 1801_H + (04_H × m),
RSCAN0THLACCmHL: <RSCAN0_base> + 1802_H + (04_H × m),
RSCAN0THLACCmHH: <RSCAN0_base> + 1803_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 19.82 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.

19.3.63 RSCAN0THLPCTRm — Transmit History Pointer Control Register (m = 0 to 5)

Access: RSCAN0THLPCTRm register can only be written in 32-bit units
RSCAN0THLPCTRmL, RSCAN0THLPCTRmH registers can only be written in 16-bit units
RSCAN0THLPCTRmLL, RSCAN0THLPCTRmLH, RSCAN0THLPCTRmHL, RSCAN0THLPCTRmHH registers can only be written in 8-bit units

Address: RSCAN0THLPCTRm: <RSCAN0_base> + 0440_H + (04_H × m)

RSCAN0THLPCTRmL: <RSCAN0_base> + 0440_H + (04_H × m),
RSCAN0THLPCTRmH: <RSCAN0_base> + 0442_H + (04_H × m)

RSCAN0THLPCTRmLL: <RSCAN0_base> + 0440_H + (04_H × m),
RSCAN0THLPCTRmLH: <RSCAN0_base> + 0441_H + (04_H × m),
RSCAN0THLPCTRmHL: <RSCAN0_base> + 0442_H + (04_H × m),
RSCAN0THLPCTRmHH: <RSCAN0_base> + 0443_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 19.83 RSCAN0THLPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, 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.

19.3.64 RSCAN0GTSTCFG — Global Test Configuration Register

Access: RSCAN0GTSTCFG register can be read/written in 32-bit units
RSCAN0GTSTCFGL, RSCAN0GTSTCFGH registers can be read/written in 16-bit units
RSCAN0GTSTCFGLL, RSCAN0GTSTCFGLH, RSCAN0GTSTCFGHL, RSCAN0GTSTCFGHH registers can be read/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, RSCAN0GTSTCFGLH: <RSCAN0_base> + 0469_H,
RSCAN0GTSTCFGHL: <RSCAN0_base> + $046A_H$, RSCAN0GTSTCFGHH: <RSCAN0_base> + $046B_H$

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 19.84 RSCAN0GTSTCFG Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, 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.

19.3.65 RSCAN0GTSTCTR — Global Test Control Register

Access: RSCAN0GTSTCTR register can be read/written in 32-bit units
RSCAN0GTSTCTRL, RSCAN0GTSTCTRH registers can be read/written in 16-bit units
RSCAN0GTSTCTRLL, RSCAN0GTSTCTRLH, RSCAN0GTSTCTRHL, RSCAN0GTSTCTRHH registers can be read/written in 8-bit units

Address: RSCAN0GTSTCTR: <RSCAN0_base> + $046C_H$

RSCAN0GTSTCTRL: <RSCAN0_base> + $046C_H$, RSCAN0GTSTCTRH: <RSCAN0_base> + $046E_H$

RSCAN0GTSTCTRLL: <RSCAN0_base> + $046C_H$, RSCAN0GTSTCTRLH: <RSCAN0_base> + $046D_H$,
RSCAN0GTSTCTRHL: <RSCAN0_base> + $046E_H$, RSCAN0GTSTCTRHH: <RSCAN0_base> + $046F_H$

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 19.85 RSCAN0GTSTCTR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, 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 to these bits, 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.

19.3.66 RSCAN0GLOCKK — Global Lock Key Register

Access: RSCAN0GLOCKK register can be write only in 32-bit units.
RSCAN0GLOCKKL, RSCAN0GLOCKKH registers can be write only in 16-bit units.

Address: RSCAN0GLOCKK: <RSCAN0_base> + $047C_H$

RSCAN0GLOCKKL: <RSCAN0_base> + $047C_H$, RSCAN0GLOCKKH: <RSCAN0_base> + $047E_H$

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 19.86 RSCAN0GLOCKK Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing these bits, 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 19.10.4.2, 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 (<RCAN0_base> + 0000_H to <RCAN0_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.

19.3.67 RSCAN0RPGACCr — RAM Test Page Access Register (r = 0 to 63)

Access: RSCAN0RPGACCr register can be read/written in 32-bit units
RSCAN0RPGACCrL, RSCAN0RPGACCrH registers can be read/written in 16-bit units
RSCAN0RPGACCrLL, RSCAN0RPGACCrLH, RSCAN0RPGACCrHL, RSCAN0RPGACCrHH registers can be read/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 19.87 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.

19.4 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 transmit 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 19.88 lists the CAN interrupt sources. **Figure 19.2** shows the CAN global interrupt block diagram. **Figure 19.3** shows the CAN channel interrupt block diagram.

Table 19.88 List of CAN Interrupt Sources

Interrupt Source			Corresponding Interrupt Request Flag	Corresponding Interrupt Enable Bit
Global interrupts	Receive FIFO	Receive FIFO 0	RFIF in the RSCAN0RFSTS0 register	RFIE in the RSCAN0RFCC0 register
		Receive FIFO 1	RFIF in the RSCAN0RFSTS1 register	RFIE in the RSCAN0RFCC1 register
		Receive FIFO 2	RFIF in the RSCAN0RFSTS2 register	RFIE in the RSCAN0RFCC2 register
		Receive FIFO 3	RFIF in the RSCAN0RFSTS3 register	RFIE in the RSCAN0RFCC3 register
		Receive FIFO 4	RFIF in the RSCAN0RFSTS4 register	RFIE in the RSCAN0RFCC4 register
		Receive FIFO 5	RFIF in the RSCAN0RFSTS5 register	RFIE in the RSCAN0RFCC5 register
		Receive FIFO 6	RFIF in the RSCAN0RFSTS6 register	RFIE in the RSCAN0RFCC6 register
		Receive FIFO 7	RFIF in the RSCAN0RFSTS7 register	RFIE in the RSCAN0RFCC7 register
	Global error		• DEF in the RSCAN0GERFL register • MES in the RSCAN0GERFL register • THLES in the RSCAN0GERFL register	• DEIE in the RSCAN0GCTR register • MEIE in the RSCAN0GCTR register • THLEIE in the RSCAN0GCTR register
Channel interrupts (m = 0 to 5)	CANm transmit	CANm transmit complete	TMTRF[1:0] in the RSCAN0TMSTSp register	TMIE in the RSCAN0TMIECy register
		CANm transmit abort	TMTRF[1:0] in the RSCAN0TMSTSp register	TAIE in the RSCAN0CmCTR register
		CANm transmit/receive FIFO transmit complete	CFTXIF in the RSCAN0CFSTSk register	CFTXIE in the RSCAN0CFCCk register
		CANm transmit queue	TXQIF in the RSCAN0TXQSTSm register	TXQIE in the RSCAN0TXQCCm register
		CANm transmit history	THLIF in the RSCAN0THLSTSm register	THLIE in the RSCAN0THLCCm register
	CANm transmit/receive FIFO receive complete		CFRXIF in the RSCAN0CFSTSk register	CFRXIE in the RSCAN0CFCCk register
	CANm error		• BEF in the RSCAN0CmERFL register • ALF in the RSCAN0CmERFL register • BLF in the RSCAN0CmERFL register • OVLF in the RSCAN0CmERFL register • BORF in the RSCAN0CmERFL register • BOEF in the RSCAN0CmERFL register • EPF in the RSCAN0CmERFL register • EWF in the RSCAN0CmERFL register	• BEIE in the RSCAN0CmCTR register • ALIE in the RSCAN0CmCTR register • BLIE in the RSCAN0CmCTR register • OLIE in the RSCAN0CmCTR register • BORIE in the RSCAN0CmCTR register • BOEIE in the RSCAN0CmCTR register • EPIE in the RSCAN0CmCTR register • EWIE in the RSCAN0CmCTR register

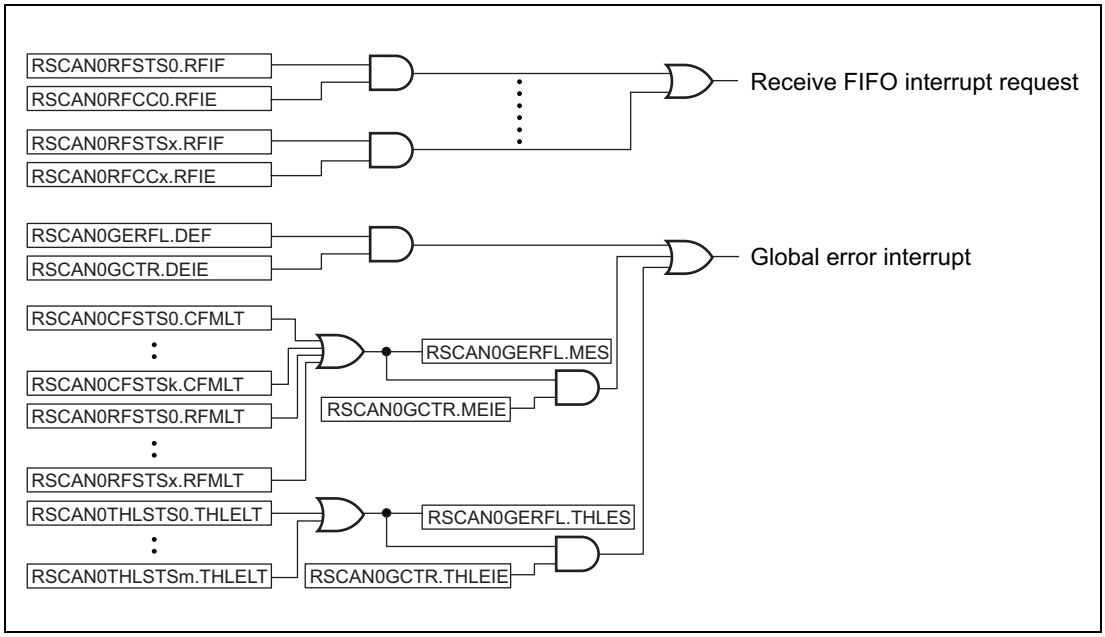


Figure 19.2 CAN Global Interrupt Block Diagram

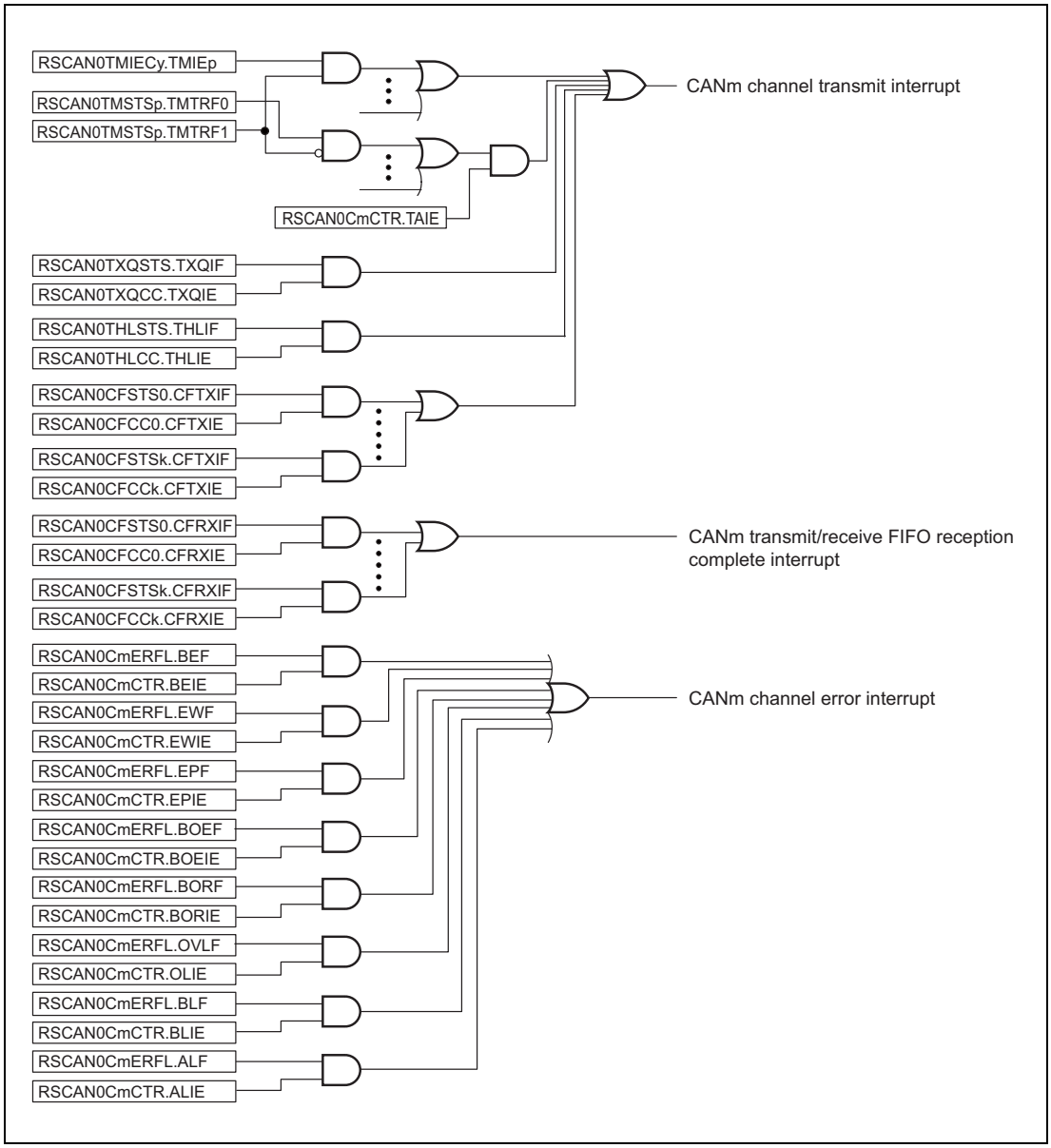


Figure 19.3 CAN Channel Interrupt Block Diagram

19.5 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 19.5.1, Global Modes**, and details of channel modes are described in **Section 19.5.2, 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.

19.5.1 Global Modes

Figure 19.4 shows the transitions of global modes.

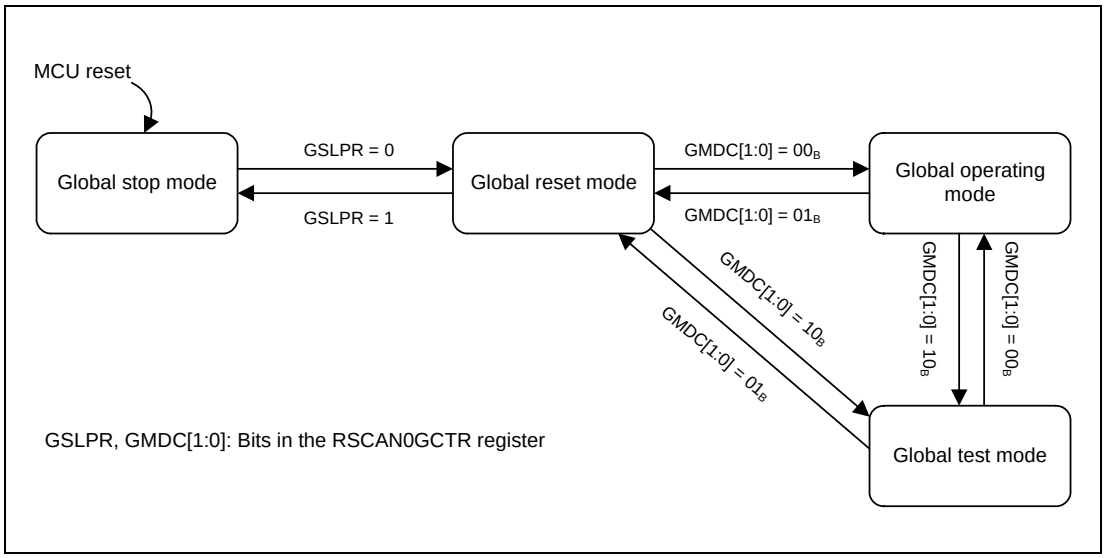


Figure 19.4 Transitions of Global Modes

In some cases, global mode transitions also force channel mode transitions. **Table 19.89** shows the channel mode transitions depending on the global mode setting dictated by the GMDC[1:0] bits and the GSLPR bit.

Table 19.89 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 19.90 shows the global mode transition time.

Table 19.90 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	Three pclk cycles
Global test	Global operating	Three pclk cycles
Global operating	Global reset	Three pclk cycles
Global operating	Global test	Two CAN frames*1

Note 1. CAN frame time of the lowest communication speed of the channels in use

19.5.1.1 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.

19.5.1.2 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 19.93** and **Table 19.94** 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).

19.5.1.3 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.

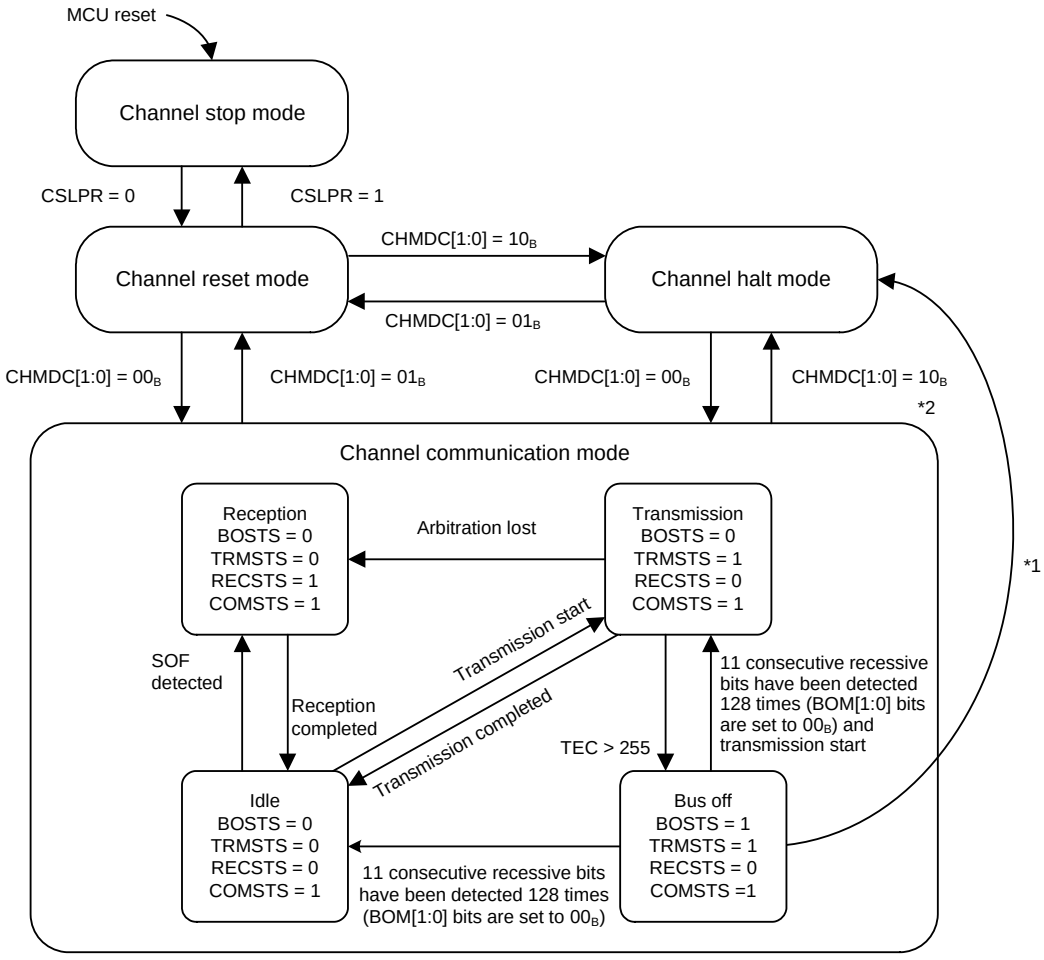
19.5.1.4 Global Operating Mode

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.

19.5.2 Channel Modes

Figure 19.5 shows a channel mode state transition chart. **Table 19.91** 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 19.5 Channel Mode State Transition Chart

Table 19.91 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	Two CANm bit times

Table 19.91 Channel Mode Transition Time (2/2)

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel halt	Channel reset	Three pclk cycles
Channel halt	Channel communication	Three CANm bit times
Channel communication	Channel reset	Three pclk cycles
Channel communication	Channel halt	Two CANm frames

19.5.2.1 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 GSLPR bit in the RSCAN0CmCTR register (m = 0 to 5) is set to 1 (channel stop mode) in channel reset mode. The GSLPR bit should not be modified in channel communication mode and channel halt mode.

19.5.2.2 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 19.93** 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 19.92** shows the operation when the CHMDC[1:0] bits are set to 01_B (channel reset mode) during CAN communication.

19.5.2.3 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 19.92 shows operation when the CHMDC[1:0] bits are set to 10_B (channel halt mode) during CAN communication.

Table 19.92 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] = 00_B] Transitions to channel halt mode (CHMDC[1:0] = 10_B) only after bus off recovery. [When BOM[1:0] = 01_B] Transitions to channel halt mode automatically when the condition for transition to bus off state is met. [When BOM[1:0] = 10_B] Transitions to channel halt mode automatically after bus off recovery. [When BOM[1:0] = 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 wait mode is to be made, set the RSCAN0CmCFG register in channel reset mode and then shift to channel wait mode.

19.5.2.4 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.

19.5.2.5 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] = 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] = 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] = 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] = 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] value is 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 19.93 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 19.94 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

19.6 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.

19.6.1 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 19.6** illustrates how receive rules are registered.

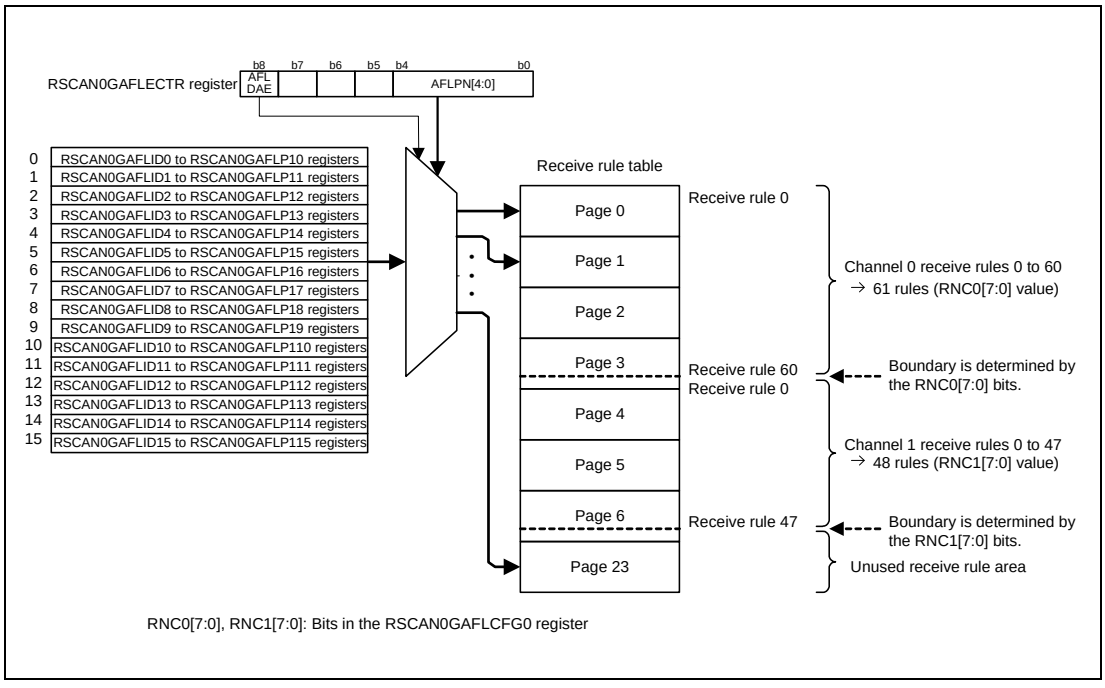


Figure 19.6 Entry of Receive Rules (for Setting Channel 0 and 1)

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.

19.6.1.1 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.

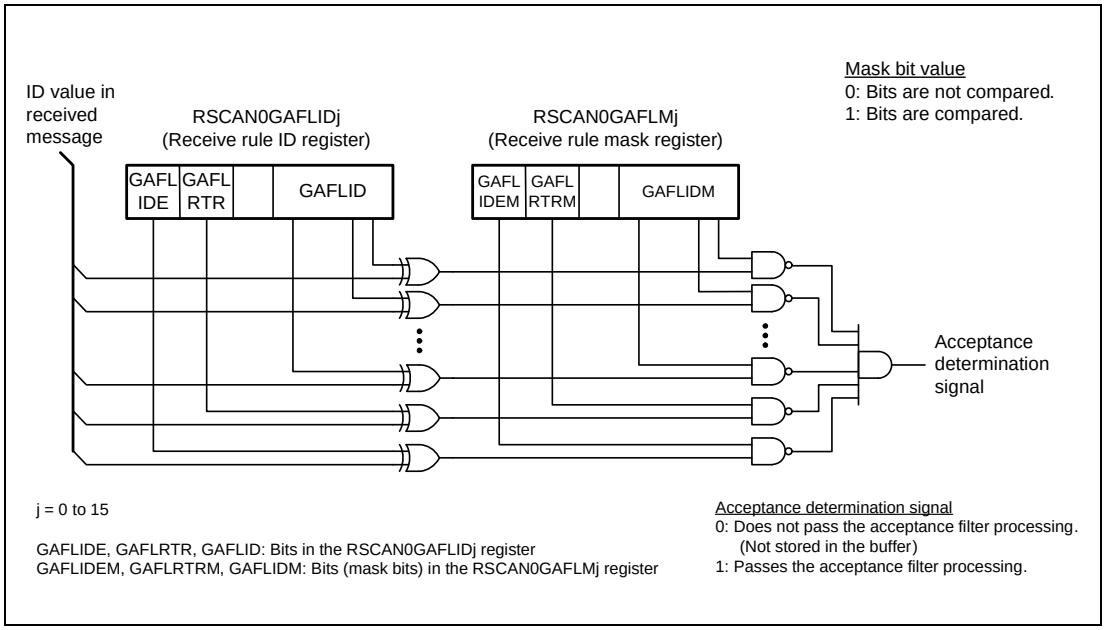


Figure 19.7 Acceptance Filter Function

19.6.1.2 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).

19.6.1.3 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.

19.6.1.4 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.

19.6.1.5 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.

19.6.1.6 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] value 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.

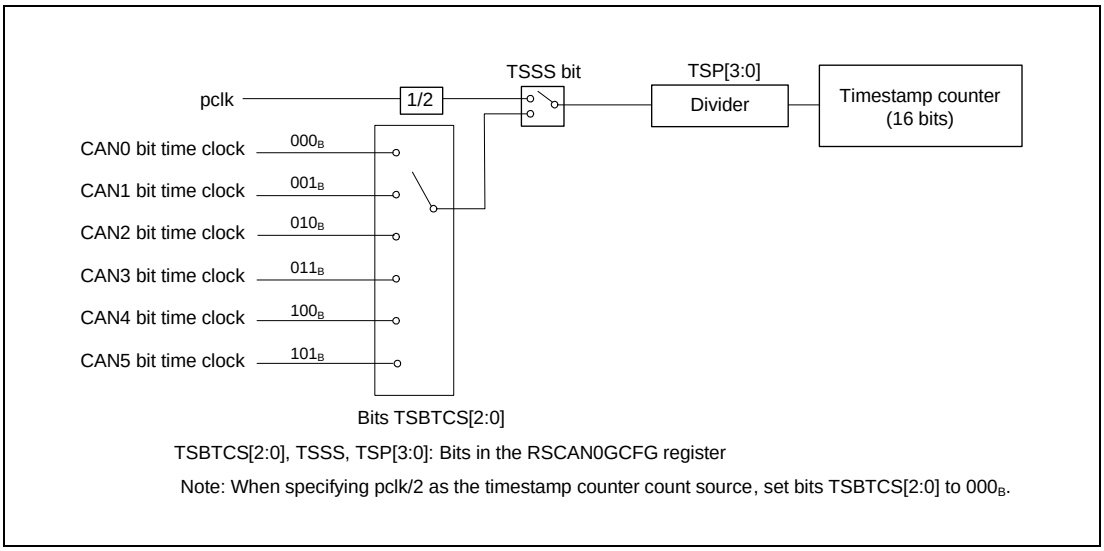


Figure 19.8 Timestamp Function Block Diagram

19.7 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 19.9 shows the allocation of transmit queues and transmit/receive FIFO buffer link.

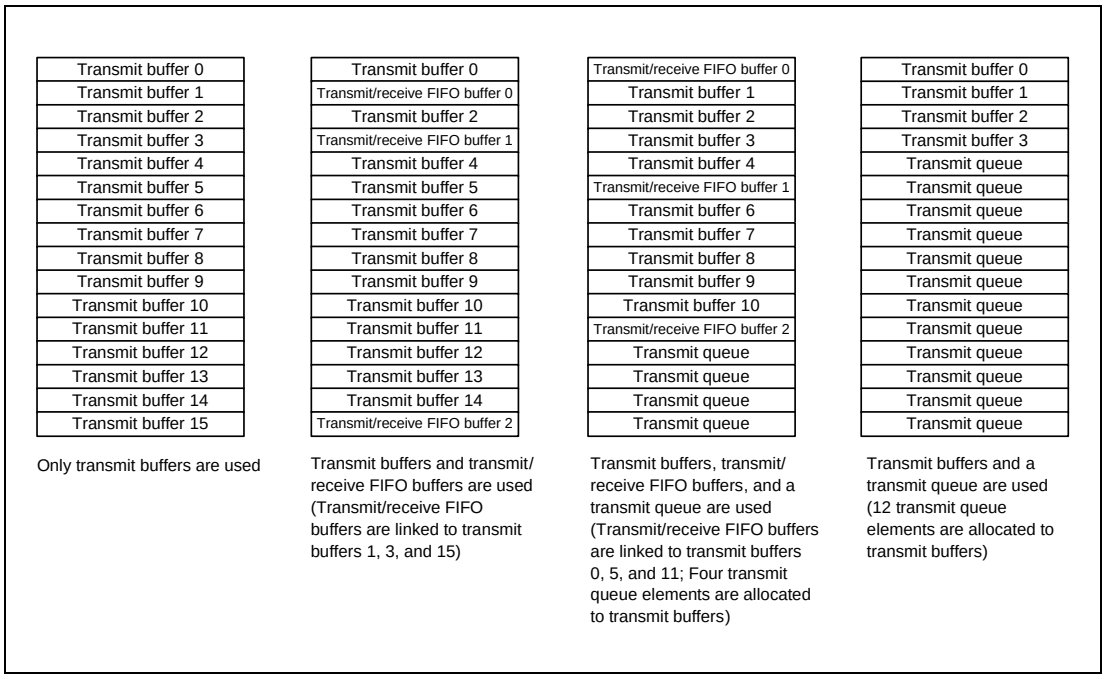


Figure 19.9 Allocation of Transmit Queues and Transmit/Receive FIFO Buffer Links

19.7.1 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 regardless of the TPRI bit.

19.7.2 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)).

19.7.2.1 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.

19.7.2.2 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).

19.7.3 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.

19.7.3.1 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 value of ITRCP[15:0] and N is the value of CFITT[7:0].

- When CFITR and CFITSS = 00_B (fPBA is the frequency of pclk):

$$\frac{1}{f_{PBA}} \times 2 \times M \times N$$

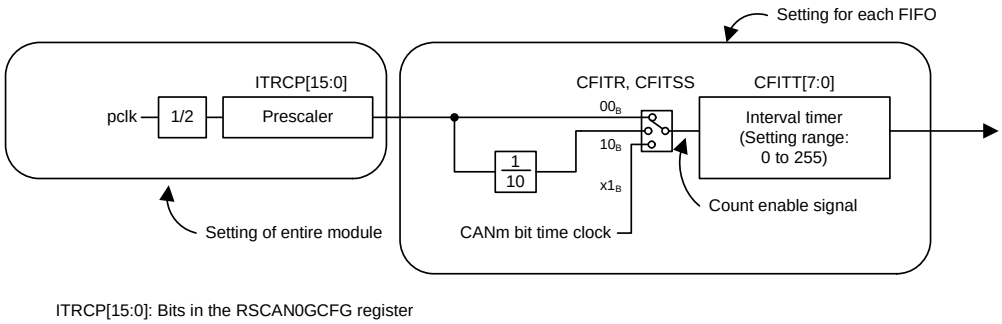
- When CFITR and CFITSS = 10_B:

$$\frac{1}{f_{PBA}} \times 2 \times M \times 10 \times N$$

- When CFITR and CFITSS = $x1_B$ (fCANBIT is the frequency of CANm bit time clock):

$$\frac{1}{f_{CANBIT}} \times N$$

Figure 19.10 shows the interval timer block diagram.



ITRCP[15:0]: Bits in the RSCAN0GCFG register
CFITR, CFITSS, CFITT[7:0] : Bits in the RSCAN0CFCCk register

Note: m = 0 to 5, k = 0 to 17

Figure 19.10 Interval Timer Block Diagram

Figure 19.11 shows the interval timer timing diagram.

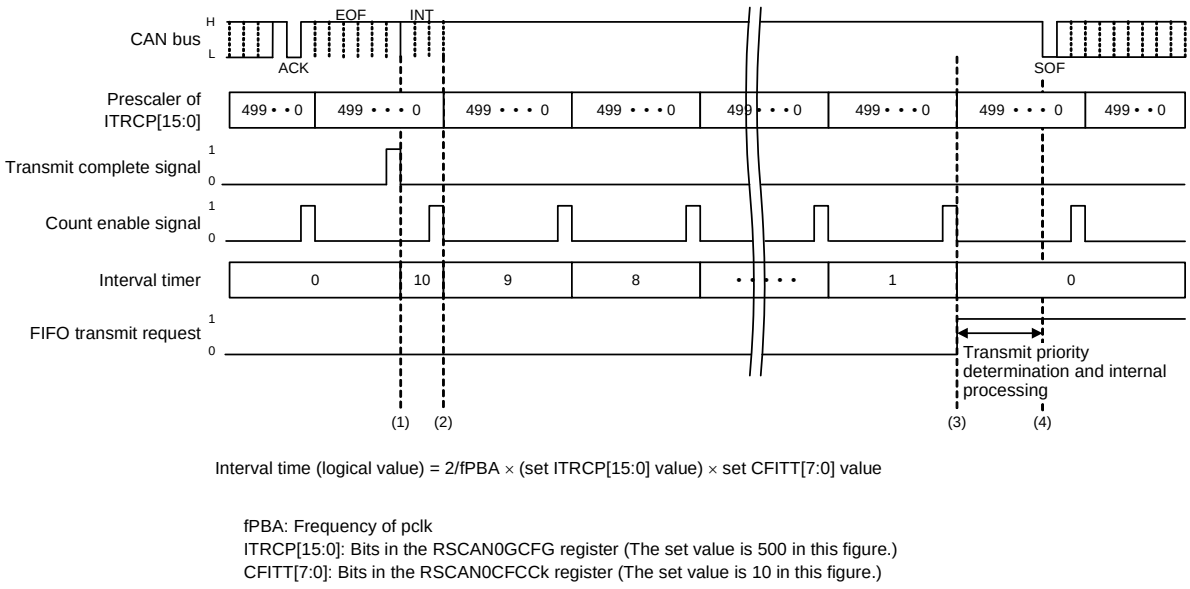


Figure 19.11 Interval Timer Timing Chart

(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.

19.7.4 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.

19.7.5 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 150 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 19.95**.
- Label data — Label information of the transmit message

Table 19.95 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.

19.8 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) for the transmit/receive FIFO buffer selected by the RSCAN0GAFLP1j register of a channel being used for transmission, 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.

19.9 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

19.9.1 Standard Test Mode

Standard test mode allows CRC test.

19.9.2 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 19.12 shows the connection when listen-only mode is selected.

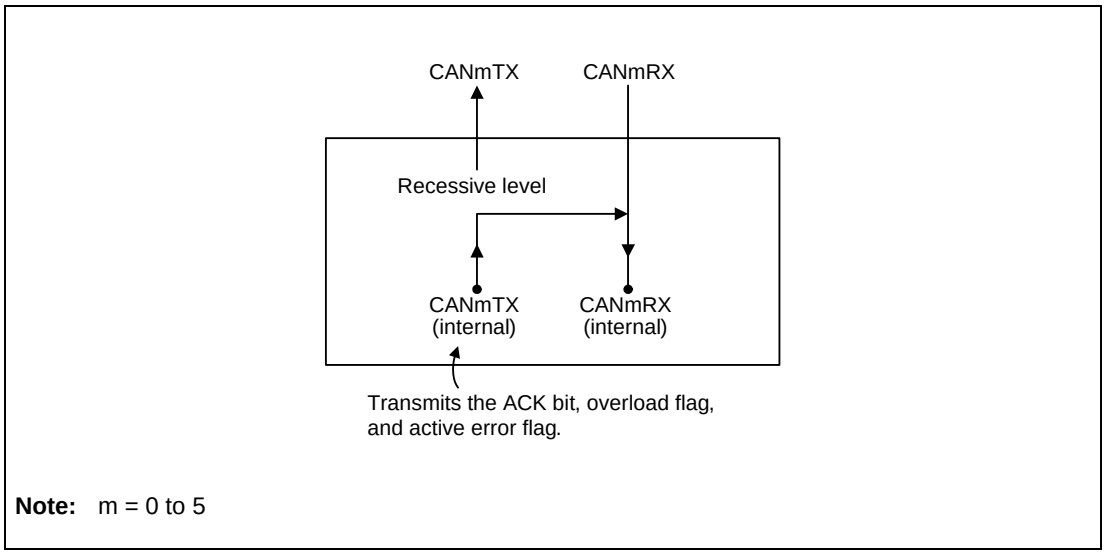


Figure 19.12 Connection when Listen-Only Mode is Selected

19.9.3 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).

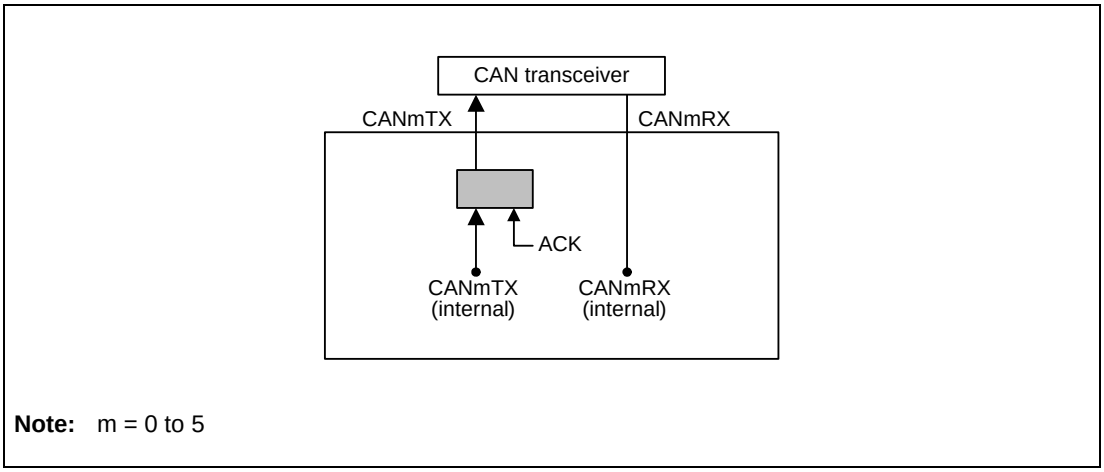
If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

19.9.3.1 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 19.13 shows the connection when self-test mode 0 is selected.



Note: m = 0 to 5

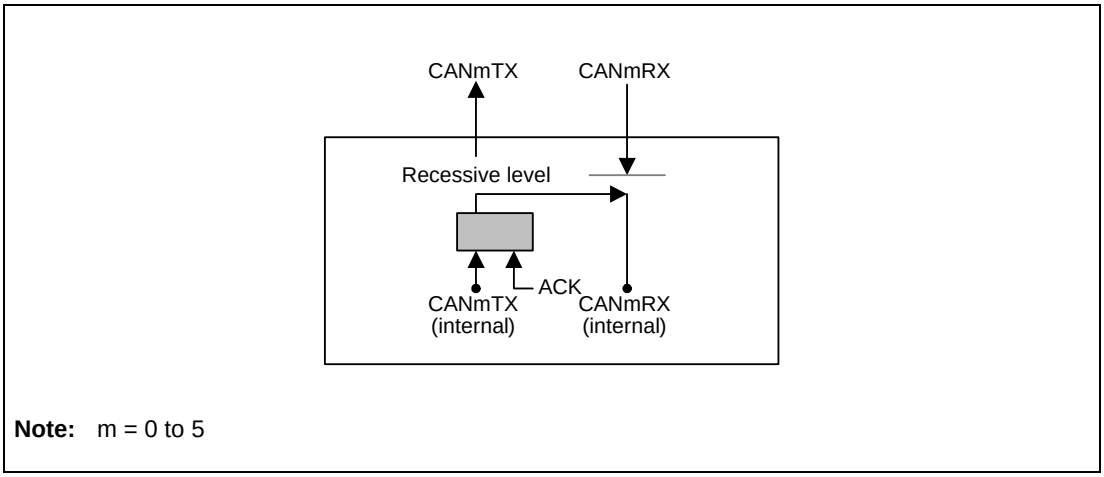
Figure 19.13 Connection when Self-Test Mode 0 is Selected

19.9.3.2 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 19.14 shows the connection when self-test mode 1 is selected.



Note: m = 0 to 5

Figure 19.14 Connection when Self-Test Mode 1 is Selected

19.9.4 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 14592 bytes (3900_H).

19.9.5 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 19.15 shows the connection for inter-channel communication test.

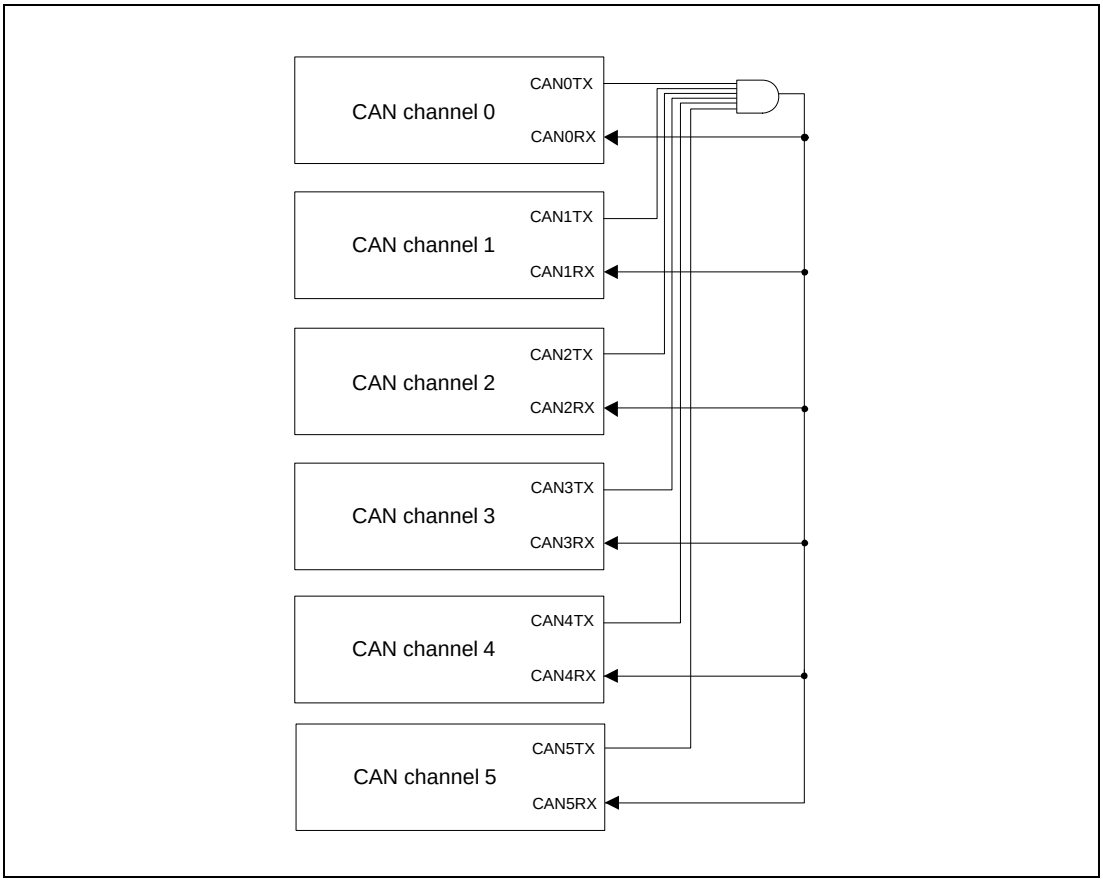
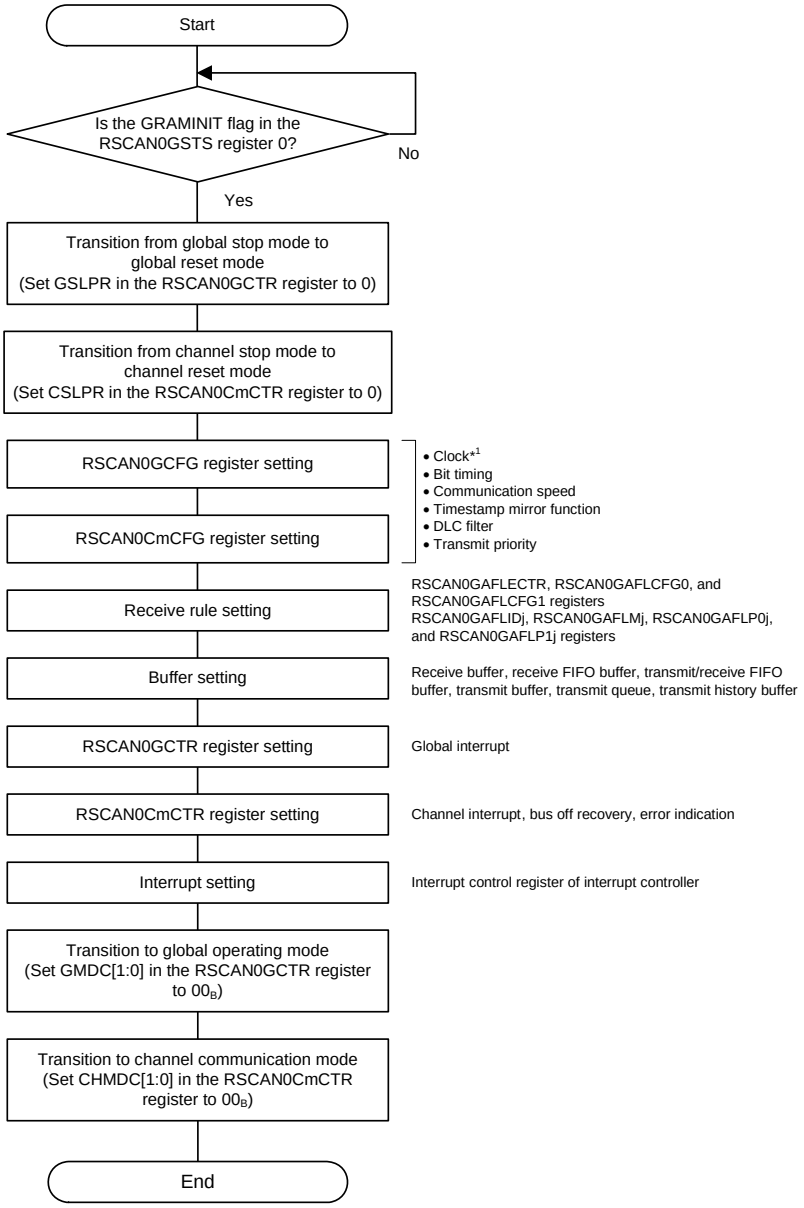


Figure 19.15 Connection for Inter-Channel Communication Test

19.10 RS-CAN Setting Procedure

19.10.1 Initial Settings

The RS-CAN module initializes the CAN RAM after the MCU is reset. The RAM initialization time is 7298 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 19.16** shows the CAN setting procedure after the MCU is reset.



Note 1. For the setting of CAN clock frequency, see the **Table 19.7, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/F1L**.

Note: m = 0 to 5, j = 0 to 15

Figure 19.16 CAN Setting Procedure after the MCU is Reset

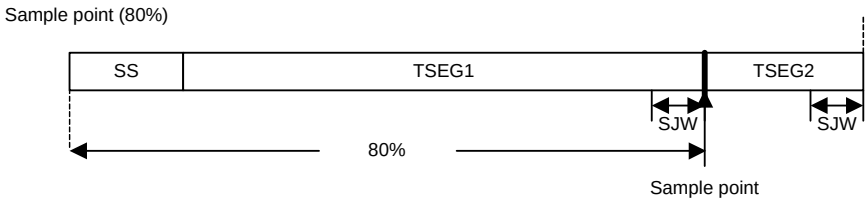
19.10.1.1 Clock Setting

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.

19.10.1.2 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 19.17 shows the bit timing chart. **Table 19.96** 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 19.17 Bit Timing Chart

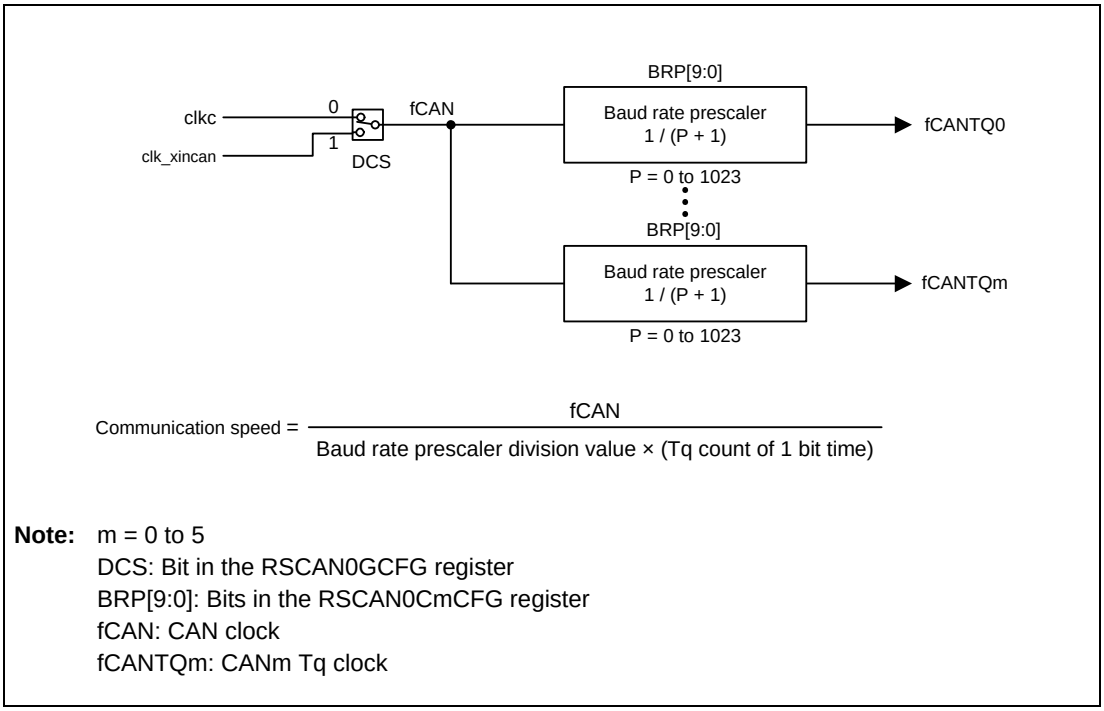
Table 19.96 Example of Bit Timing Setting

1 Bit	Set Value (Tq)				Sample Point (%) Note: See Figure 19.17.
	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

19.10.1.3 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 19.18 shows the CAN clock control block diagram, and **Table 19.97** 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 19.18 CAN Clock Control Block Diagram

Table 19.97 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)

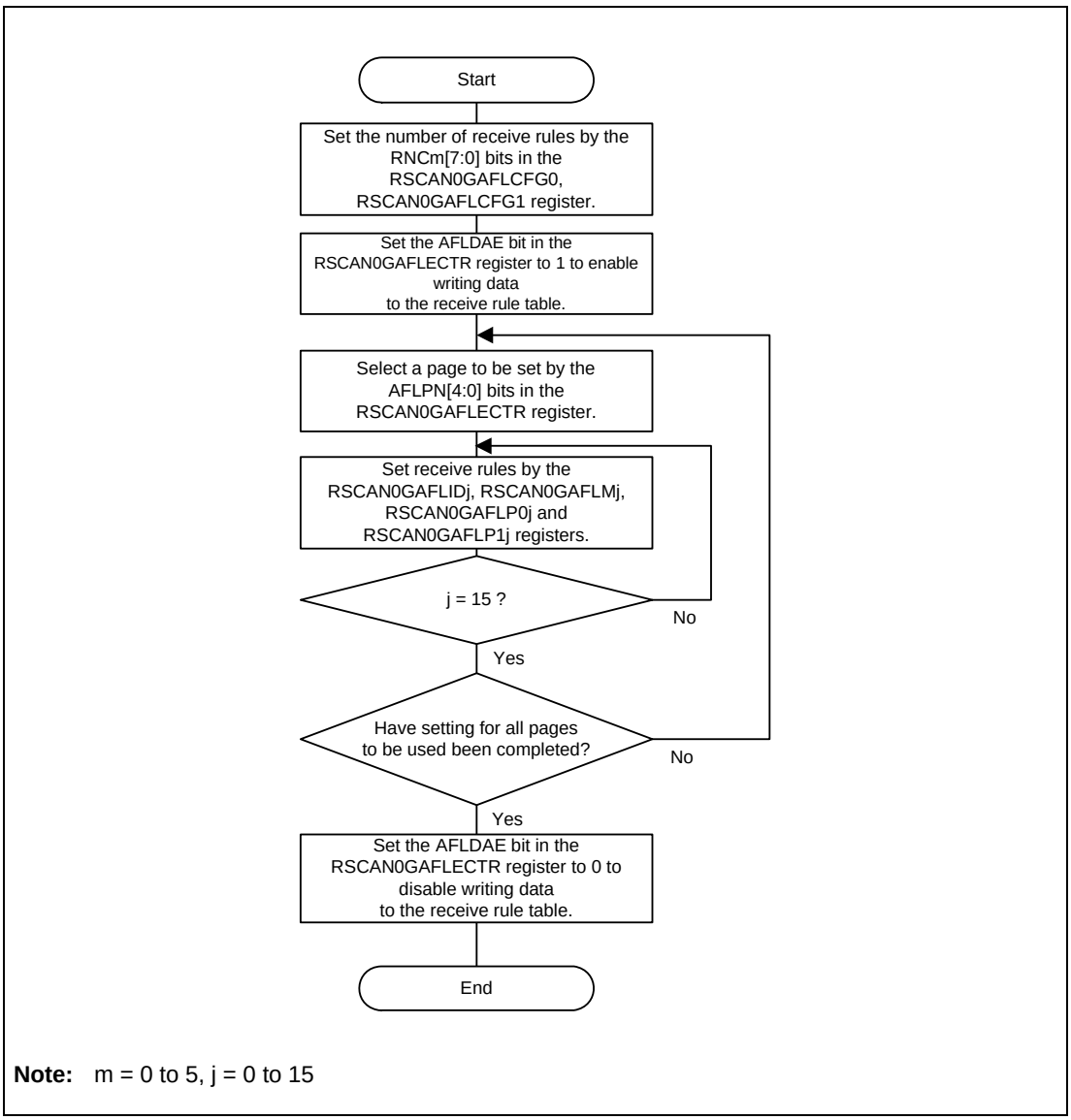
Note: Values in ( ) are baud rate prescaler division values.

19.10.1.4 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 19.19 shows the receive rule setting procedure.



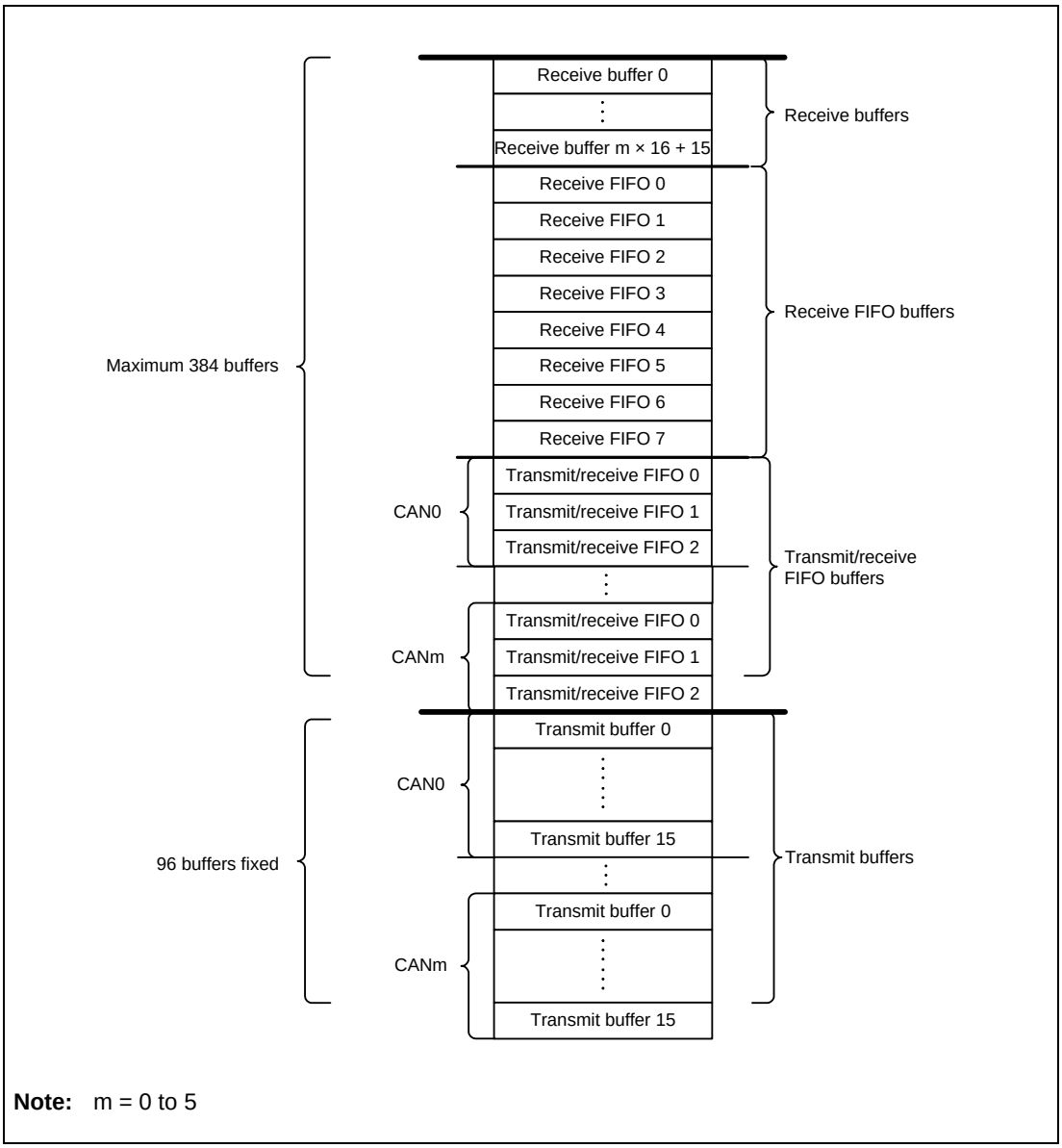
Note: m = 0 to 5, j = 0 to 15

Figure 19.19 Receive Rule Setting Procedure

19.10.1.5 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 19.20 shows the buffer configuration. **Figure 19.21** shows the buffer setting procedure.



Note: m = 0 to 5

Figure 19.20 Buffer Configuration

CAUTION

Receive buffers, receive FIFO buffers, transmit/receive FIFO buffers, and transmit buffers are located in succession.

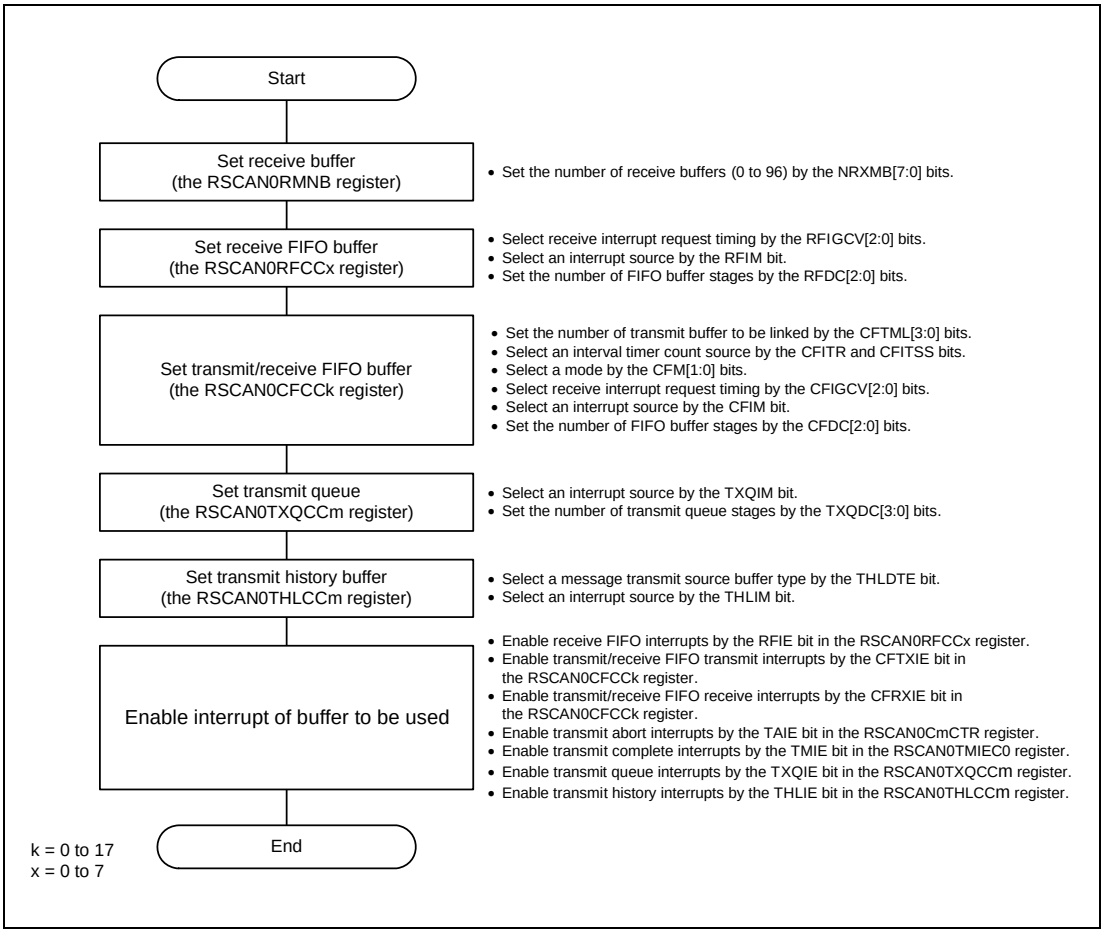
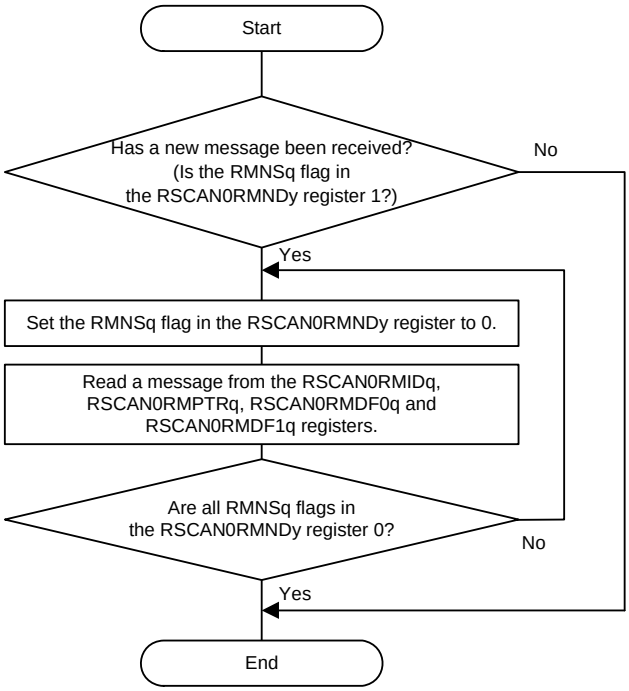


Figure 19.21 Buffer Setting Procedure

19.10.2 Reception Procedure

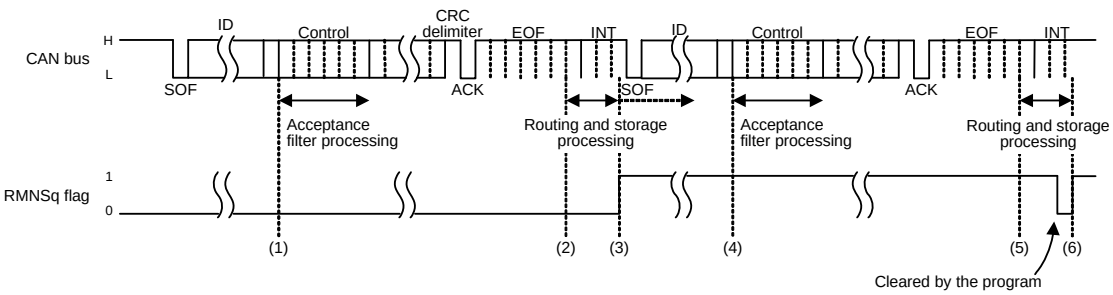
19.10.2.1 Receive Buffer Reading Procedure

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 19.22** shows the receive buffer reading procedure.



Note: y = 0 to 2, q = 0 to 95

Figure 19.22 Receive Buffer Reading Procedure



Note: y = 0 to 2, q = 0 to 95
RMNSq: Flag in the RSCAN0RMNDy register

Figure 19.23 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.

19.10.2.2 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.

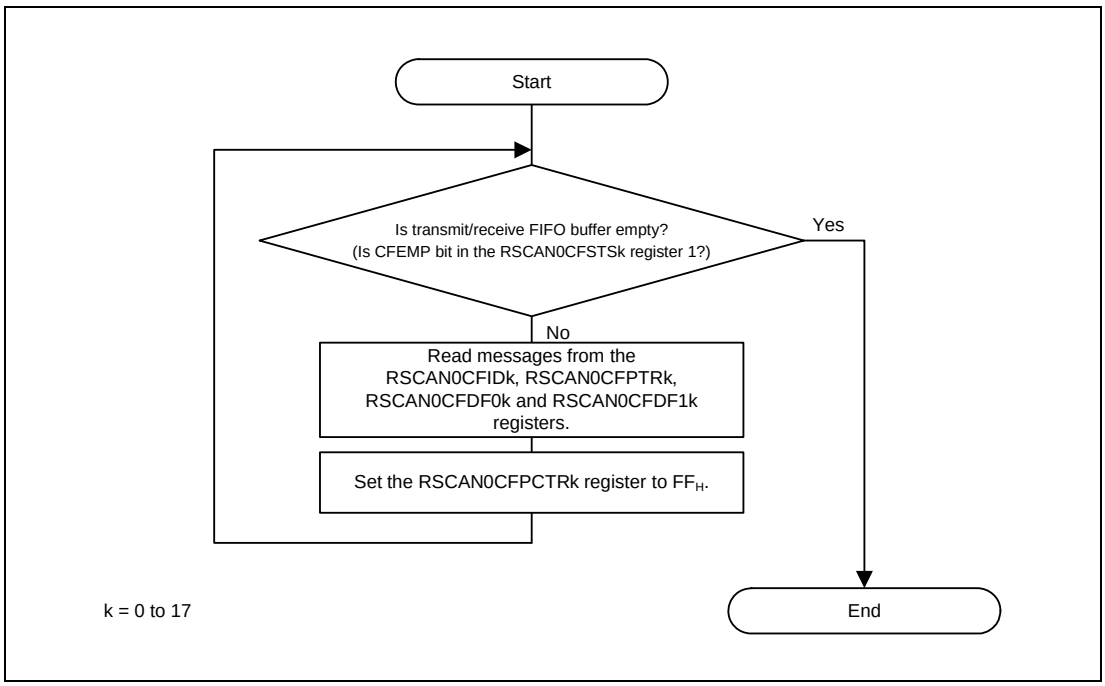


Figure 19.24 Transmit/Receive FIFO Buffer Reading Procedure

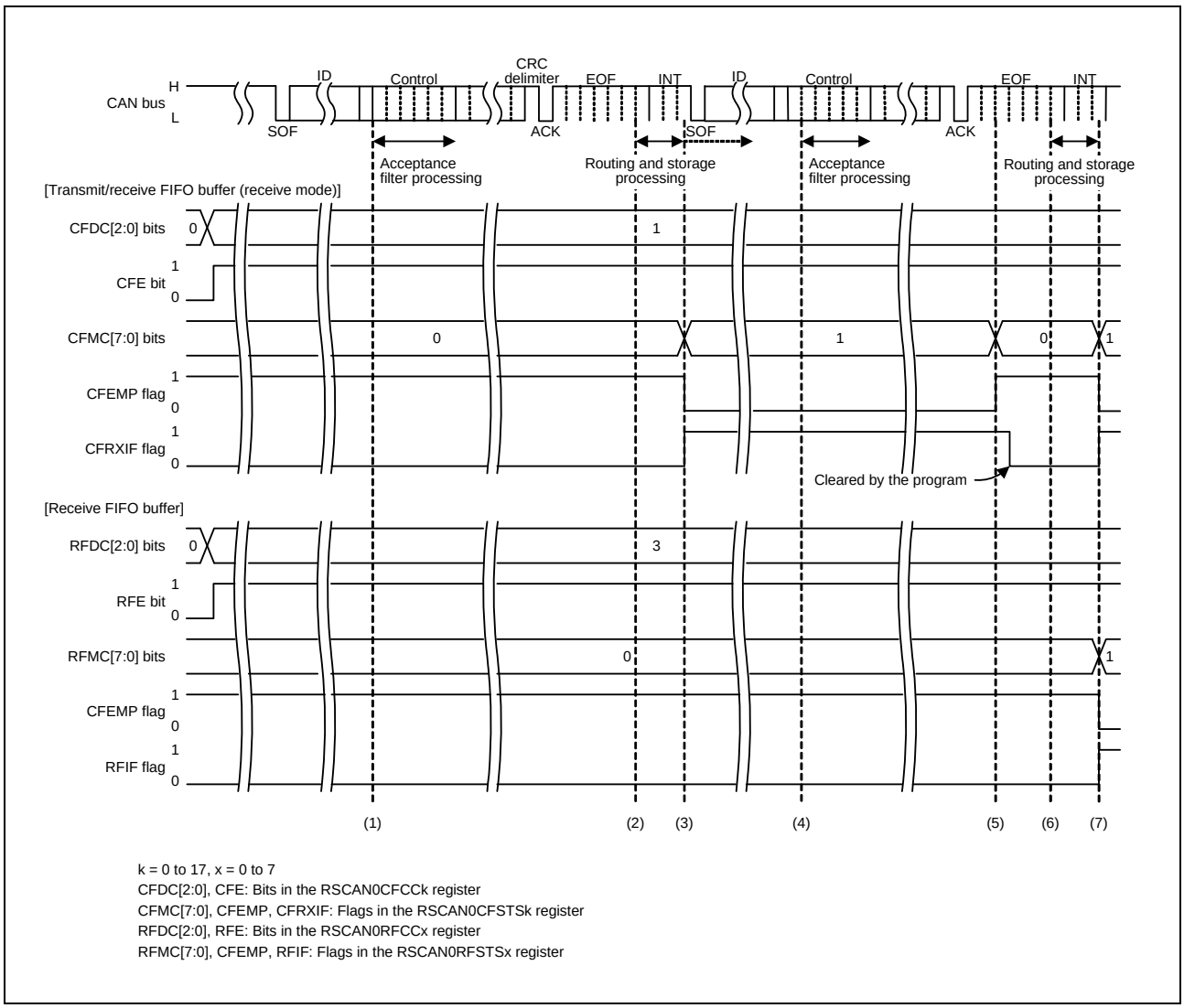


Figure 19.25 FIFO Buffer Reception Timing Chart

(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] 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] value in the RSCAN0CFSTSk register is 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] becomes 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] bits are 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] bits in the RSCAN0RFCCx register are set to 001_B or more. The RFMC[7:0] bits in the RSCAN0RFSTSx register are 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).

19.10.3 Transmission Procedure

19.10.3.1 Procedure for Transmission from Transmit Buffers

Figure 19.26 shows the procedure for transmission from transmit buffers.

Figure 19.27 shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmission has been successfully completed. **Figure 19.28** shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmit abort has been completed.

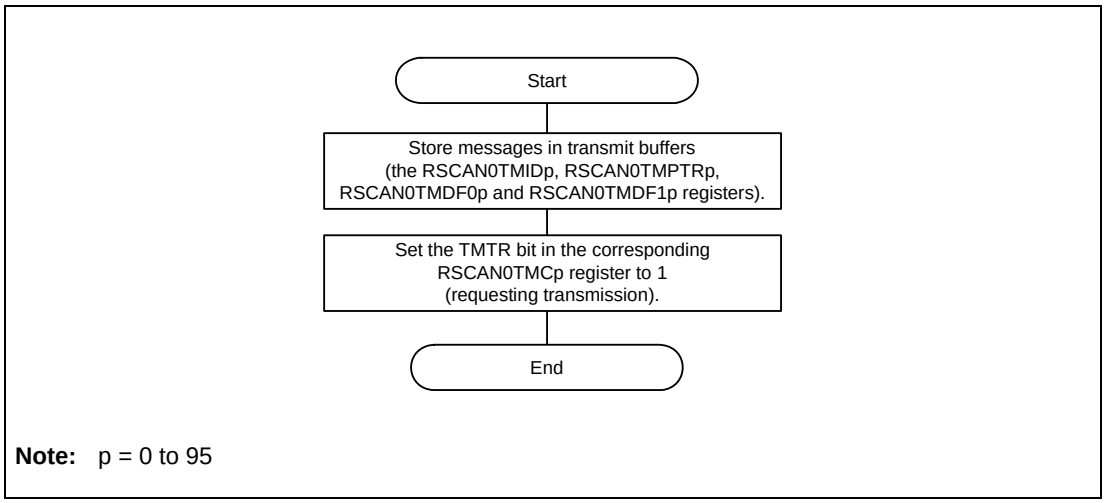
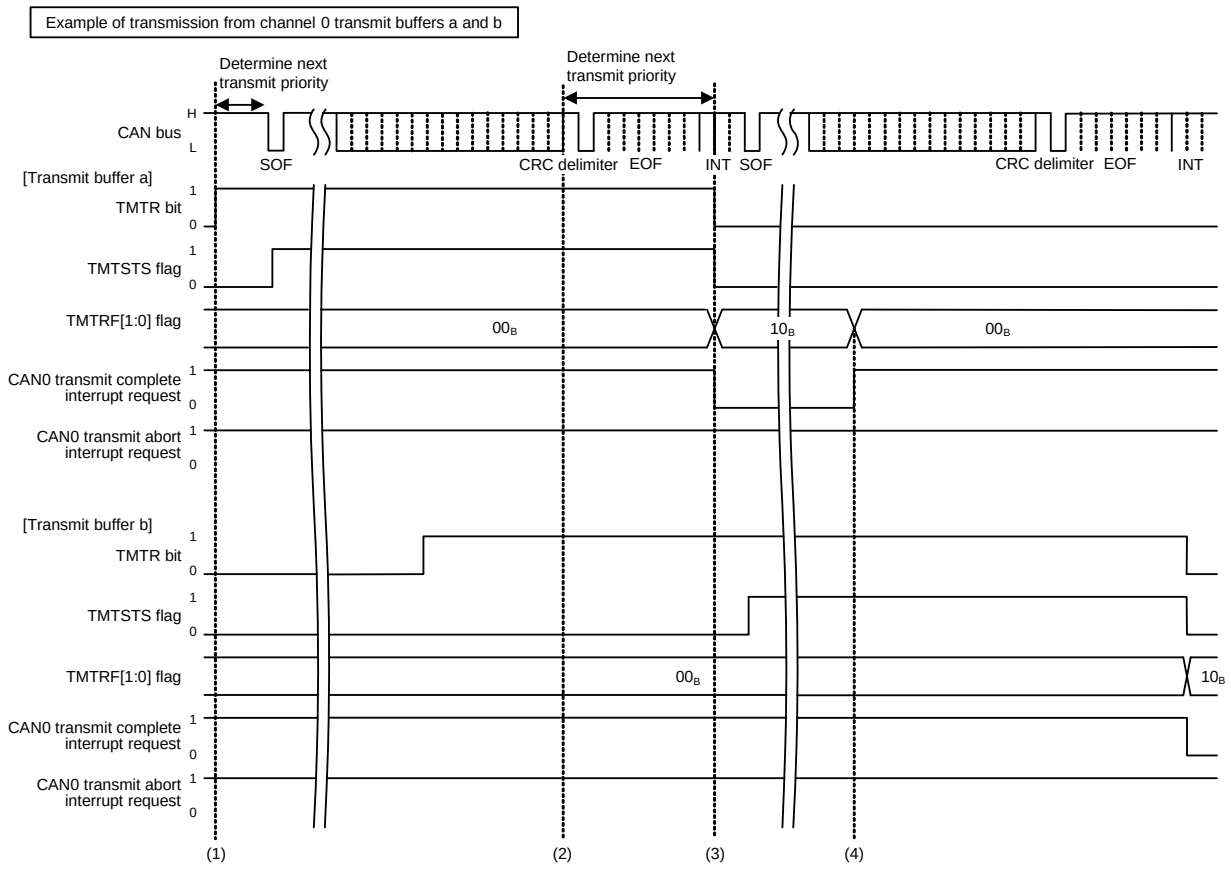


Figure 19.26 Procedure for Transmission from Transmit Buffers



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 19.27 Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)

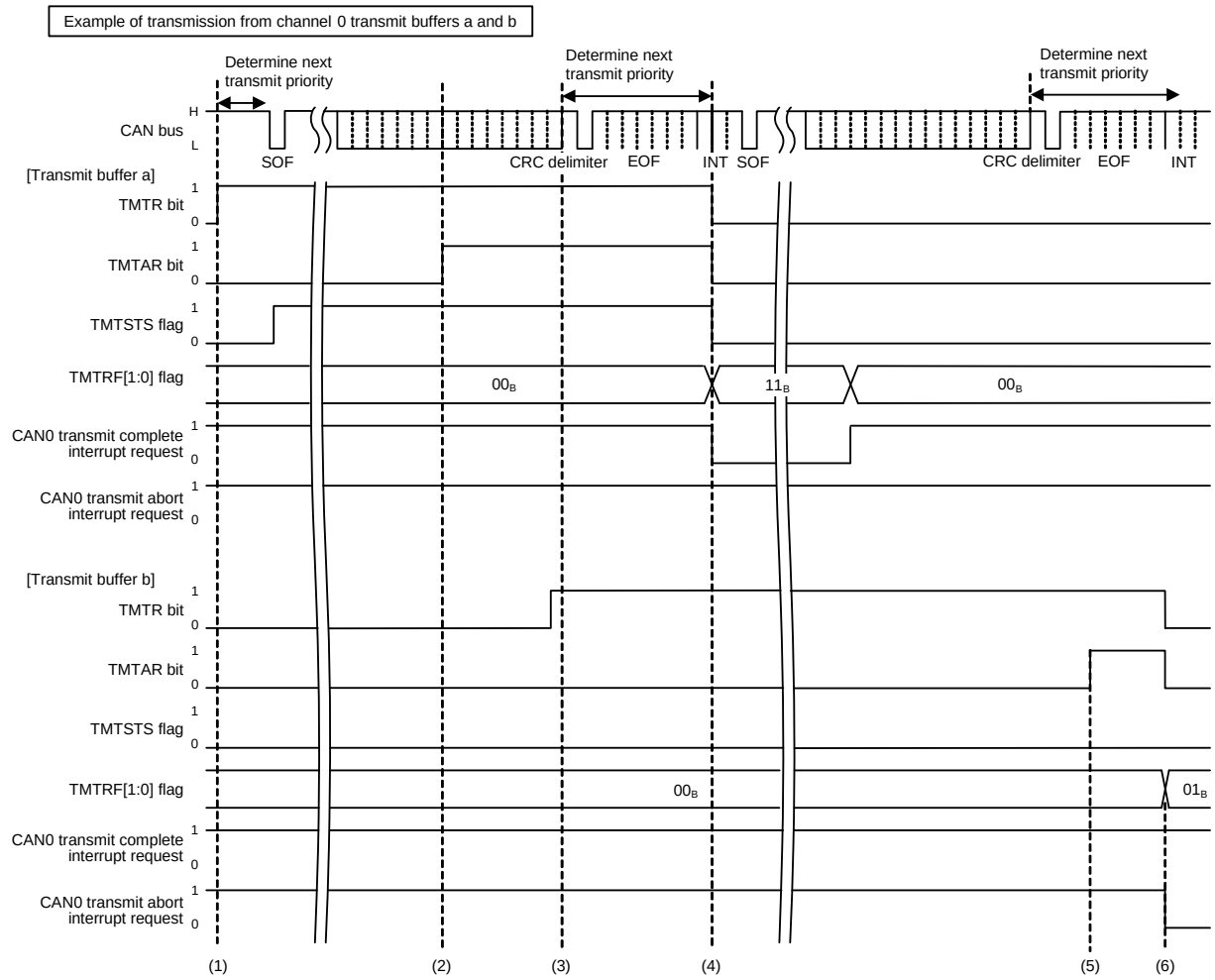
(1) When the TMTR bit in the RSCAN0TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding 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 with the CRC delimiter 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 value is 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 delimiter 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 19.28 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 with the CRC delimiter 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] flag 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 delimiter 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.

19.10.3.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

Figure 19.29 shows the procedure for transmission from transmit/receive FIFO buffers.

Figure 19.30 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 19.31** shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmit abort has been completed.

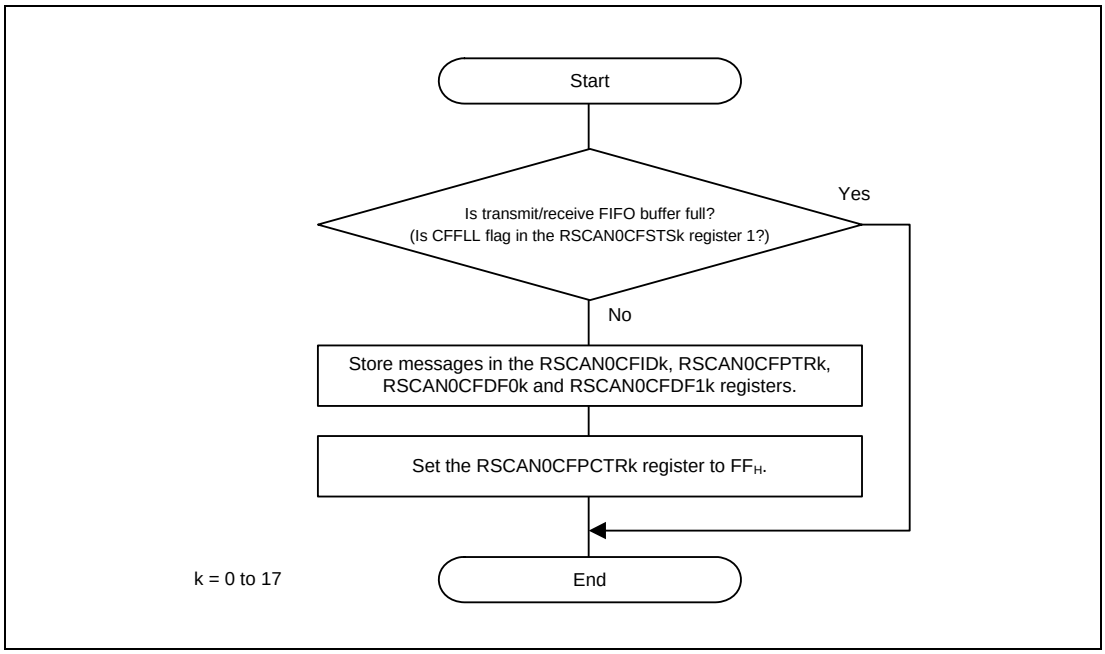


Figure 19.29 Procedure for Transmission from Transmit/Receive FIFO Buffers

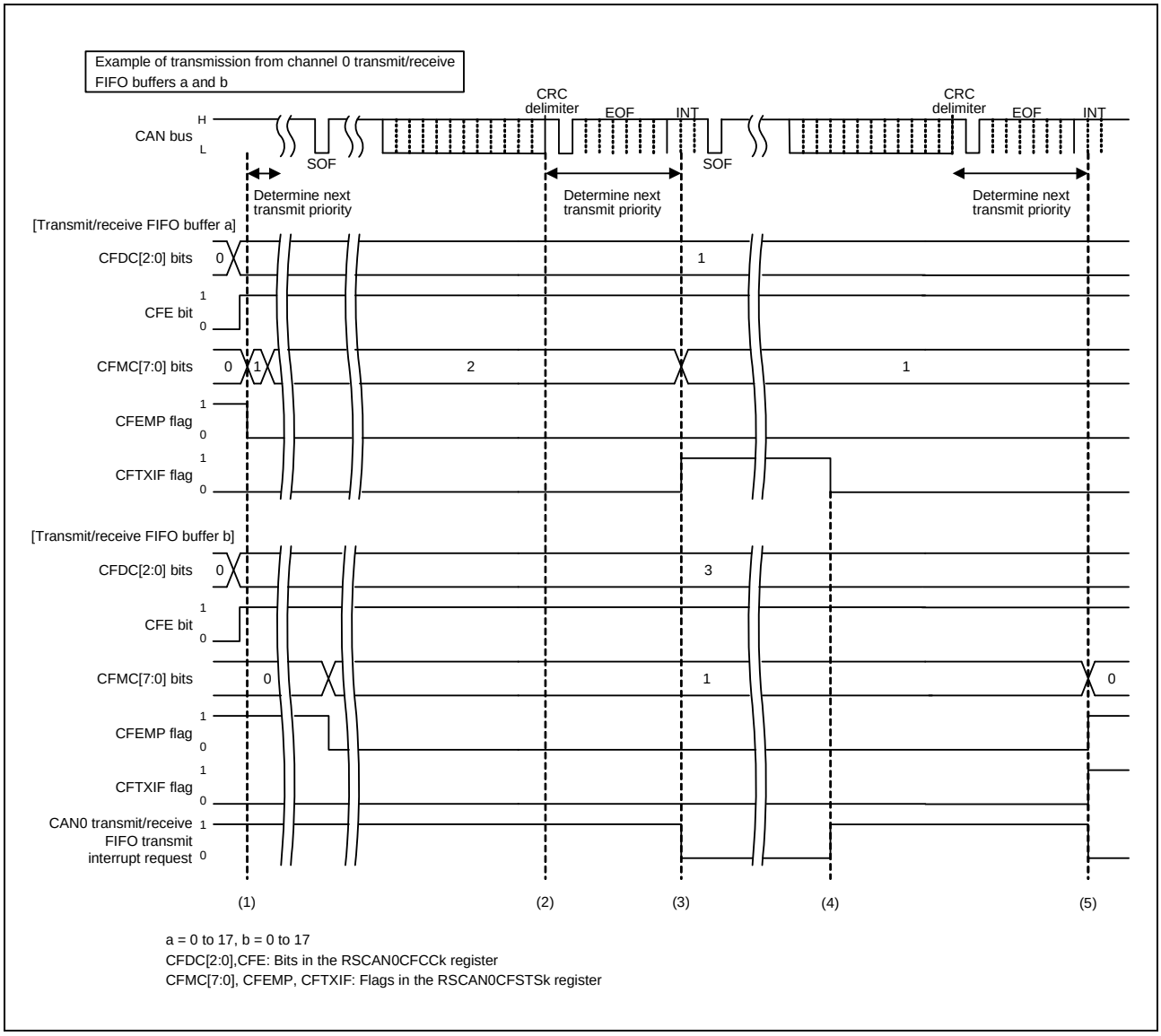


Figure 19.30 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)

(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] value in the RSCAN0CFCCa register is 001_B (4 messages) or more and the CFMC[7:0] 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 with the CRC delimiter 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] value in the RSCAN0CFSTSa register is 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] value in the RSCAN0CFSTSb register is 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).

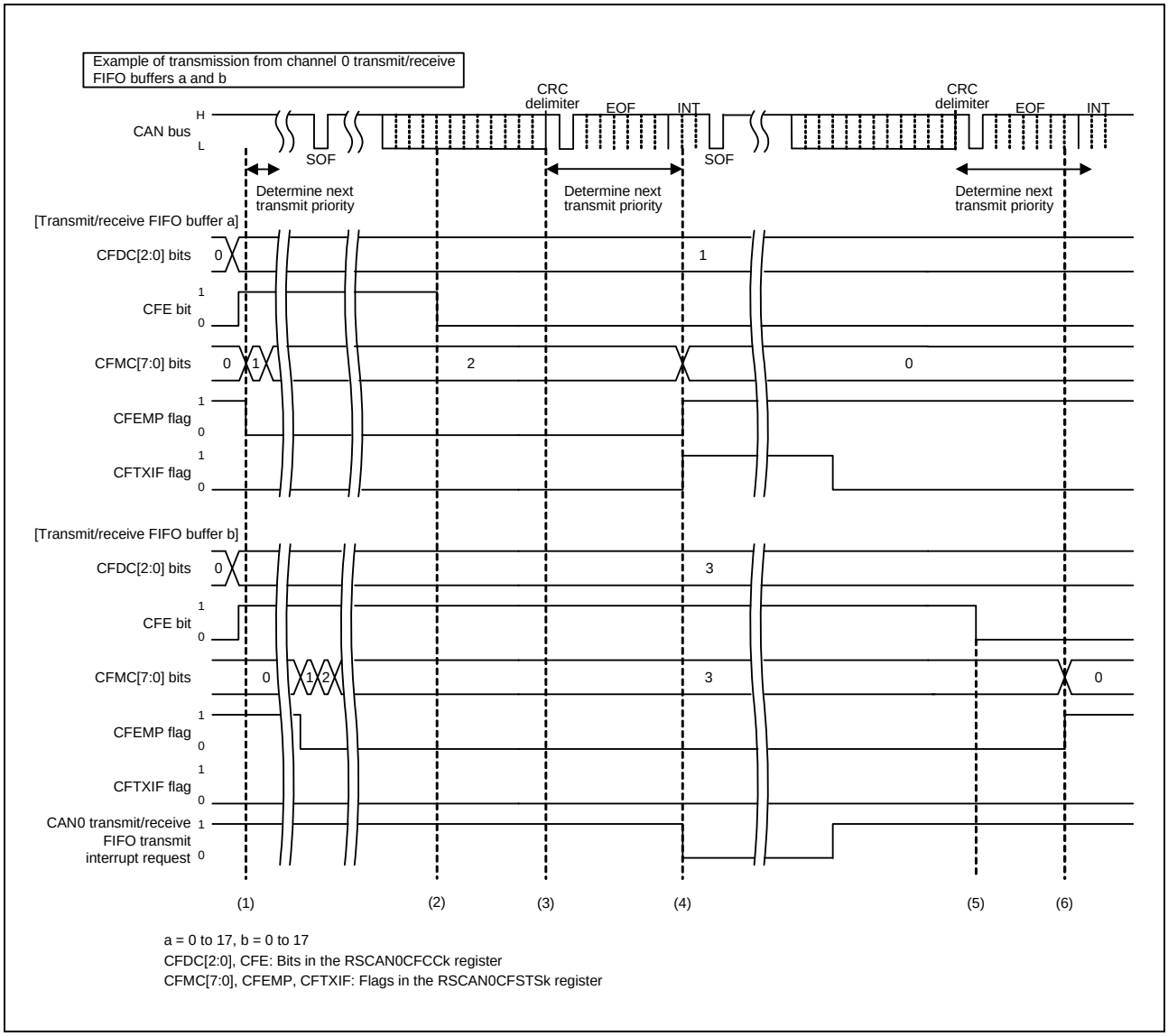


Figure 19.31 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)

(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] value in the RSCAN0CFCCa register is 001_B (4 messages) or more and the CFMC[7:0] 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 with the CRC delimiter 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] 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.)

19.10.3.3 Procedure for Transmission from the Transmit Queue

Figure 19.32 shows the procedure for transmission from the transmit queue.

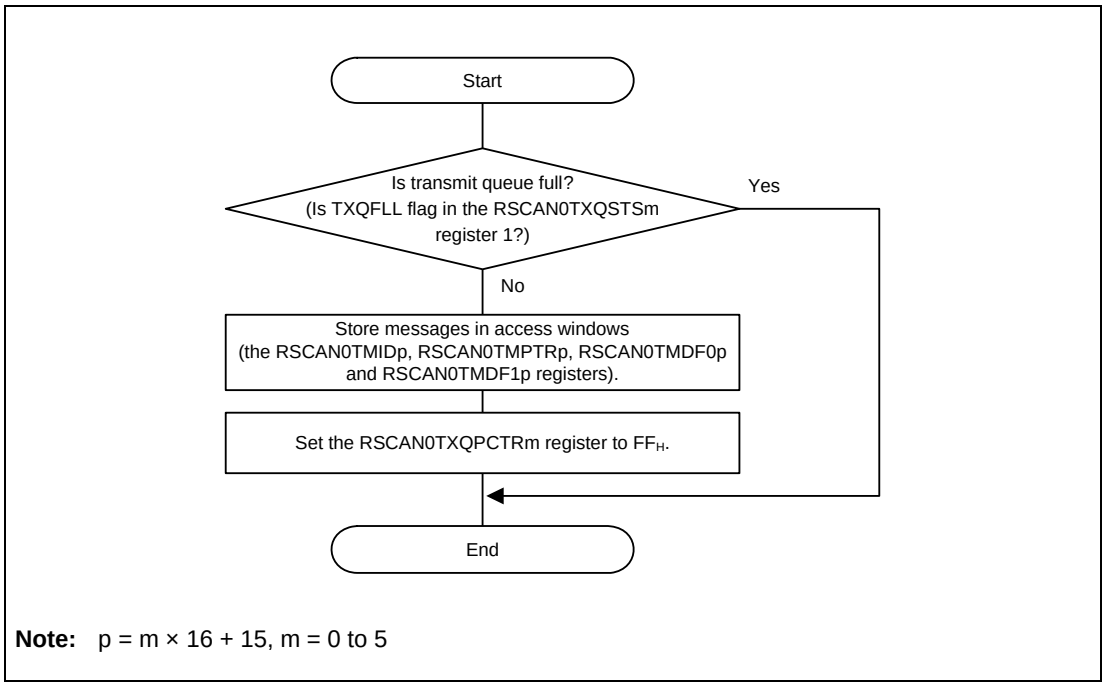
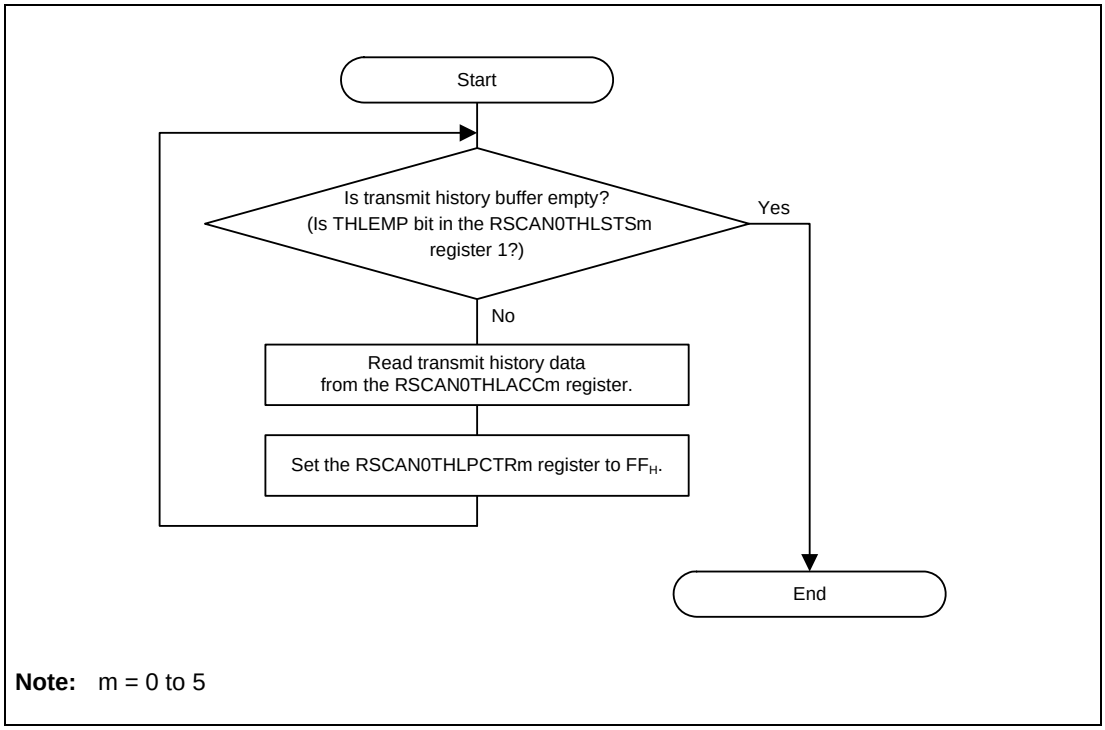


Figure 19.32 Procedure for Transmission from the Transmit Queue

19.10.3.4 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 19.33** shows the transmit history buffer reading procedure.



Note: m = 0 to 5

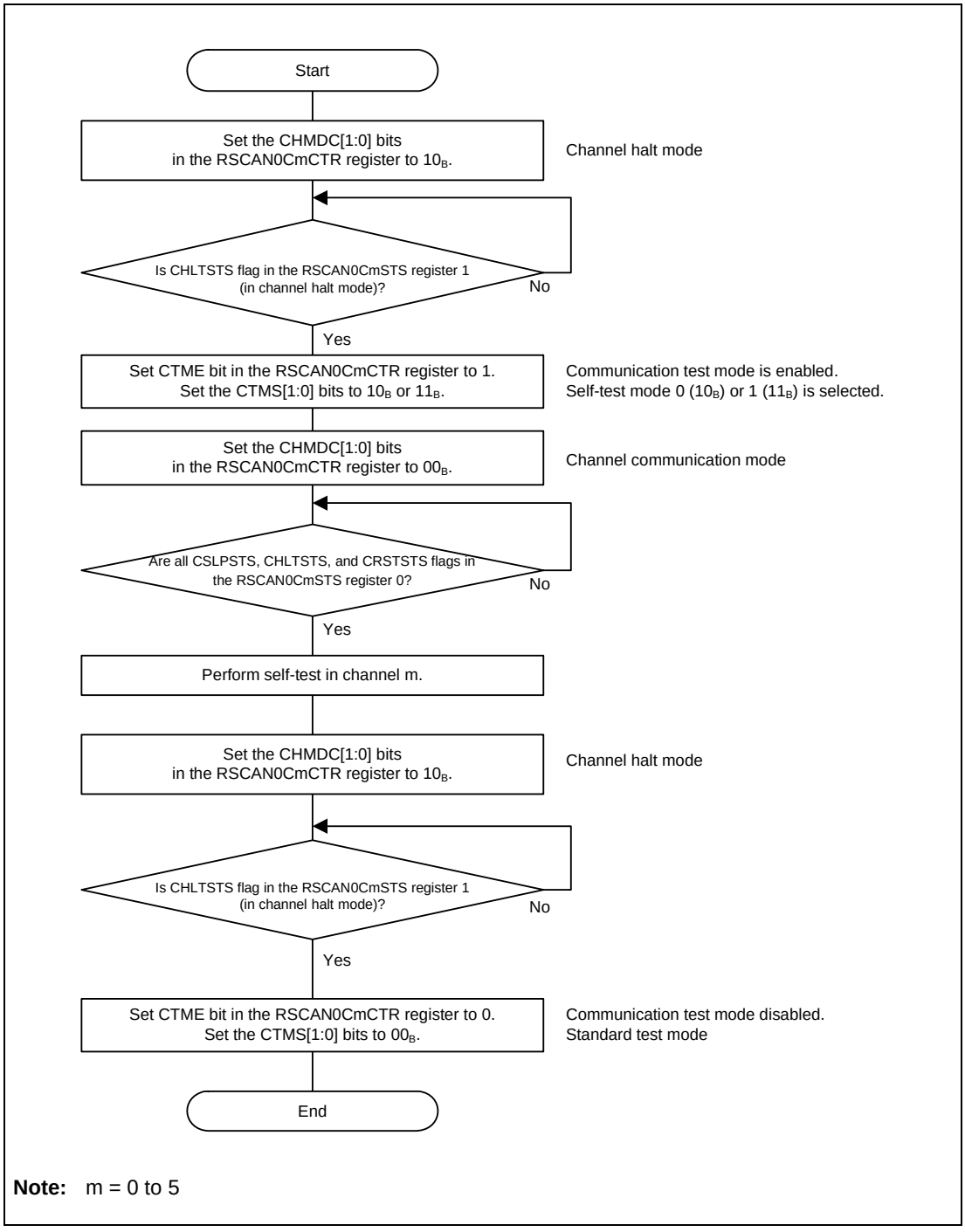
Figure 19.33 Transmit History Buffer Reading Procedure

19.10.4 Test Settings

19.10.4.1 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 19.34 shows the self-test mode setting procedure.



Note: m = 0 to 5

Figure 19.34 Self-Test Mode Setting Procedure

19.10.4.2 Procedure for Releasing the Protection

Since the global test function in **Table 19.98** 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 19.98 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 19.35** shows the procedure for releasing the protection.

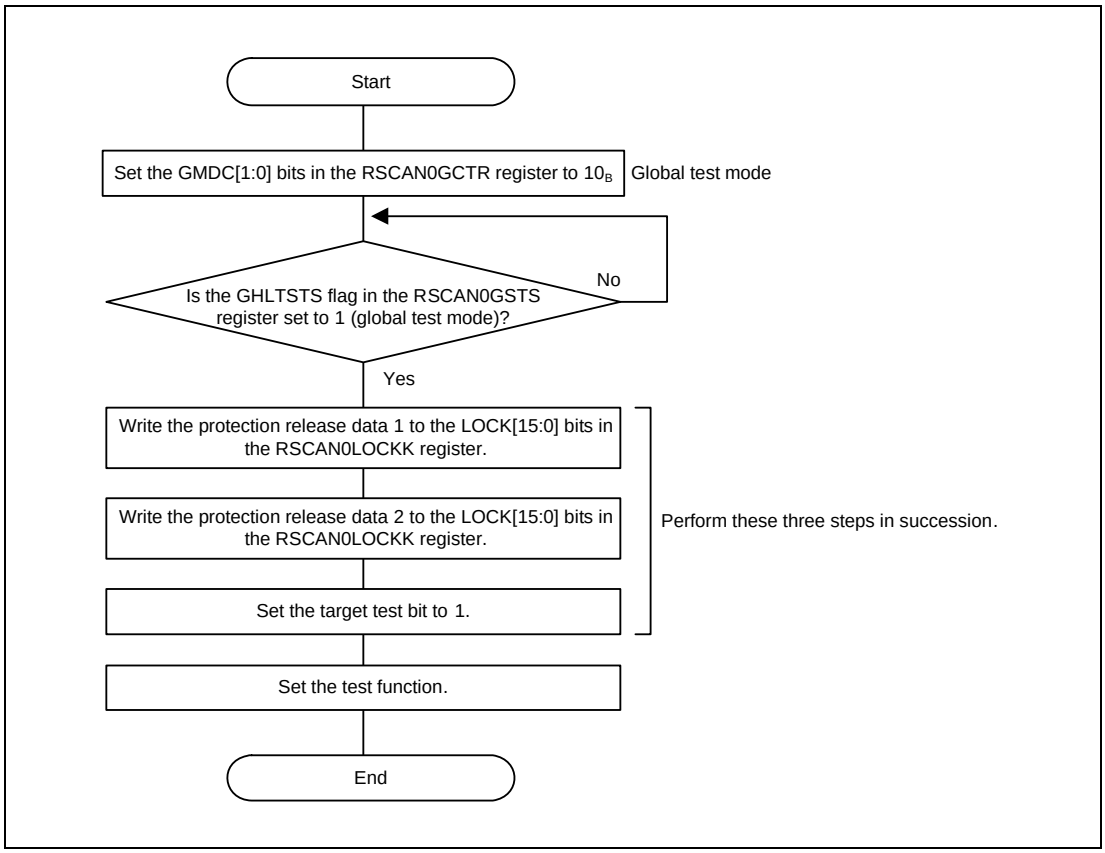


Figure 19.35 Protection Release Procedure

19.10.4.3 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 19.36 shows the RAM test setting procedure.

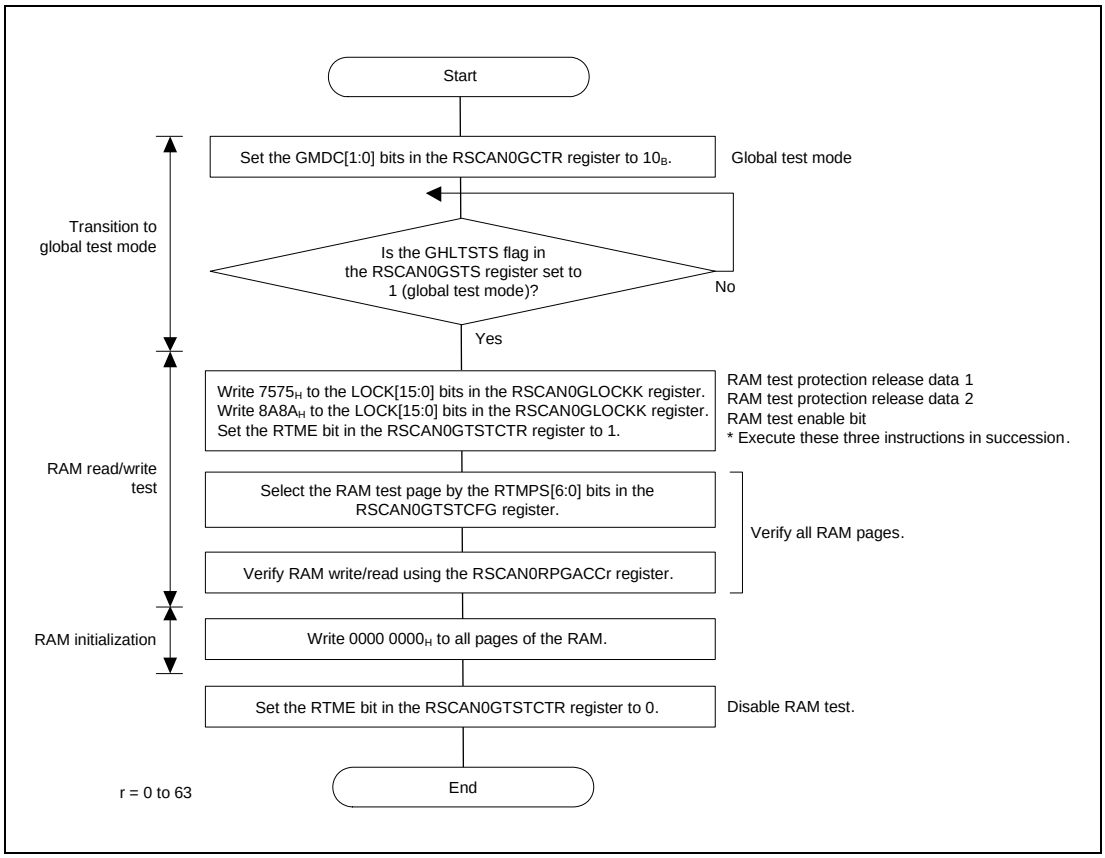
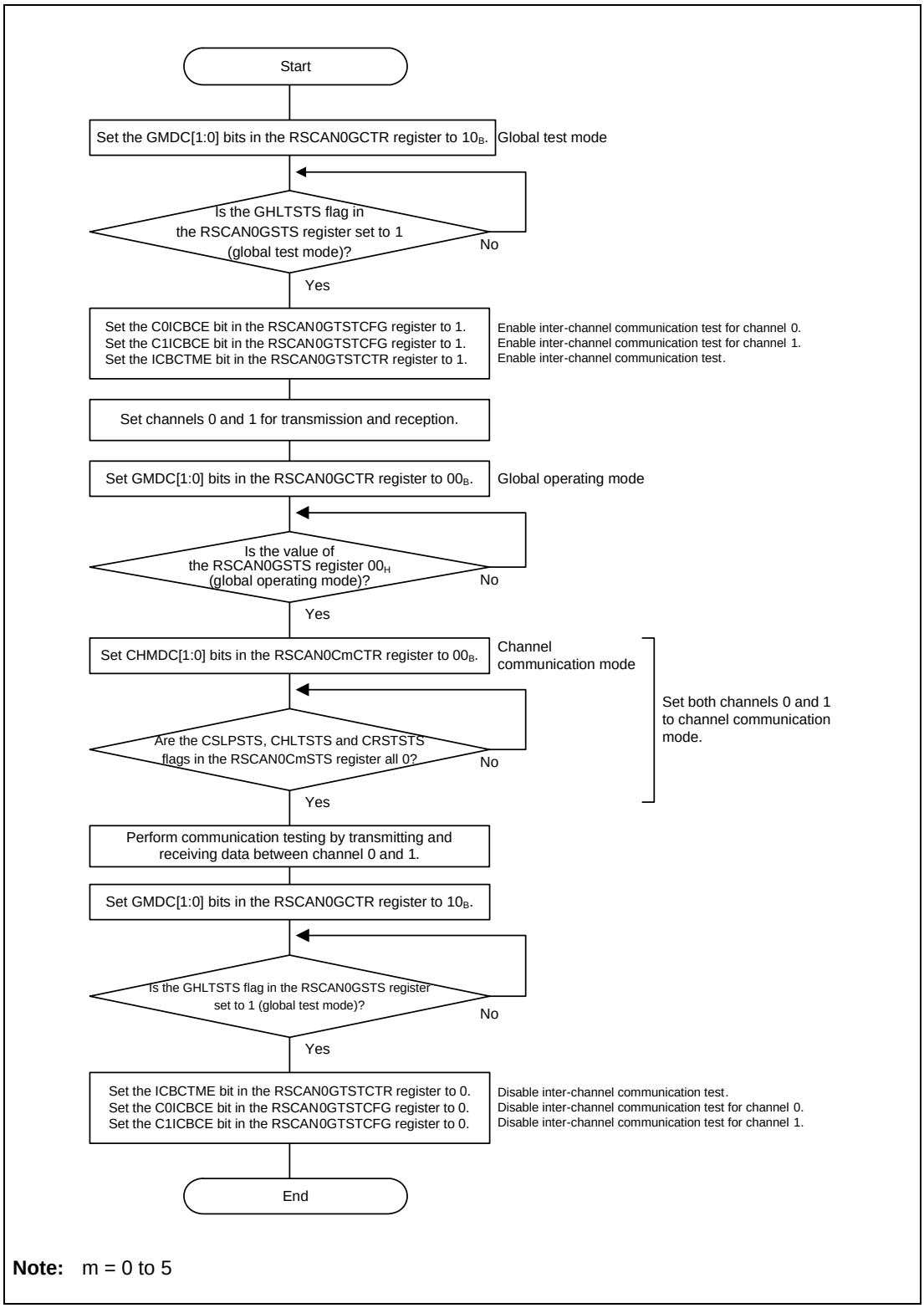


Figure 19.36 RAM Test Setting Procedure

19.10.4.4 Inter-Channel Communication Test Setting Procedure

Communication testing can be performed by transmitting and receiving data between different channels.

Figure 19.37 shows the inter-channel communication test setting procedure.



Note: m = 0 to 5

Figure 19.37 Inter-Channel Communication Test Setting Procedure (Example of Communication Test between Channel 0 and Channel 1)

19.11 Detection and Correction of Errors in RS-CAN RAM

19.11.1 ECC for the RSCAN0 RAM

Table 19.99 gives an outline of the ECC functions for the RSCAN0 RAM.

Table 19.99 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, initialize the RSCAN0 RAM by the RS-CAN module before it is used.

19.11.2 Interrupt Request

Table 19.100 lists the ECC interrupt request of RSCAN0 RAM.

Table 19.100 RS-CAN ECC Interrupt Request (FE-Level Maskable Interrupt)

Unit Interrupt Signal Name	Outline	Name	DMA Trigger Number
—	RSCAN ECC2 bit error interrupt	INTECCDCNRAM	—

19.11.3 ECCRCAN0CTL — RSCAN0 ECC Control Register

The ECCRCAN0CTL 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/written in 16-bit units.

Address: FFC7 1000_H

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	PROT0	—	—	—	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 19.101 ECCRCAN0CTL 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. Writing to these bits, 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. Writing to these bits, 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. Writing to these bits, 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 19.101 ECCRCAN0CTL 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 INTECCDCNRAM interrupt will not be generated. 1: When 2-bit error is detected, a INTECCDCNRAM interrupt will be generated. (initial value)
3	Reserved	When read, the value after reset is returned. Writing to these bits, 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 (INTECCDCNRAM) 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.

19.11.4 ECCRCAN0TMC — RSCAN0 ECC Test Mode Control Register

The ECCRCAN0TMC 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 in 16-bit units.

Address: FFC7 1004_H

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 19.102 ECCRCAN0TMC 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. Writing to these bits, 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: ECCRCAN0TED, ECCRCAN0TRC, ECCRCAN0SYND, ECCRCAN0HORD, ECCRCAN0ECRD, ECCRCAN0ERDB Register test control bit: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
4	ECTRRS	ECC RAM read test mode selection bit Select reading destination when reading ECCRCAN0TED register and reading destination when reading ECCRCAN0ERDB 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 ECCRCAN0TED register is the write value of the ECCRCAN0TED register. The read value of the ECCRCAN0ERDB register is the write value of the ECCRCAN0ERDB register. 1: The read value of the ECCRCAN0TED register can read RAM data. The read value of the ECCRCAN0ERDB register is the ECC data to be written to RAM.
3	ECREOS	ECC redundant bit output data selection bit This bit is used to identifyand select the ECC data to be stored in RAM as ECC data generated for write data or the value of the ECCRCAN0ERDB 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 ECCRCAN0ERDB register to RAM.

Table 19.102 ECCRCAN0TMC 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 ECCRCAN0TED 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 ECCRCAN0TED register.
1	ECDCS	ECC decoder input selection bit This bit is used to indetifyand select the targeted data for syndrome code generation and error detection as RAM data or the value of the ECCRCAN0TED 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 ECCRCAN0TED register and detect errors.
0	ECREIS	ECC redundant bit input data selection bit This bit is used to indetifyand select the targeted data for sy nd rom code generation and error detection as RAM data or the value of the ECCRCAN0ERDB 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 ECCRCAN0ERDB register and detect errors.

19.11.5 ECCRCAN0TED — RSCAN0 ECC Encode/Decode Input/Output Replacement Test Register

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 (ECCRCAN0TMC.ECTMCE = 1), it is accesible. When ECCRCAN0TMC.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/written in 32-bit units.

Address: FFC7 $100C_H$

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 19.103 ECCRCAN0TED Register Contents

Bit position	Bit Name	Function
31 to 0	ECEDB[31:0]	When ECCRCAN0TMC.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 ECCRCAN0TMC.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 (ECCRCAN0SYND). In addition, when ECCRCAN0TMC.ECTRRS = 1, RAM data [31:0], instead of written data, is read for the value of this register.

19.11.6 ECCRCAN0TRC — RSCAN0 ECC Redundant Bit Data Control Test Register

In ECC test mode, this test register, for ECC data, consists of four 8-bit registers, ECCRCAN0SYND, ECCRCAN0HORD, ECCRCAN0ECRD, and ECCRSCAN0ERDB.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), this register can be accessed. When ECCRCAN0TMC.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/written in 32-bit units.

Address: FFC7 1008_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECCRCAN0SYND (See **Section 19.11.7**)								ECCRCAN0HORD (See **Section 19.11.8**)							
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
	ECCRCAN0ECRD (See **Section 19.11.9**)								ECCRSCAN0ERDB (See **Section 19.11.10**)							
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

19.11.7 ECCRCAN0SYND — RSCAN0 ECC Decode Syndrome Data Register

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 (ECCRCAN0TMC.ECTMCE = 1), this register can be accessed. When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00_H is read.

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

Address: FFC7 $100B_H$

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 19.104 ECCRCAN0SYND 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.

19.11.8 ECCRCAN0HORD — RSCAN0 ECC 7-Bit Redundant Bit Data Hold Test Register

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 (ECCRCAN0TMC.ECTMCE = 1), this register can be accessed. When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00_H is read.

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

Address: FFC7 $100A_H$

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 19.105 ECCRCAN0HORD 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 ECCRCAN0TMC.ECTRRS = 1 and ECCRCAN0TED register is read, ECC code is stored.

19.11.9 ECCRCAN0ECRD — RSCAN0 ECC Encode Test Register

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 (ECCRCAN0TMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00_H is read.

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

Address: FFC7 1009_H

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 19.106 ECCRCAN0ECRD 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 ECCRCAN0TED register when ECCRCAN0TMC.ECENS = 1.

19.11.10 ECCRCAN0ERDB — RSCAN0 ECC Redundant Bit Input/Output Replacement Register

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 (ECCRCAN0TMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00_H is read.

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

Address: FFC7 1008_H

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 19.107 ECCRCAN0ERDB Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6 to 0	ERDB[6:0]	When ECCRCAN0TMC.ECREOS = 1, these bits store the value of this register as ECC data to RAM. When ECCRCAN0TMC.ECREIS = 1, the value of this register is read as ECC data read from RAM. When ECCRCAN0TMC.ECTRRS = 1, ECC data to be stored in RAM is read instead of the write data for the read value of this register.

19.11.11 SELB_READTEST — ECCREAD Test Select Register

SELB_READTEST is used to check read/write access to the CSIHn ECC registers and the RSCAN0 ECC registers. For detail, see **Section 15.7.11, SELB_READTEST — ECCREAD Test Select Register**.

19.12 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 20 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/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of WDTA.

20.1 Features of RH850/F1L WDTA

20.1.1 Number of Units and Channels

This microcontroller has the following number of WDTA units.

Table 20.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	2					
Name	WDTAn (n = 0, 1)					

Table 20.2 Index

Index	Meaning
n	Throughout this section, the individual window watchdog timer units are identified by the index “n” (n = 0,1): for example, WDTAnWDTE (n = 0, 1) is the WDTAn enable register.

20.1.2 Register Base Address

WDTAn base addresses are listed in the following table.

WDTAn register addresses are given as offsets from the base addresses in general.

Table 20.3 Register Base Address

Base Address Name	Base Address
<WDTA0_base>	FFED 0000_H
<WDTA1_base>	FFED 1000_H

20.1.3 Clock Supply

The WDTAn clock supply is shown in the following table.

Table 20.4 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
WDTA0	WDTATCKI	CKSCLK_AWDTA
WDTA1	WDTATCKI	LS IntOsc f_{RL}

20.1.4 Interrupt Request

WDTAn interrupt requests are listed in the following table.

Table 20.5 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
WDTA0			
INTWDTAn	WDTA0 75% interrupt	32	—
WDTA1			
INTWDTAn	WDTA1 75% interrupt	33	—

Table 20.6 Interrupt Request (FE Level Non-Maskable Interrupt)

Unit Interrupt Signal	Outline	Interrupt Name	DMA Trigger Number
WDTA0			
WDTAnTNMI	WDTA0 FENMI interrupt (in 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	—

20.1.5 Reset Sources

WDTAn reset sources are listed in the following table. WDTAn is initialized by these reset sources.

Table 20.7 Reset Sources

Unit Name	Reset Source
WDTA0	Reset sources (AWORES)
WDTA1	All reset sources (ISORES)

Note: WDTA1 is stopped in STOP mode.

20.2 Overview

20.2.1 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 20.8**.

- 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.

- Interrupt request generation at 75% of the counter overflow value

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 at a time outside 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 20.5.3, WDTA Error Detection**.

Table 20.8 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 20.5.1, 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 20.5.2, WDTA Trigger** and **20.5.2.1, 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 35.11, Option Bytes**.

20.2.2 Block Diagram

Figure 20.1 shows the main components of the WDTA.

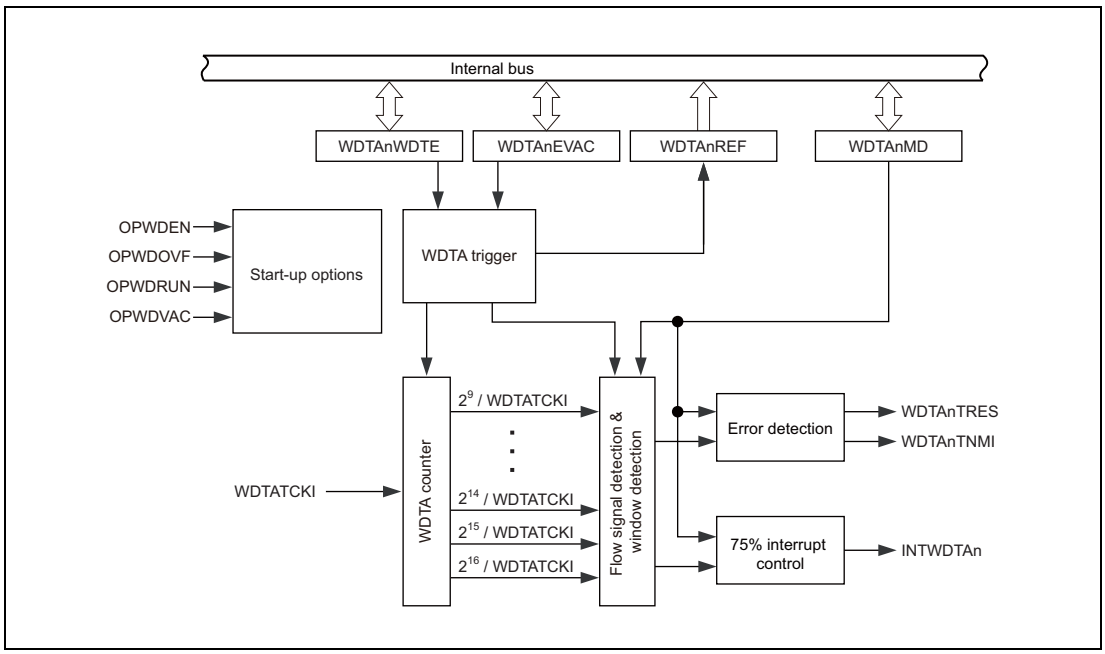


Figure 20.1 Block Diagram of the Window Watchdog Timer A

20.3 Registers

20.3.1 List of Registers

WDTAn registers are listed in the following table.

For details about <WDTAn_base>, see **Section 20.1.2, Register Base Address**.

Table 20.9 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> + $000C_H$

20.3.2 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 20.5.2, WDTA Trigger**, for details.

The behavior of this register depends on the setting of the start-up option OPWDVAC, see **Table 20.12, WDTAnWDTE Behavior**.

Access: This register can be read/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 20.11, Values of WDTAnRUN7 after Reset**.
This register is initialized by a reset of any type.

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

Table 20.10 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 counting. Wiring the value other than AC_H generates an error. The WDTAn cannot be stopped once it was started. See **Table 20.12, 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 20.11** lists the values of the WDTAnRUN7 bit after reset according to the start-up options.

Table 20.11 Values of WDTAnRUN7 after Reset

Start-Up Options				Value of WDTAnRUN7 after Reset
OPWDEN	OPWDVAC	OPWDRUN	Start Mode	
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 20.12, WDTAnWDTE Behavior**.

Table 20.12 WDTAnWDTE Behavior

		WDTAnWDTE	
OPWDVAC	Description	Read	Write
0	The VAC function is disabled. WDTAnWDTE is enabled.	$2C_H$ is read (in software trigger start mode, before the activation of WDTAn). AC_H is read (after the activation of WDTAn).	WDTA trigger Write AC_H *1.
1	The VAC function is enabled. WDTAnWDTE is disabled.	$2C_H$ is read.	Writing is ignored.

Note 1. Any other write value will cause an error.

20.3.3 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 20.5.2, WDTA Trigger**. For details about the activation codes when the VAC function is used, see **Section 20.5.2.1, 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 20.15, WDTAnEVAC Behavior**.

Access: This register can be read/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 20.14, Values of WDTAnEVAC7 after Reset**.
This register is initialized by a reset of any type.

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

Table 20.13 WDTAnEVAC Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTAnEVAC [7:0]	Writing an variable activation code generates the WDTA trigger and starts/ restarts the WDTAn counting. Wiring a wrong activation code generates an error. The WDTAn cannot be stopped once it was started. See **Table 20.15, 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 20.14** lists the values of the WDTAnEVAC7 bit after reset according to the start-up options.

Table 20.14 Values of WDTAnEVAC7 after Reset

Start-Up Options				Value of WDTAnEVAC7 after Reset
OPWDEN	OPWDVAC	OPWDRUN	Start Mode	
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 20.15**.

Table 20.15 WDTAnEVAC Behavior

		WDTAnEVAC	
OPWDVAC	**Description**	**Read**	**Write**
0	The VAC function is disabled. WDTAnEVAC is disabled.	$2C_H$ is read.	Writing is ignored.
1	The VAC function is enabled. WDTAnEVAC is enabled.	$2C_H$ is read (in software trigger start mode, before the activation of WDTAn). The variable activation code written last is read (after the activation of WDTAn).	Write the variable activation code*1 For details, see **Section 20.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

Note 1. Any other write value will cause an error.

20.3.4 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 20.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

If the VAC function is disabled (OPWDVAC = 0), reading this register returns 00_H.

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

Address: <WDTAn_base> + 0008_H

Value after reset: 00_H This register is initialized by a reset of any type.

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 20.16 WDTAnREF Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTAnREF[7:0]	Reference value for activation code calculation for the VAC function

20.3.5 WDTAnMD — WDTA Mode Register

This register specifies the overflow interval time, the 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.

Updating this register after the WDTA has been started generates an error, but an error does not occur if the same value has been written to it.

Access: This register can be read/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 20.8, WDTA Start-Up Options**. This register is initialized by a reset of any type.

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 20.17 WDTAnMD Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
6 to 4	WDTAnOVF[2:0]	Selects the overflow interval time:
3	WDTAnWIE	Enables/disables the 75% interrupt request INTWDTAn. 0: INTWDTAn disabled 1: INTWDTAn 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 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

WDTAn WS1	WDTAn WS0	Window-Open Period
0	0	25%
0	1	50%
1	0	75%
1	1	100%

20.4 Interrupt Sources

WDTA detects the status of the WDTA counter value or 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 set to 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 an NMI interrupt and a reset. For details about WDTA errors, see **Section 20.5.3, WDTA Error Detection**.

20.5 Functions

20.5.1 WDTA after Reset Release

20.5.1.1 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 OPWDRUN.

The start mode selection is listed in **Table 20.18**.

Table 20.18 Start Mode Selection

Start-Up Options 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.

20.5.1.2 WDTA Settings after Reset Release

(1) **Table 20.19** shows the WDTA settings after reset release.

Table 20.19 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 for WDTAnMD is changed after a WDTA trigger is generated, an error occurs. However, an error does not occur if the same value has been set.

20.5.1.3 Default Start Mode Timing

The default start mode timing and the changes to the WDTA settings are illustrated in **Figure 20.2**.

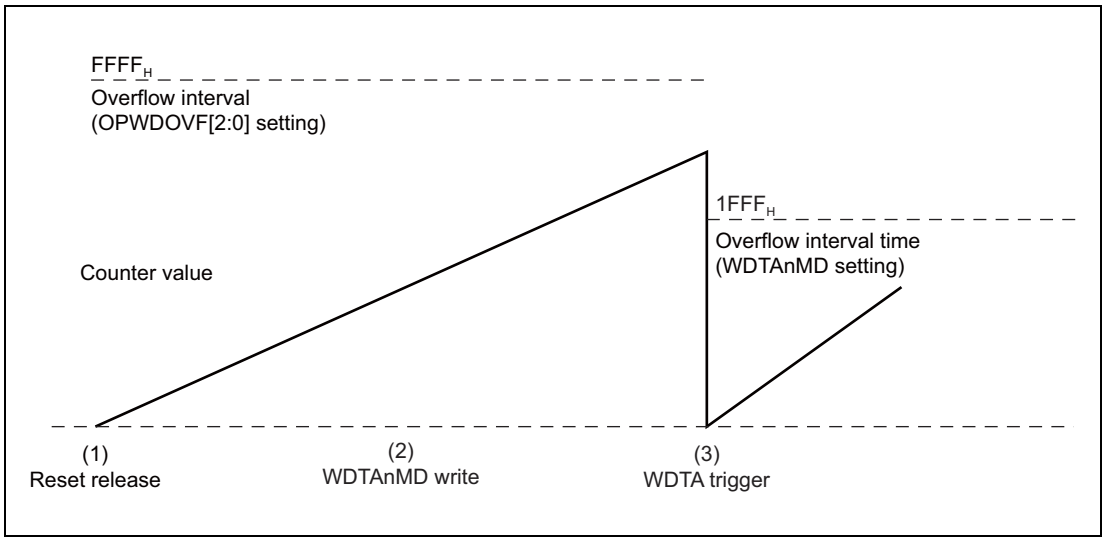


Figure 20.2 Timing Diagram of WDTA Start in Default Start Mode

The timing diagram shown in **Figure 20.2** 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, however, that the setting is not applied immediately.

(3) Write to the WDTA trigger register before the WDTA counter overflows.
The WDTAnMD setting is applied due to the WDTA trigger.

Example: Overflow interval time after a WDTA trigger is generated
= 2^{13}/WDTATCKI

20.5.1.4 Software Trigger Start Mode Timing

The software trigger start mode timing and the changes to the WDTA settings are illustrated in **Figure 20.3**.

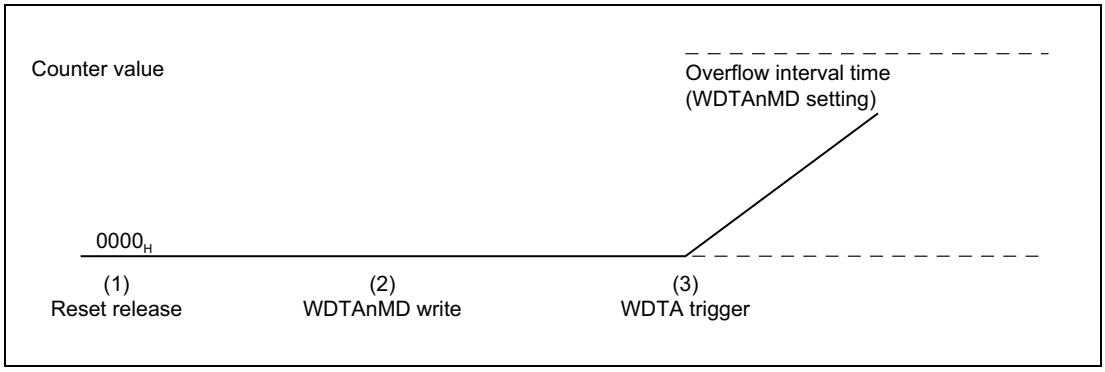


Figure 20.3 Timing Diagram of WDTA Start in Software Trigger Start Mode

The timing diagram shown in **Figure 20.3** 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 it does not affect because the counter operation has not been performed.

(2) WDTAnMD is set before a WDTA trigger is generated. Note, however, that the setting is not applied immediately.

(3) The WDTA counter starts due a WDTA trigger.
The overflow interval time specified in WDTAnMD and other settings are applied.

20.5.2 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
- WDTA mode setting 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 20.20 lists the WDTA trigger registers and activation codes.

Table 20.20 WDTA Trigger and Activation Code

Type of Activation Code	Trigger Register	Activation Code
Fixed (OPWDVAC = 0)	WDTAnWDTE	AC_H
Variable (OPWDVAC = 1)	WDTAnEVAC	For details, see **20.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

20.5.2.1 Calculating an Activation Code when the VAC Function is Used

Use the following expression to calculate the variable activation code (ExpectWDTE) to be set to 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 20.21 lists the variable activation codes according to the number of WDTA triggers.

Table 20.21 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

Wiring a wrong activation code generates an error.

20.5.3 WDTA Error Detection

WDTA detects an error, including generation of the WDTA count overflow or illegal operations.

The following shows when a WDTA error is detected:

- WDTA counter overflow
- Wrong 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 setting value in the WDTA mode register (WDTAnMD) is changed twice before the WDTA trigger is generated

20.5.3.1 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 20.4 shows the reset or NMI request generation when the counter overflows and default start mode is selected.

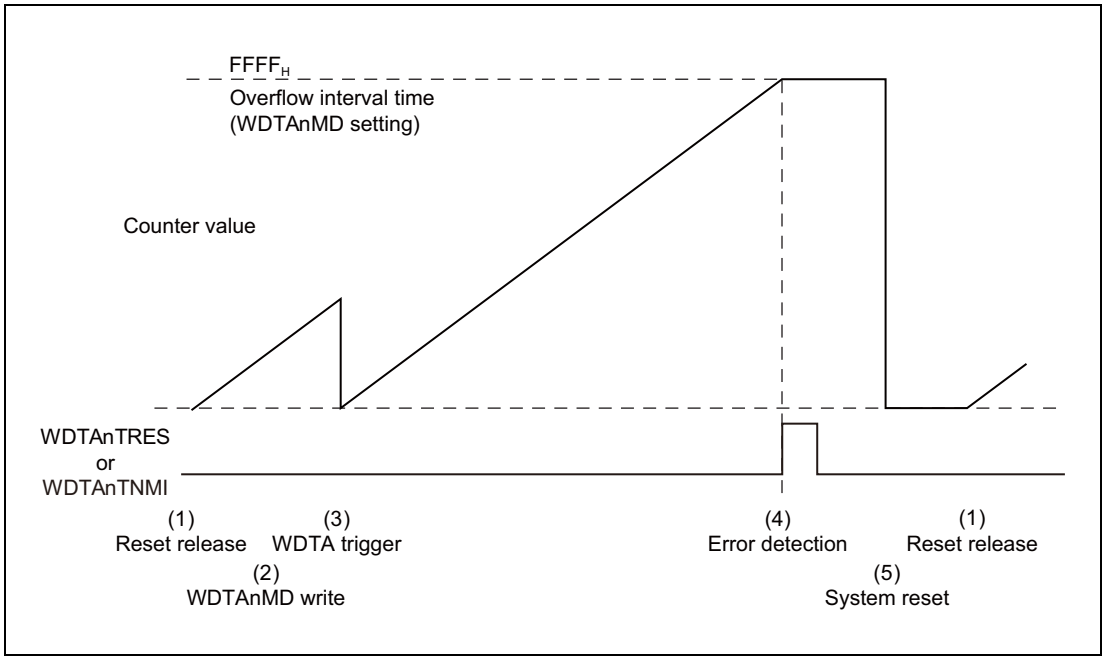


Figure 20.4 Timing Diagram of WDTA NMI Request or Reset Generation

The timing diagram shown in **Figure 20.4** 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 until a system reset is performed.

(5) When the system is reset, the counter is cleared and stopped until reset release.

20.5.4 75% Interrupt Request Signals

When the WDTA counter reaches 75% of the time set for the overflow interval, the interrupt request INTWDTAn is generated.

By use of the WDTAnMD.WDTAnWIE register, this function can be enabled or disabled afterwards.

Figure 20.5 shows the 75% interrupt request generation under following conditions:

- Default start mode selected
- 75% interrupt request is enabled after the first WDTA trigger is generated
- WDTA overflow interval time: 2^{16}/WDTATCKI

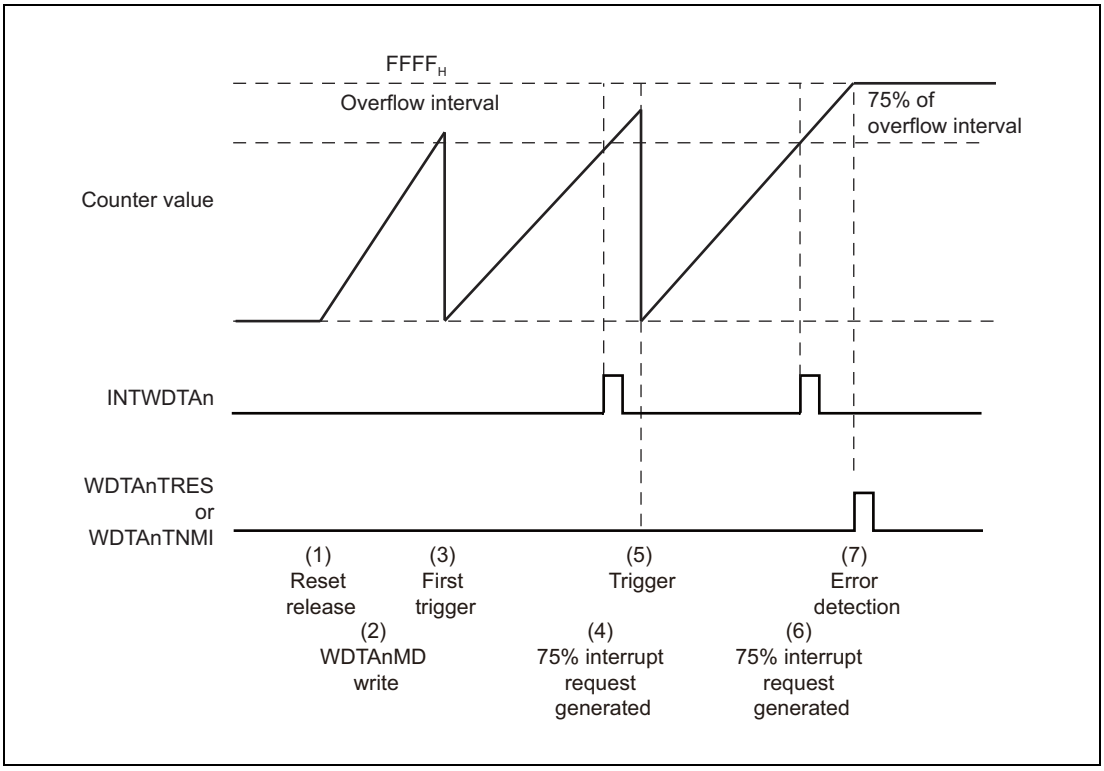


Figure 20.5 Timing Diagram of WDTA 75% Interrupt Request Signals

(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 the 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 until a system reset is performed.

20.5.5 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%, an error occurs by the WDTA trigger generated not in the window-open period. 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 20.6 shows the behavior of the window function under the following conditions.

- Default start mode selected
- 25% window-open period is enabled after the first WDTA trigger is generated (WDTAnWS[1:0] = 00_B)
- WDTA overflow interval time: 2^{16}/WDTATCKI

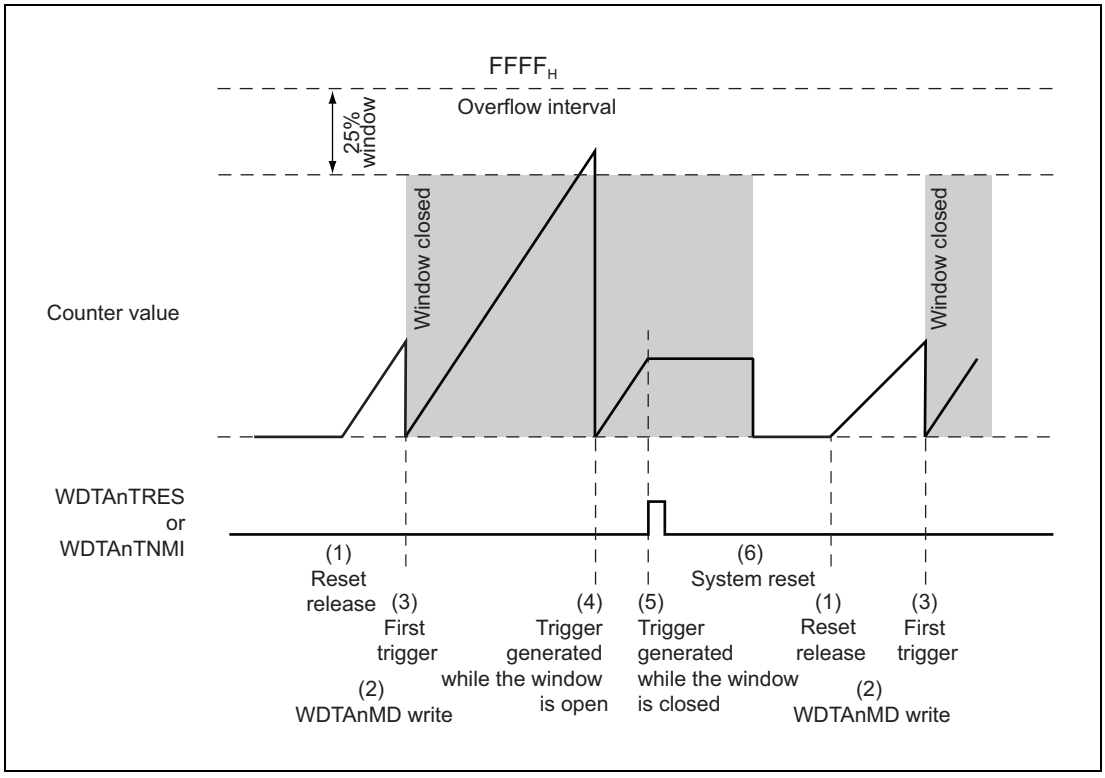


Figure 20.6 Timing Diagram of WDTA Window Function

(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 the 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 until a system reset is performed.

(6) When the system is reset, the counter is cleared and stopped until reset release.

Section 21 OS Timer (OSTM)

This section contains a generic description of the OS Timer (OSTM).

The first part of this section describes all RH850/F1L 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.

21.1 Features of RH850/F1L OSTM

21.1.1 Number of Units

This microcontroller has the following number of units of the OSTM.

Table 21.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1					
Name	OSTMn (n=0)					

Table 21.2 Index

Index	Meaning
n	Throughout this section, the individual OSTM units are identified by the index “n”; for example, OSTMnCNT is the OSTM counter register.

21.1.2 Register Base Address

OSTM base addresses are listed in the following table.

OSTM register addresses are given as offsets from the base addresses in general.

Table 21.3 Register Base Address

Base Address Name	Base Address
<OSTM0_base>	FFEC 0000_H

21.1.3 Clock Supply

The OSTM clock supply is shown in the following table.

Table 21.4 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
OSTM0	PCLK	CPUCLK2

21.1.4 Interrupt Request

OSTM interrupt requests are listed in the following table.

Table 21.5 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
OSTM0			
OSTMTINT	OSTM interrupt	76	—

Table 21.6 Interrupt Request (FE Level Maskable Interrupt Request)

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
OSTM0			
OSTMTINT	OSTM interrupt	INTOSTM0_FE	—

21.1.5 Reset Sources

OSTM reset sources are listed in the following table. OSTM is initialized by these reset sources.

Table 21.7 Reset Sources

Unit Name	Reset Source
OSTM0	All reset sources (ISORES)

21.2 Overview

OSTM is a 32-bit timer/counter.

It can be used in interval timer mode or in free-run compare mode. The settings for operating mode specify the direction of counting (up or down) to control the generation of interrupt requests.

21.2.1 Functional Overview

OSTM has the following features.

- Two operating modes
  - Interval timer mode
  - Free-run compare mode
- OSTMTINT interrupt

21.2.2 Block Diagram

The following block diagram shows the main components of OSTM.

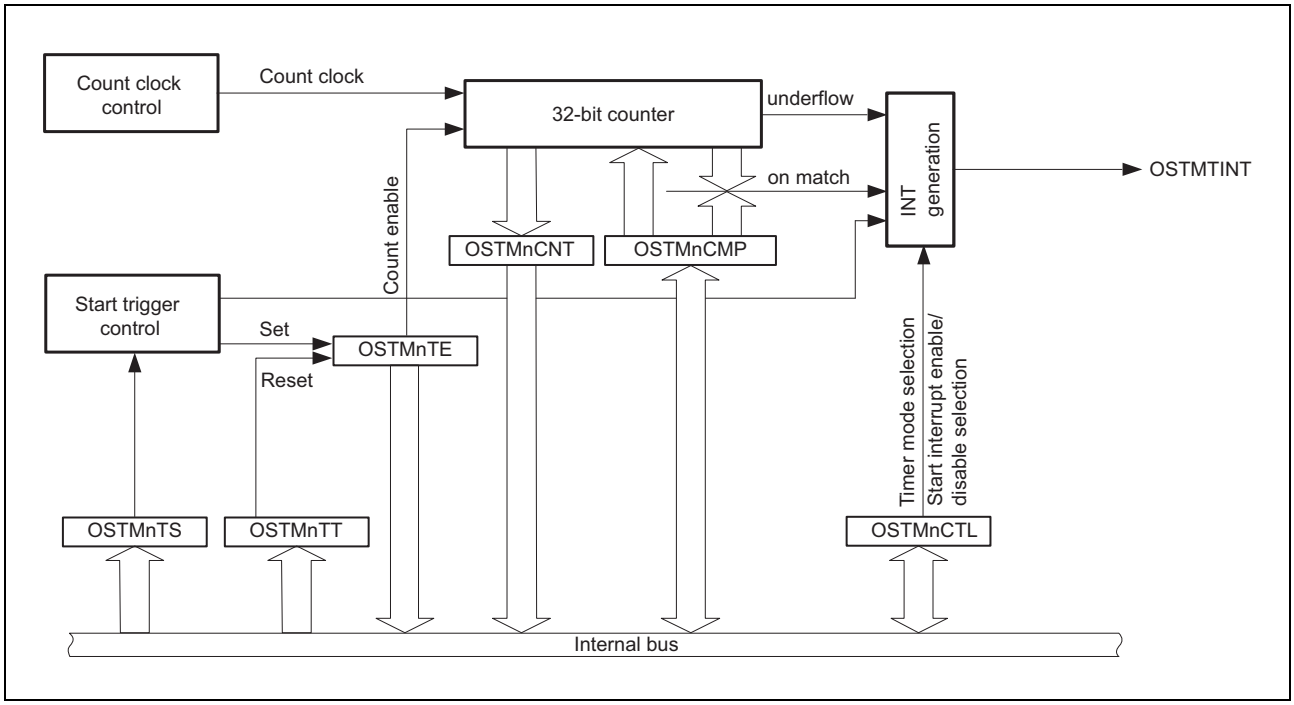


Figure 21.1 Block Diagram of OSTM

21.2.3 Count Clock

The count clock used by OSTM is PCLK.

21.2.4 Interrupt Sources (OSTMTINT)

By default, 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.

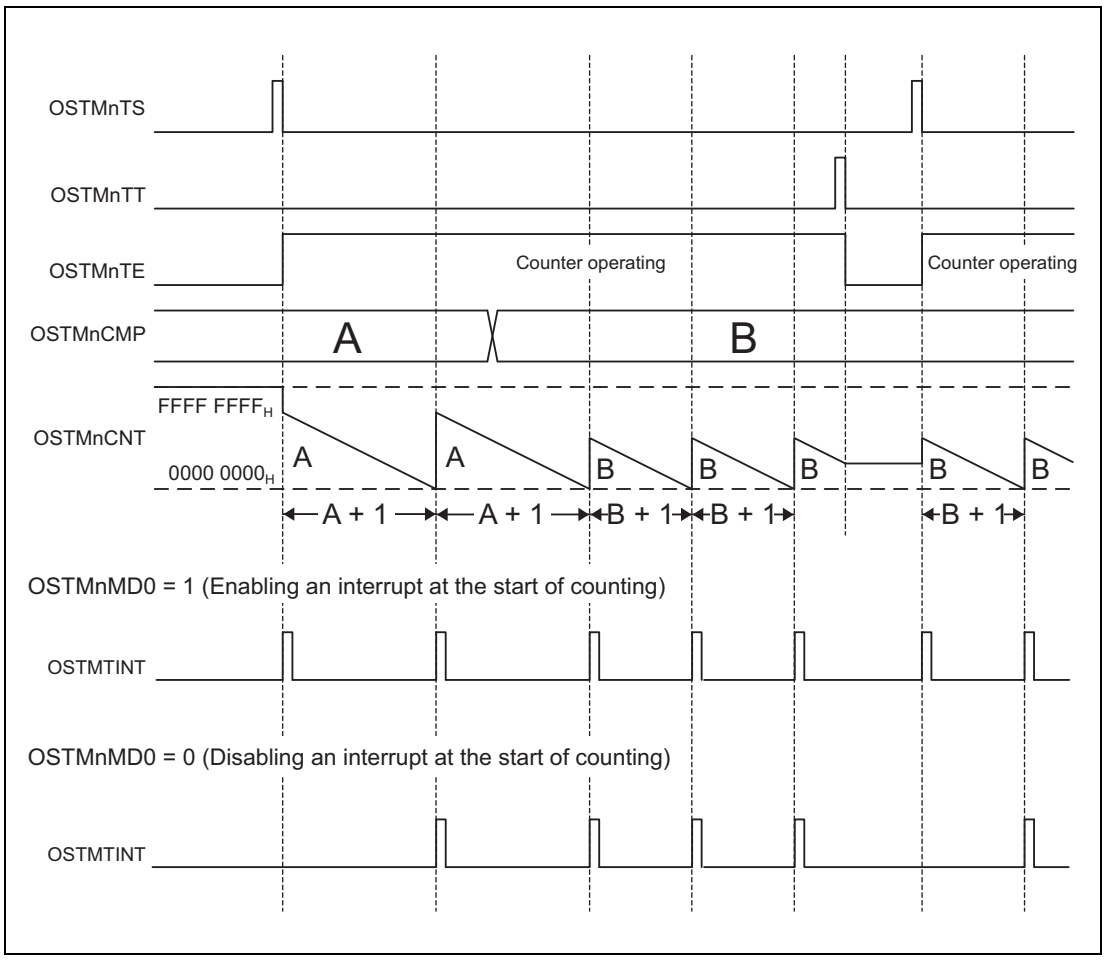


Figure 21.2 Generating an Interrupt when Counting Starts (in Interval Timer Mode)

21.3 Registers

21.3.1 List of Registers

OSTMn registers are listed in the following table.

For details about <OSTMn_base>, see **Section 21.1.2, Register Base Address**.

Table 21.8 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

21.3.2 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 operation mode.

Access: This register can be read/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

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

Table 21.9 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

21.3.3 OSTMnCNT — OSTMn Counter Register

This register indicates the counter value of the timer.

Access: This register can only 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 21.10 OSTMnCNT Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCNT [31:0]	Timer counter value

Table 21.11 lists the correspondence among the operating mode, counting direction, and start value. The start value indicates the value to be read after the operating mode is changed.

Table 21.11 Correspondence among 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.

21.3.4 OSTMnTE — OSTMn Count Enable Status Register

This register indicates whether the counter is enabled or disabled.

Access: This register can only 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 21.12 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.OSTMnTTF to 1 resets this bit to 0.

NOTE

If the counter is disabled, the counter value OSTMnCNT retains its value.

If the counter is restarted, it

- 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.

21.3.5 OSTMnTS — OSTMn Count Start Trigger Register

This register starts the counter.

Access: This register can only 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 21.13 OSTMnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	OSTMnTS	Starts the counter: 0: This setting disables the counter. 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.

21.3.6 OSTMnTT — OSTMn Count Stop Trigger Register

This register stops the counter.

Access: This register can only 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 21.14 OSTMnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	OSTMnTT	Stops the counter: 0: This setting disables the counter. 1: Stops the counter and clears the OSTMnTE.OSTMnTE bit.

21.3.7 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/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 21.15 OSTMnTE Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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.

21.3.8 OSTMnEMU — OSTMn Emulation Register

This register controls operation in combination with SVSTOP.

Access: This register can be read/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: 0000_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 21.16 OSTMnEMU Register Contents

Bit Position	Bit Name	Function
7	OSTMnSVSDIS	When EPC.SVSTOP = 0 Supply of the counter clock is continued when the debugger acquires control of the microcontroller (at breakpoints and so on) regardless of the value of this bit. When EPC.SVSTOP = 1 0: The counter clock is stopped when the debugger acquires control of the microcontroller (at breakpoints and so on). 1: Supply of the counter 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 to these bits, write the value after reset.

21.4 Operation

21.4.1 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 21.4.2, Interval Timer Mode** and **Section 21.4.3, Free-Run Compare Mode**.

Stopping the timer

Setting the OSTMnTT.OSTMnTT bit to 1 stops OSTM.

This also clears the OSTMnTE.OSTMnTE status bit.

21.4.2 Interval Timer Mode

In interval timer mode, OSTM can be used as a reference timer generating interrupt requests at fixed intervals.

21.4.2.1 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.

New values can be written to the OSTMnCMP register at any time. If it is rewritten during count operation, the counter loads the new OSTMnCMP value when the next 0000 0000_H is reached. Then the counter continues with the new value.

OSTMTINT period

The periods of OSTMTINT is:

- OSTMTINT generation period = counter clock period × (OSTMnCMP + 1)

The following figure shows the basic operation of OSTM when counter-start interrupts is enabled in interval timer mode.

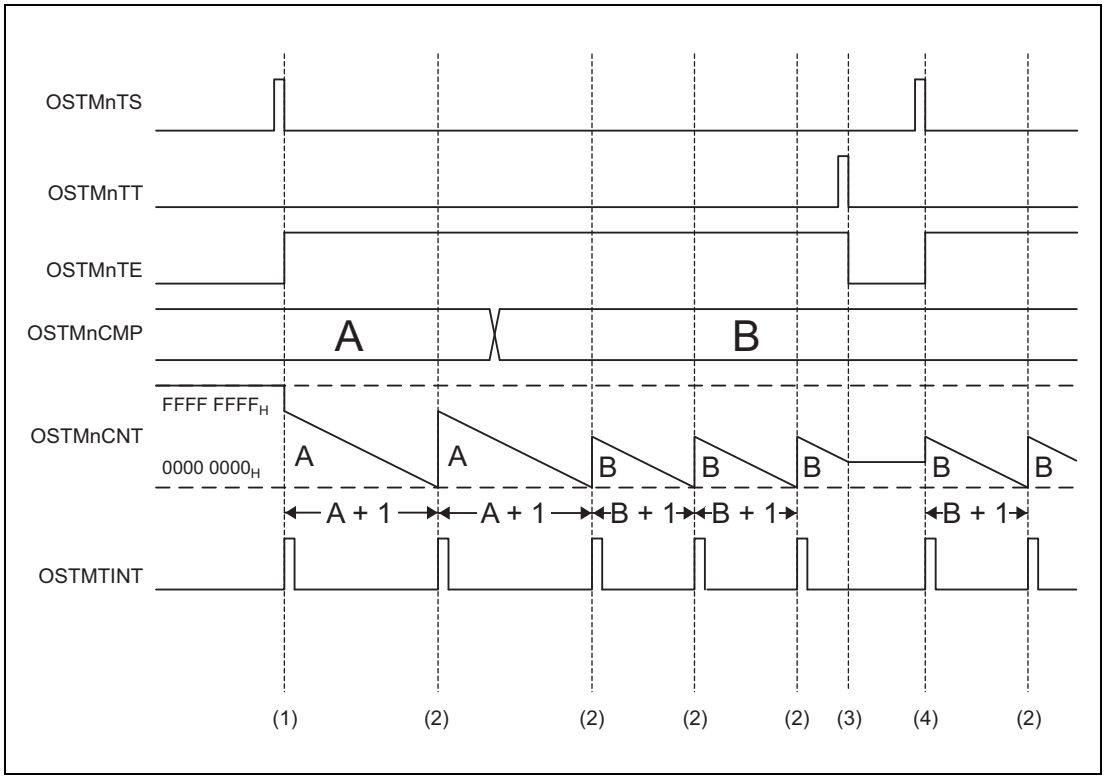


Figure 21.3 Timing Diagram of OSTM in Interval Timer Mode

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 is 1, OSTMTINT interrupt requests are generated at the start of counting. The OSTMnCNT register contains the current value as the counter.

(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).

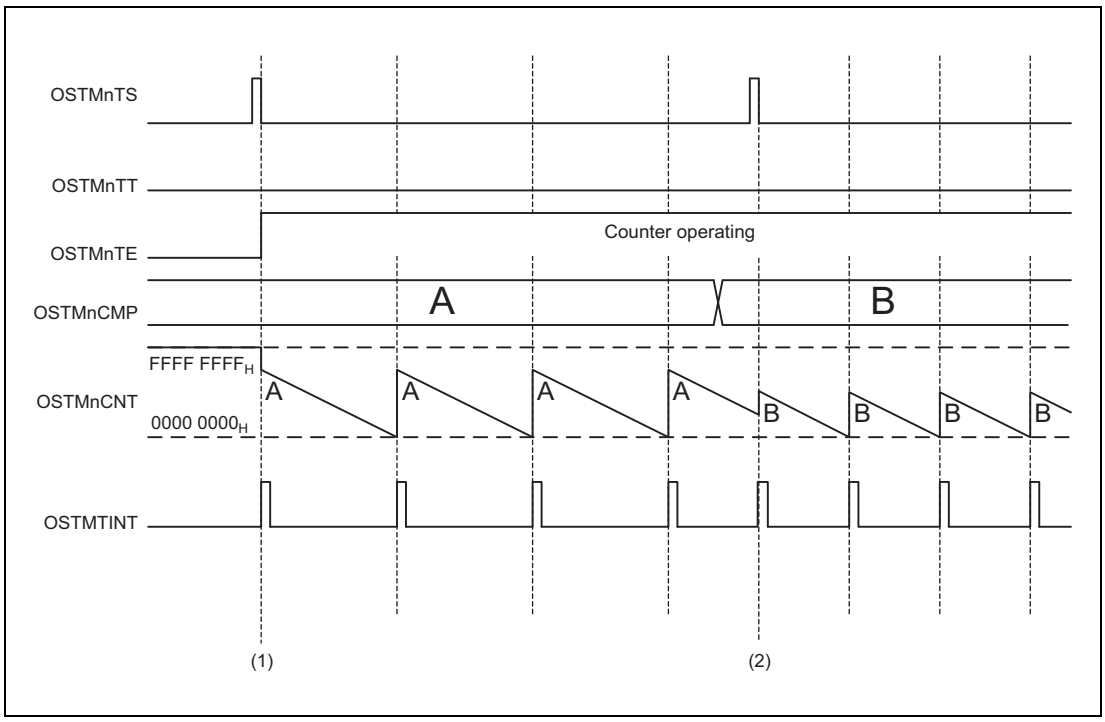


Figure 21.4 Timing Diagram of Forced Restart in Interval Timer Mode

Operations shown in the above timing diagram are as follows.

(1) The counter is started and stopped as described under **Figure 21.3, Timing Diagram of OSTM in Interval Timer Mode**.

(2) Setting OSTMnTS.OSTMnTS = 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.

21.4.2.2 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.

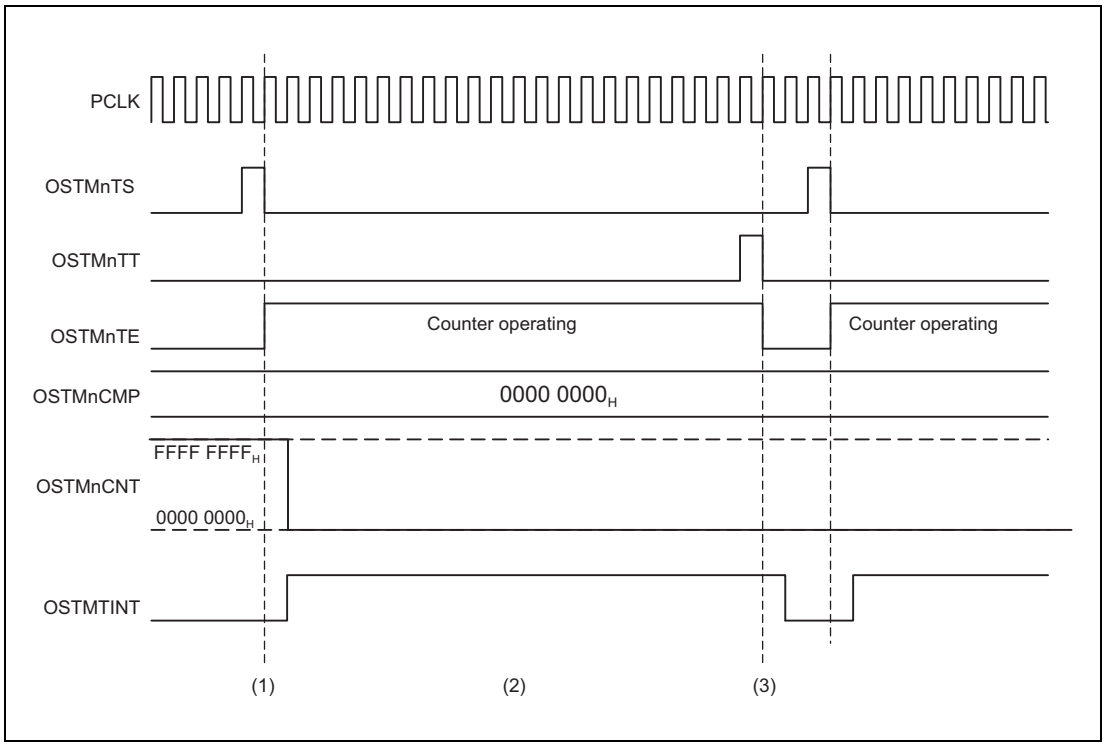


Figure 21.5 Timing Diagram when OSTMnCMP = 0000 0000$_H$ in Interval Timer Mode

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.

21.4.2.3 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 clearing the OSTMnCTL.OSTMnMD1 bit.

(3) Enable or disable interrupts when counting starts (OSTMnCTL.OSTMnMD0).

21.4.3 Free-Run Compare Mode

21.4.3.1 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).

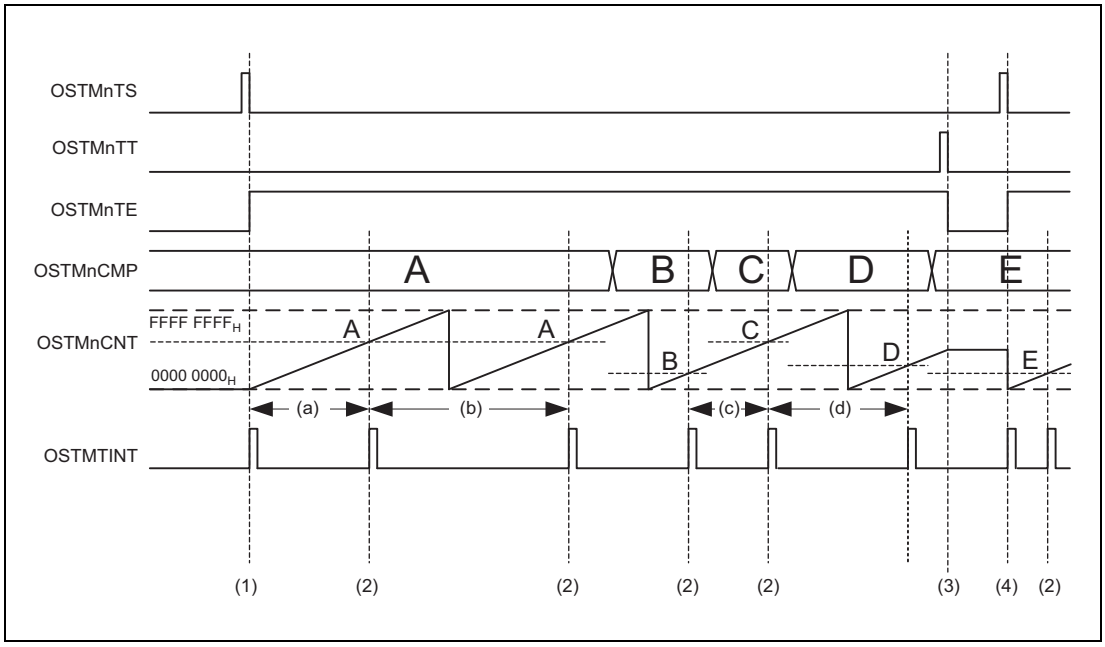


Figure 21.6 Timing Diagram of OSTM in Free-Run Compare Mode

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 is the counter, so it contains the current 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 at the start of counting and depends on the old and new compare values if OSTMnCMP is rewritten during operation.

Table 21.17 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) × counter clock period	(a)
A	A	No rewriting	(FFFF FFFF$_H$ + 1) × counter clock period	(b)
B	C > B	B < counter value < C	(C – B) × counter clock period	(c)
C	D < C	Counter value > D,C	(FFFF FFFF$_H$ – C + D + 1) × counter clock period	(d)

Forced restart

Forced restarting does not proceed during counting even if the OSTMnTS.OSTMnTS bit is set. The counter ignores the attempted setting and continues counting.

21.4.3.2 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).

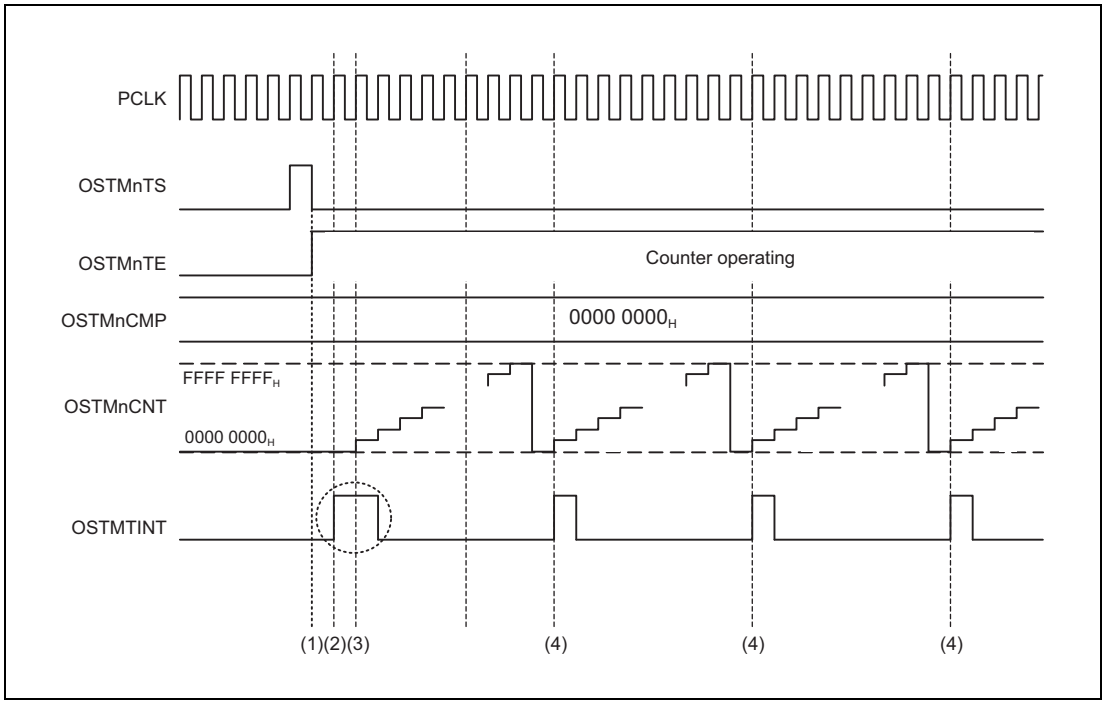


Figure 21.7 Timing Diagram when OSTMnCMP = 0000 0000$_H$ in Free-Run Compare Mode

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$_H$ in the above case, OSTMTINT is generated over two clock cycles.

(4) Every FFFF $FFFF_H$ clock cycles the OSTMTINT interrupt request is asserted.

When interrupts on starting of the counter are disabled, no interrupt is generated when counting starts.

21.4.3.3 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.

(3) Enable or disable interrupts when counting starts by the OSTMnCTL.OSTMnMD0 bit.

Section 22 Timer Array Unit B (TAUB)

This section contains a generic description of the timer array unit B (TAUB).

The first part in this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the TAUB.

22.1 Features of RH850/F1L TAUB

22.1.1 Number of Units and Channels

This microcontroller has the following number of TAUB units.

Table 22.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	—	—	—	1	1	2
Name	—	—	—	TAUBn (n = 0)	TAUBn (n = 0)	TAUBn (n = 0, 1)

TAUBn has the following number of channels of timers.

Table 22.2 Unit Configurations and Channels

Unit Name (Channel Name) TAUBn	Channels per Unit	RH850/F1L 48 pins (—)	RH850/F1L 64 pins (—)	RH850/F1L 80 pins (—)	RH850/F1L 100 pins (16 ch)	RH850/F1L 144 pins (16 ch)	RH850/F1L 176 pins (32 ch)
TAUB0	16	—	—	—	√	√	√
TAUB1	16	—	—	—	—	—	√

Table 22.3 Index

Index	Meaning
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.

22.1.2 Register Base Addresses

TAUBn base addresses are listed in the following table.

TAUBn register addresses are given as offsets from the base addresses in general.

Table 22.4 Register Base Addresses <TAUBn_base>

Name	Base Address
<TAUB0_base>	FFE3 0000_H
<TAUB1_base>	FFE3 1000_H

22.1.3 Clock Supply

The TAUBn clock supply is shown in the following table.

Table 22.5 TAUBn Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
TAUBn	PCLK	CKSCLK_IPERI2

22.1.4 Interrupt Requests

TAUBn interrupt requests are listed in the following table.

Table 22.6 Interrupt Requests (1/2)

Unit Interrupt Name	Outline	Interrupt Number	DMA Trigger Number
TAUB0			
INTTAUB0I0	Channel 0 interrupt	134	33 (Channels 0 to 7)
INTTAUB0I1	Channel 1 interrupt	135	28 (Channels 8 to 15)
INTTAUB0I2	Channel 2 interrupt	136	34 (Channels 0 to 7)
INTTAUB0I3	Channel 3 interrupt	137	29 (Channels 8 to 15)
INTTAUB0I4	Channel 4 interrupt	138	35 (Channels 0 to 7)
INTTAUB0I5	Channel 5 interrupt	139	30 (Channels 8 to 15)
INTTAUB0I6	Channel 6 interrupt	140	36 (Channels 0 to 7)
INTTAUB0I7	Channel 7 interrupt	141	31 (Channels 8 to 15)
INTTAUB0I8	Channel 8 interrupt	142	32 (Channels 8 to 15)
INTTAUB0I9	Channel 9 interrupt	143	37 (Channels 0 to 7)
INTTAUB0I10	Channel 10 interrupt	144	33 (Channels 8 to 15)
INTTAUB0I11	Channel 11 interrupt	145	38 (Channels 0 to 7)
INTTAUB0I12	Channel 12 interrupt	146	34 (Channels 8 to 15)
INTTAUB0I13	Channel 13 interrupt	147	39 (Channels 0 to 7)
INTTAUB0I14	Channel 14 interrupt	148	35 (Channels 8 to 15)
INTTAUB0I15	Channel 15 interrupt	149	40 (Channels 0 to 7)
TAUB1			
INTTAUB1I0	Channel 0 interrupt	248	52 (Channels 0 to 7)
INTTAUB1I1	Channel 1 interrupt	249	51 (Channels 8 to 15)
INTTAUB1I2	Channel 2 interrupt	250	53 (Channels 0 to 7)
INTTAUB1I3	Channel 3 interrupt	251	52 (Channels 8 to 15)
INTTAUB1I4	Channel 4 interrupt	252	54 (Channels 0 to 7)
INTTAUB1I5	Channel 5 interrupt	253	53 (Channels 8 to 15)
INTTAUB1I6	Channel 6 interrupt	254	55 (Channels 0 to 7)
INTTAUB1I7	Channel 7 interrupt	255	54 (Channels 8 to 15)
INTTAUB1I8	Channel 8 interrupt	256	55 (Channels 8 to 15)
INTTAUB1I9	Channel 9 interrupt	257	56 (Channels 0 to 7)
INTTAUB1I10	Channel 10 interrupt	258	56 (Channels 8 to 15)
INTTAUB1I11	Channel 11 interrupt	259	57 (Channels 0 to 7)
INTTAUB1I12	Channel 12 interrupt	260	57 (Channels 8 to 15)

Table 22.6 Interrupt Requests (2/2)

Unit Interrupt Name	Outline	Interrupt Number	DMA Trigger Number
INTTAUB1I13	Channel 13 interrupt	261	58 (Channels 0 to 7)
INTTAUB1I14	Channel 14 interrupt	262	58 (Channels 8 to 15)
INTTAUB1I15	Channel 15 interrupt	263	59 (Channels 0 to 7)

22.1.5 Reset Sources

TAUBn reset sources are listed in the following table. TAUBn is initialized by these reset sources.

Table 22.7 Reset Sources

Unit Name	Reset Source
TAUBn	All reset sources (ISORES)

22.1.6 External I/O Signals

External input/output signals of TAUBn are listed below.

Table 22.8 External I/O Signals (1/2)

TAUB Signals	Function	Alternative Port Pin Signal
TAUB0		
TAUBTTIN0	Channel 0 input	Port TAUB0I0
TAUBTTIN1	Channel 1 input	Port TAUB0I1
TAUBTTIN2	Channel 2 input	Port TAUB0I2
TAUBTTIN3	Channel 3 input	Port TAUB0I3
TAUBTTIN4	Channel 4 input	Port TAUB0I4
TAUBTTIN5	Channel 5 input	Port TAUB0I5
TAUBTTIN6	Channel 6 input	Port TAUB0I6
TAUBTTIN7	Channel 7 input	Port TAUB0I7
TAUBTTIN8	Channel 8 input	Port TAUB0I8
TAUBTTIN9	Channel 9 input	Port TAUB0I9
TAUBTTIN10	Channel 10 input	Port TAUB0I10
TAUBTTIN11	Channel 11 input	Port TAUB0I11
TAUBTTIN12	Channel 12 input	Port TAUB0I12
TAUBTTIN13	Channel 13 input	Port TAUB0I13
TAUBTTIN14	Channel 14 input	Port TAUB0I14
TAUBTTIN15	Channel 15 input	Port TAUB0I15
TAUBTTOUT0	Channel 0 output	Port TAUB0O0
TAUBTTOUT1	Channel 1 output	Port TAUB0O1
TAUBTTOUT2	Channel 2 output	Port TAUB0O2
TAUBTTOUT3	Channel 3 output	Port TAUB0O3
TAUBTTOUT4	Channel 4 output	Port TAUB0O4
TAUBTTOUT5	Channel 5 output	Port TAUB0O5
TAUBTTOUT6	Channel 6 output	Port TAUB0O6

Table 22.8 External I/O Signals (2/2)

TAUB Signals	Function	Alternative Port Pin Signal
TAUBTTOUT7	Channel 7 output	Port TAUB0O7
TAUBTTOUT8	Channel 8 output	Port TAUB0O8
TAUBTTOUT9	Channel 9 output	Port TAUB0O9
TAUBTTOUT10	Channel 10 output	Port TAUB0O10
TAUBTTOUT11	Channel 11 output	Port TAUB0O11
TAUBTTOUT12	Channel 12 output	Port TAUB0O12
TAUBTTOUT13	Channel 13 output	Port TAUB0O13
TAUBTTOUT14	Channel 14 output	Port TAUB0O14
TAUBTTOUT15	Channel 15 output	Port TAUB0O15
TAUB1		
TAUBTTIN0	Channel 0 input	Port TAUB1I0
TAUBTTIN1	Channel 1 input	Port TAUB1I1
TAUBTTIN2	Channel 2 input	Port TAUB1I2
TAUBTTIN3	Channel 3 input	Port TAUB1I3
TAUBTTIN4	Channel 4 input	Port TAUB1I4
TAUBTTIN5	Channel 5 input	Port TAUB1I5
TAUBTTIN6	Channel 6 input	Port TAUB1I6
TAUBTTIN7	Channel 7 input	Port TAUB1I7
TAUBTTIN8	Channel 8 input	Port TAUB1I8
TAUBTTIN9	Channel 9 input	Port TAUB1I9
TAUBTTIN10	Channel 10 input	Port TAUB1I10
TAUBTTIN11	Channel 11 input	Port TAUB1I11
TAUBTTIN12	Channel 12 input	Port TAUB1I12
TAUBTTIN13	Channel 13 input	Port TAUB1I13
TAUBTTIN14	Channel 14 input	Port TAUB1I15
TAUBTTIN15	Channel 15 input	Port TAUB1I15
TAUBTTOUT0	Channel 0 output	Port TAUB1O0
TAUBTTOUT1	Channel 1 output	Port TAUB1O1
TAUBTTOUT2	Channel 2 output	Port TAUB1O2
TAUBTTOUT3	Channel 3 output	Port TAUB1O3
TAUBTTOUT4	Channel 4 output	Port TAUB1O4
TAUBTTOUT5	Channel 5 output	Port TAUB1O5
TAUBTTOUT6	Channel 6 output	Port TAUB1O6
TAUBTTOUT7	Channel 7 output	Port TAUB1O7
TAUBTTOUT8	Channel 8 output	Port TAUB1O8
TAUBTTOUT9	Channel 9 output	Port TAUB1O9
TAUBTTOUT10	Channel 10 output	Port TAUB1O10
TAUBTTOUT11	Channel 11 output	Port TAUB1O11
TAUBTTOUT12	Channel 12 output	Port TAUB1O12
TAUBTTOUT13	Channel 13 output	Port TAUB1O13
TAUBTTOUT14	Channel 14 output	Port TAUB1O14
TAUBTTOUT15	Channel 15 output	Port TAUB1O15

22.2 Overview

22.2.1 Features Summary

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 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 TAUBnCNTm and a 16-bit data register TAUBnCDRm 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 either independently or in combination with other channels (synchronously); i.e. multiple channels depend on each other with one master and one or more slave channels.

When a channel is operated independently, it can be operated independent of all other channels. The synchronous operation function is implemented using a combination of channel groups (consisted of master and slave channels).

Several rules apply to the settings of channels.

22.2.2 Terms

In this section, the following terms are used:

- Independent channel operation function/synchronous operation channel operation function

  TAUB has 16 channels, and provides an independent channel operation function that individual channels operate independently and a synchronous channel operation function that is implemented using a combination of channels.

  – The independent channel operation function can use any channel independent of all other channels.
  – The synchronous channel operation function is implemented 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

  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)
- INTTAUBnIm from master channel
- TAUBTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- 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 TAUBnTUDSm 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

22.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 22.9 Functional List of TAUB Operations

Operation Function	Setting Example
Independent Channel Operation Functions	**Section 22.12**
Interval Timer Function	**Section 22.12.1**
TAUBTTINm Input Interval Timer Function	**Section 22.12.2**
Clock Divide Function	**Section 22.12.3**
External Event Count Function	**Section 22.12.4**
One-Pulse Output Function	**Section 22.12.5**
TAUBTTINm Input Pulse Interval Measurement Function	**Section 22.12.6**
TAUBTTINm Input Signal Width Measurement Function	**Section 22.12.7**
TAUBTTINm Input Position Detection Function	**Section 22.12.8**
TAUBTTINm Input Period Count Detection Function	**Section 22.12.9**
TAUBTTINm Input Pulse Interval Judgment Function	**Section 22.12.10**
TAUBTTINm Input Signal Width Judgment Function	**Section 22.12.11**
Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)	**Section 22.12.12**
Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)	**Section 22.12.13**
Independent Channel Simultaneous Rewrite Functions	**Section 22.13**
Simultaneous Rewrite Trigger Generation Function Type 1	**Section 22.13.1**
Synchronous Channel Operation Functions	**Section 22.14**
PWM Output Function	**Section 22.14.1**
One-Shot Pulse Output Function	**Section 22.14.2**
Delay Pulse Output Function	**Section 22.14.3**
AD Conversion Trigger Output Function Type 1	**Section 22.14.4**
Triangle PWM Output Function	**Section 22.14.5**
Triangle PWM Output Function with Dead Time	**Section 22.14.6**
AD Conversion Trigger Output Function Type 2	**Section 22.14.7**

22.2.4 Input/Output Interrupt Request Signals

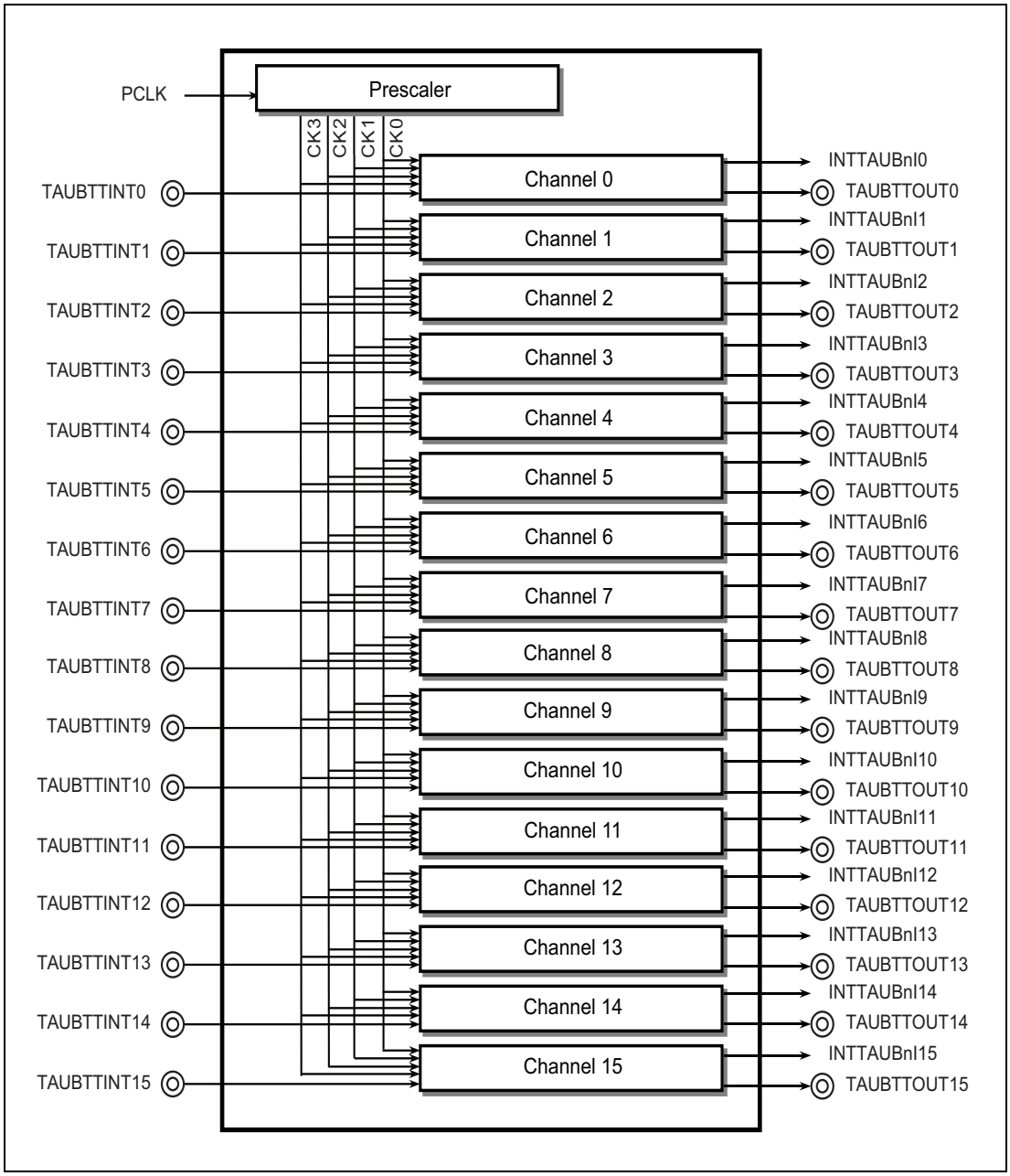


Figure 22.1 TAUB Input/Output and Interrupt Request Signals

22.2.5 Block Diagram

The following figure shows the main components of the TAUB.

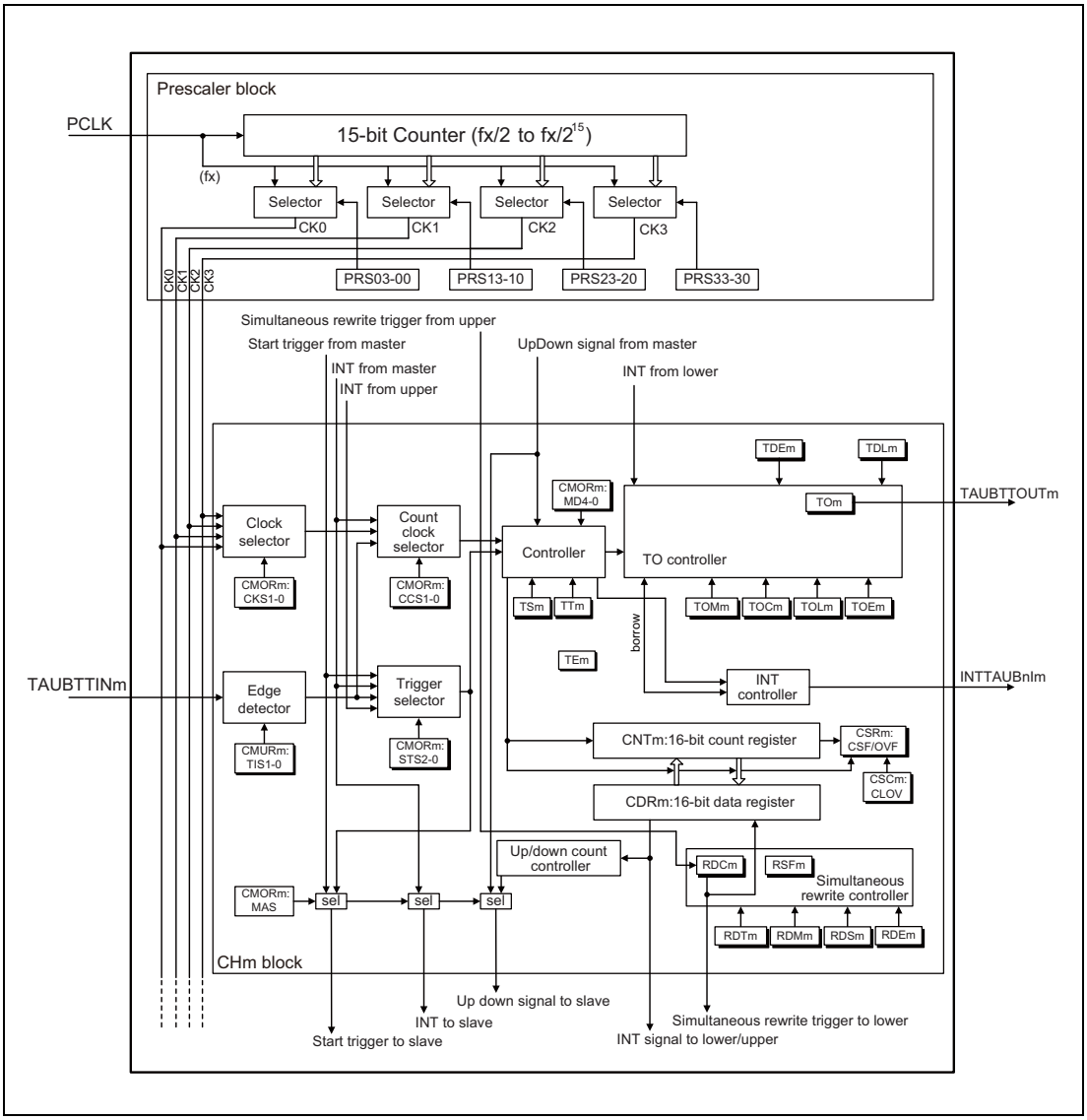


Figure 22.2 Block Diagram of the TAUB

The prefix “TAUBn” has been omitted from the register names for the sake of clarity in the above figure.

22.2.6 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 CK2 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)
- INTTAUBnIm from master channel
- 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 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:

- 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.

22.3 Registers

22.3.1 List of Registers

TAUB registers are listed in the following table.

For details about <TAUBn_base>, see **Section 22.1.2, Register Base Addresses**.

Table 22.10 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> + 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

22.3.2 Details of TAUBn Prescaler Registers

22.3.2.1 TAUBnTPS — TAUBn Prescaler Clock Select Register

This register specifies the PCLK predeclares for clocks CK0, CK1, CK2, and CK3 for all channels.

Access: This register can be read/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 22.11 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 22.11 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 22.11 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 22.1.3, Clock Supply**.

22.3.3 Details of TAUBn Control Registers

22.3.3.1 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: Access This register can be read/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 22.12 TAUBnCDRm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnCDR [15:0]	Data register for the capture/compare value.

22.3.3.2 TAUBnCNTm — TAUBn Channel Counter Register

This register is the channel m counter register.

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

Address: <TAUBn_base> + 80_H + m × 4_H

Value after reset: $FFFF_H$ The initial value after reset depends on the operation mode, see **Table 22.13, TAUBnCNTm Register Contents**.

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 22.13 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 22.14 TAUBnCNTm Read Values after Re-Enabling Counter

		TAUBnCNTm		
Mode Name	Count Method (Up/Down)	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 release

22.3.3.3 TAUBnCMORm — TAUBn Channel Mode OS Register

This register controls channel m operation.

Access: This register can be read/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/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 22.15 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 bit TAUBnCMORm.TAUBnCCS0. **TAUBn CKS1 / TAUBn CKS0 / Selected Operation Clock** 0 / 0 / CK0 0 / 1 / CK1 1 / 0 / CK2 1 / 1 / CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Selects the count clock for TAUBnCNTm counter: **TAUBnCCS0 / Selected Count Clock** 0 / Operation clock as specified by TAUBnCMORm.TAUBnCKS[1:0]. 1 / Valid edge of TAUBTTINm input signal
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.

Table 22.15 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>Valid edge of the TAUBTTINm input signal is the tart trigger and the reverse edge is the stop (capture) trigger</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Reserved (simultaneous rewrite trigger)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INT of the master channel</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>INT of the upper channel (m-1), regardless of the 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, TAUBnCSR.TAUBnOVF is cleared.</td></tr>
<tr><td>0</td><td>1</td><td>Set upon counter overflow and cleared by a TAUBnCSCm.TAUBnCLOV = 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 a TAUBnCSCm.TAUBnCLOV = 1.</td></tr>
</table></td></tr>
<tr><td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing to this bit, write the value after reset.</td></tr>
</table>

Table 22.15 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	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 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

22.3.3.4 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/written in 8-bit units.

Address: <TAUBn_base> + $C0_H$ + m × 4_H

Value after reset: 0000_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 22.16 TAUBnCMURm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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].

22.3.3.5 TAUBnCSRm — TAUBn Channel Status Register

This register indicates the count direction and the overflow status of channel m's counter.

Access: This register can only 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	TAUBnOSF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 22.17 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].

22.3.3.6 TAUBnCSCm — TAUBn Channel Status Clear Register

This register is a trigger register for clearing the overflow flag TAUBnCSRm.TAUBnOVF of a channel m.

Access: This register can only be written in 16-bit units. It is always read as 0000_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 22.18 TAUBnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	TAUBnCLOV	0: No function 1: Clears the overflow flag TAUBnCSRm.TAUBnOVF

22.3.3.7 TAUBnTS — TAUBn Channel Start Trigger Register

This register enables the counter for each channel.

Access: This register can only 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 22.19 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.

22.3.3.8 TAUBnTE — TAUBn Channel Enable Status Register

This register indicates whether counter is enabled or disabled.

Access: This register can only 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 22.20 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 or trigger input detection TAUBnTSSTm = 1 sets this bit to 1. Setting TAUBnTT.TAUBnTTm to 1 resets this bit to 0.

22.3.3.9 TAUBnTT — TAUBn Channel Stop Trigger Register

This register stops the counter for each channel.

Access: This register can only 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 22.21 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 = 0. TAUBnCNTm, TAUBnTO.TAUBnTOm, and TAUBTTOUTm all retain the values they had before the counter was stopped.

22.3.4 Details of TAUBn Simultaneous Rewrite Registers

22.3.4.1 TAUBnRDE — TAUBn Channel Reload Data Enable Register

This register enables and disables simultaneous rewrite of the data register TAUBnCDRm/ TOUBnTOLm.

Access: This register can be read/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 22.22 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

22.3.4.2 TAUBnRDS — TAUBn Channel Reload Data Control Channel Select Register

This register selects the control channel for simultaneous rewrite.

Access: This register can be read/written in 16-bit or 1-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 22.23 TAUBnRDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDSm	Specifies which channel is controlled for the simultaneous rewrite trigger: 0: Master channel 1: Another upper channel

22.3.4.3 TAUBnRDM — TAUBn Channel Reload Data Mode Register

This register selects when the signal that controls simultaneous rewrite is loaded.

Access: This register can be read/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 22.24 TAUBnRDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDMm	Selects when the signal that triggers simultaneous is generated: 0: When the master channel counter starts counting 1: At the top of a triangle wave cycle These bits only apply when TAUBnRDE.TAUBnRDEm = 1 and TAUBnRDS.TAUBnRDSm = 0.

22.3.4.4 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/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 22.25 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.

22.3.4.5 TAUBnRDT — TAUBn Channel Reload Data Trigger Register

This register triggers the simultaneous rewrite enabling state.

Access: This register can only 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 22.26 TAUBnRDT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDTm	Triggers the simultaneous rewrite enabling state: 0: No function. The operation writing 0 is ignored. 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

22.3.4.6 TAUBnRSF — TAUBn Channel Reload Status Register

This flag register indicates that simultaneous rewrite is possible.

Access: This register can only 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 22.27 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).

22.3.5 Details of TAUBn Output Registers

22.3.5.1 TAUBnTOE — TAUBn Channel Output Enable Register

This register enables and disables independent channel output mode controlled by software.

Access: This register can be read/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 22.28 TAUBnTOE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOEm	Enables/disables independent channel output mode: 0: Enables independent channel output mode (controlled by software) 1: Disables independent channel output mode Only TAUBnTOm bits for which timer output of a channel is disabled (TAUBnTOEm = 0) can be written.

22.3.5.2 TAUBnTO — TAUBn Channel Output Register

This register specifies and reads the level of TAUBTTOUTm.

Access: This register can be read/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 22.29 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.

22.3.5.3 TAUBnTOM — TAUBn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read/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 22.30 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

22.3.5.4 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/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 22.31 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: TAUBnTTOUTm toggles when INTTAUBnIm occurs.
	1	Set/reset mode: TAUBnTTOUTm 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: TAUBnTTOUTm set when INT occurs on the master channel and reset when INT occurs on the slave channel.
	1	Synchronous Channel Operation Mode 2: TAUBnTTOUTm set when INTTAUBnIm occurs while the slave channel is counting down and reset when INTTAUBnIm occurs while the slave channel is counting up

22.3.5.5 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/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
	TAUBn TOL15	TAUBn TOL14	TAUBn TOL13	TAUBn TOL12	TAUBn TOL11	TAUBn TOL10	TAUBn TOL09	TAUBn TOL08	TAUBn TOL07	TAUBn TOL06	TAUBn TOL05	TAUBn TOL04	TAUBn TOL03	TAUBn TOL02	TAUBn TOL01	TAUBn 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 22.32 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)

22.3.6 Details of TAUBn Dead Time Output Registers

22.3.6.1 TAUBnTDE — TAUBn Channel Dead Time Output Enable Register

This register enables/disables dead time operation for each channel.

Access: This register can be read/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 22.33 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

22.3.6.2 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/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 22.34 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

22.3.7 TAUBn Emulation Register

22.3.7.1 TAUBnEMU — TAUB Emulation Register

This register controls SVSTOP operations.

Access: This register can be read/written in 8-bit units only when TAUBnTE.TEm = 0. This register can be written only while the counter is stopped (TAUBnTE.TAUBnTEm = 0 and 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 22.35 TTAUBnEMU 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 to these bits, write the value after reset.

22.4 General Operating Procedure

The following lists the general operation procedure for the 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 22.12, Independent Channel Operation Functions**
  - **Section 22.14, Synchronous Channel Operation Functions**
- The function can be changed while the counter is stopped (TAUBnTE.TAUBnTEm=0).

22.5 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 22.5.1, Rules of Synchronous Channel Operation Function**.

Two special features for synchronous channel operation are detailed in the following:

- **Section 22.5.2, Simultaneous Start and Stop of Synchronous Channel Counters**
- **Section 22.6, Simultaneous Rewrite**

22.5.1 Rules of Synchronous Channel Operation Function

Number of masters and slaves

- 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 using the TAUBnCMORm.TAUBnCKS[1:0] bits of the slave and master channel.

The basic concepts of master/slave usage and operation clocks are illustrated in the following figure.

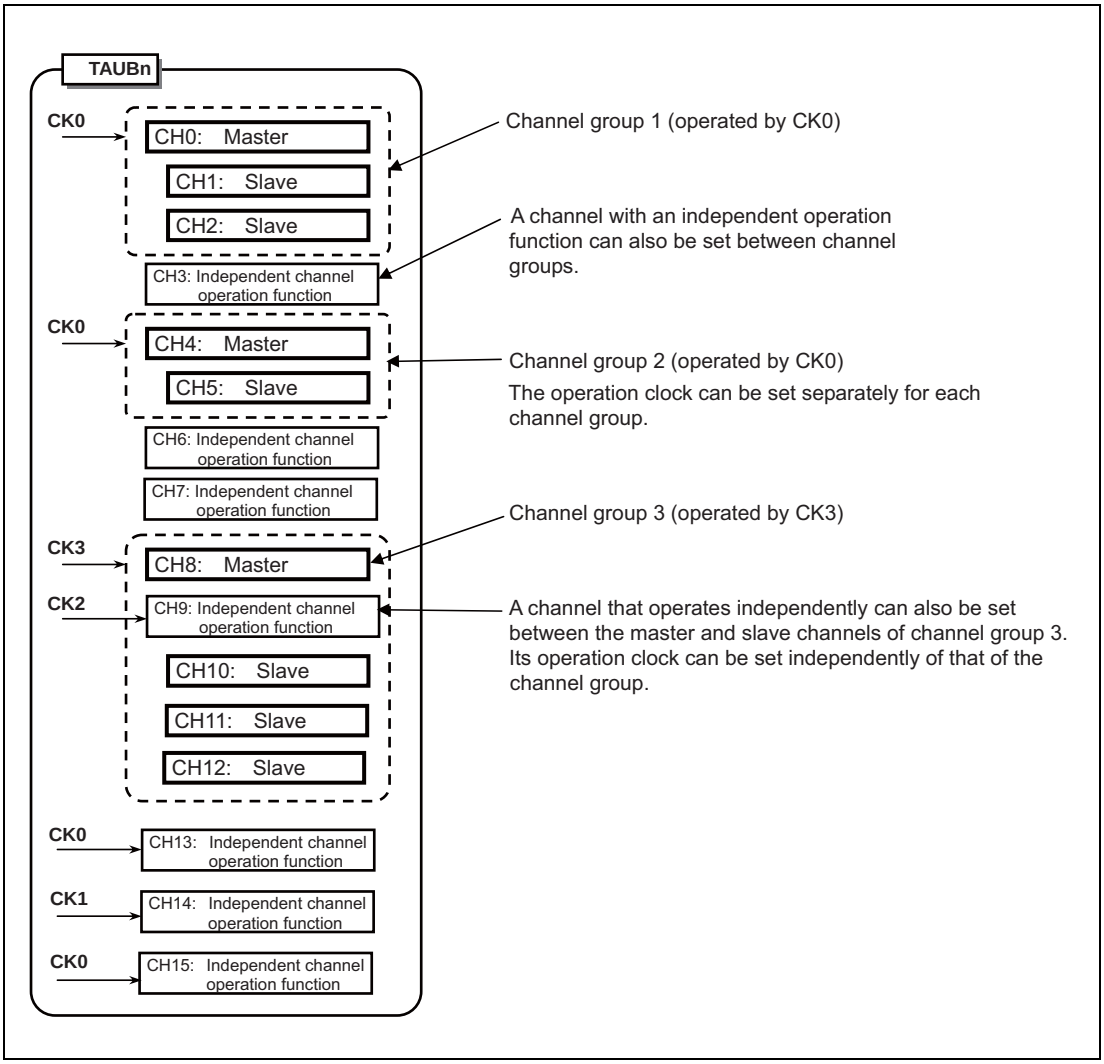


Figure 22.3 Grouping of the Channels and Assignment of Operation Clocks

22.5.2 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.

22.5.2.1 Simultaneous Start and Stop within a TAUB 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.

22.6 Simultaneous Rewrite

22.6.1 Introduction

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 (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 22.36 Simultaneous Rewrite Methods and when They are Triggered

Method	Simultaneous Rewrite Triggered when	TAUBnRDE.TAUBnRDEm	TAUBnRDS.TAUBnRDSm	TAUBnRDM.TAUBnRDMm
—	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 wave of the corresponding slave channel.	1	0	1
C1	INTTAUBnIm is generated on an upper channel specified by TAUBnRDC.TAUBnRDCm	1	1	0

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 22.12, Independent Channel Operation Functions** and **Section 22.14, Synchronous Channel Operation Functions**.

Table 22.37 Simultaneous Rewrite Methods and when They are Triggered

Functions	A	B	C	TAUBnTOL.TAUBnTOLm
Simultaneous Rewrite Trigger Output Function Type 1			√	
PWM Output Function	√		√	√
One-Shot Pulse Output Function	√			
Delay Pulse Output Function	√			
Triangle PWM Output Function		√	√	√
Triangle PWM Output Function with Dead Time		√	√	
AD Conversion Trigger Output Function Type 1	√		√	
AD Conversion Trigger Output Function Type 2		√	√	

Note: √: Available, (Blank): Unavailable

22.6.2 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.

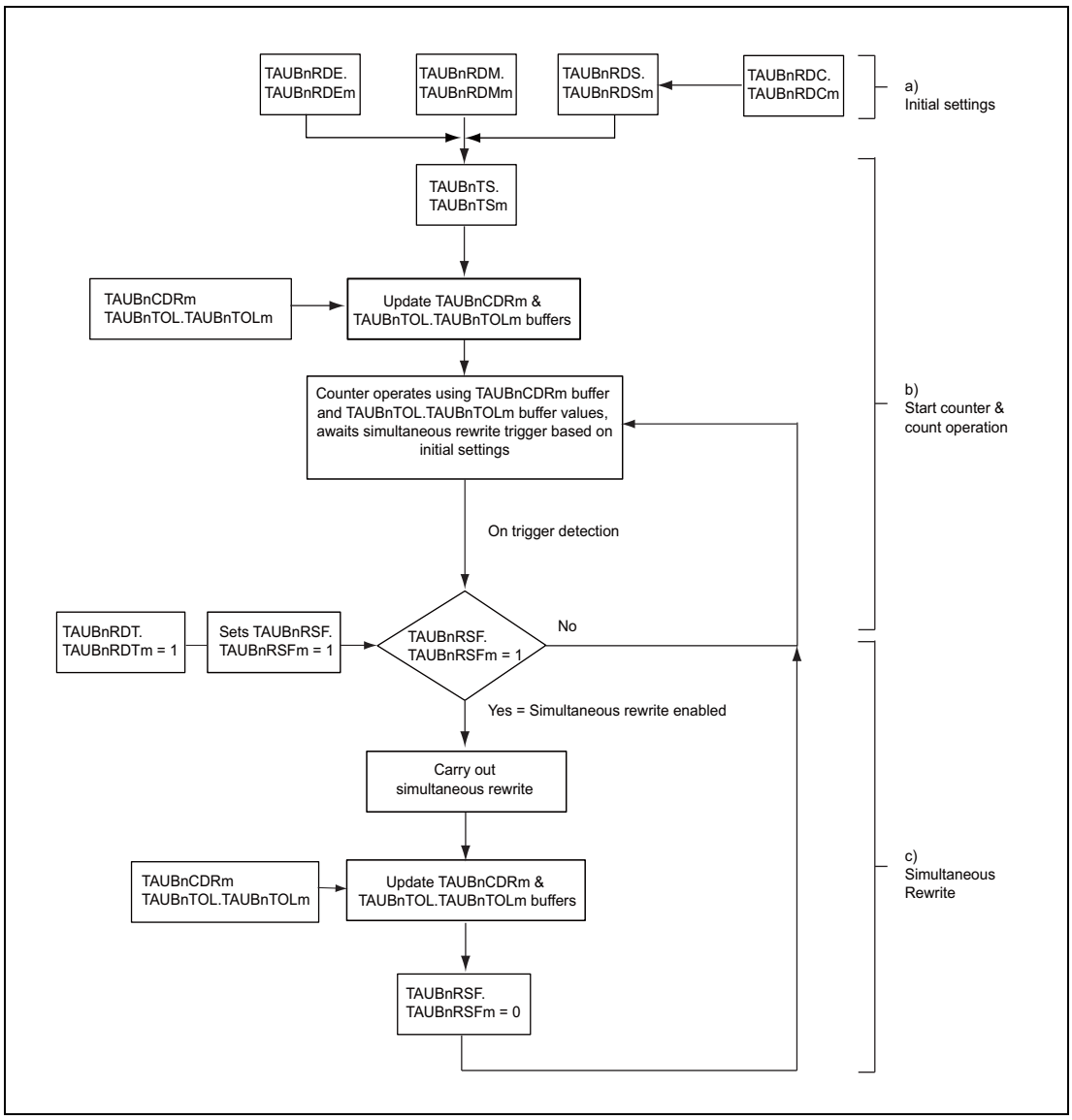


Figure 22.4 General Procedure for Simultaneous Rewrite

22.6.2.1 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 **Section Table 22.36, Simultaneous Rewrite Methods and when They are Triggered**.
- To select which upper channel is monitored for the simultaneous rewrite trigger use TAUBnRDC.TAUBnRDCm (prerequisite: TAUBnRDS.TAUBnRDSm is set to upper channel)

22.6.2.2 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 written 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.RSFm 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 rewire is not carried out, and the system awaits the next simultaneous rewrite trigger detection.

22.6.2.3 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 written 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.

22.6.3 Other General Rules of Simultaneous Rewrite

The following rules also apply:

- TAUBnRDE.RDEm, TAUBnRDS.RDSm, TAUBnRDM.RDMm, and TAUBnRDC.RDCm cannot be changed while the counter is in operation (TAUBnTE.TAUBnTEm = 1).
- TAUBnTOL.TOLm can only be rewritten during operation when in PWM output function or triangle PWM output function. For all other output functions, TAUBnTOL.TOLm must be written before the counter starts. If it is rewritten in another function, TAUBTTOUTm outputs an invalid wave.
- When an upper channel is used as the channel issuing the simultaneous rewrite trigger (TAUBnRDS.TAUBnRDSm = 1), the TAUBnRDC.RDCm 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 (TAUBnRDE.TAUBnRDEm and TAUBnRDS.TAUBnRDSm = 1) but no upper channel is set (TAUBnRDC.TAUBnRDC[15:0] = 0), simultaneous rewrite cannot take place.

22.6.4 Types of Simultaneous Rewrite

In the following section, the three simultaneous rewrite methods are explained using timing diagrams.

22.6.4.1 Simultaneous Rewrite when the Master Channel (Re)Starts Counting (Method A)

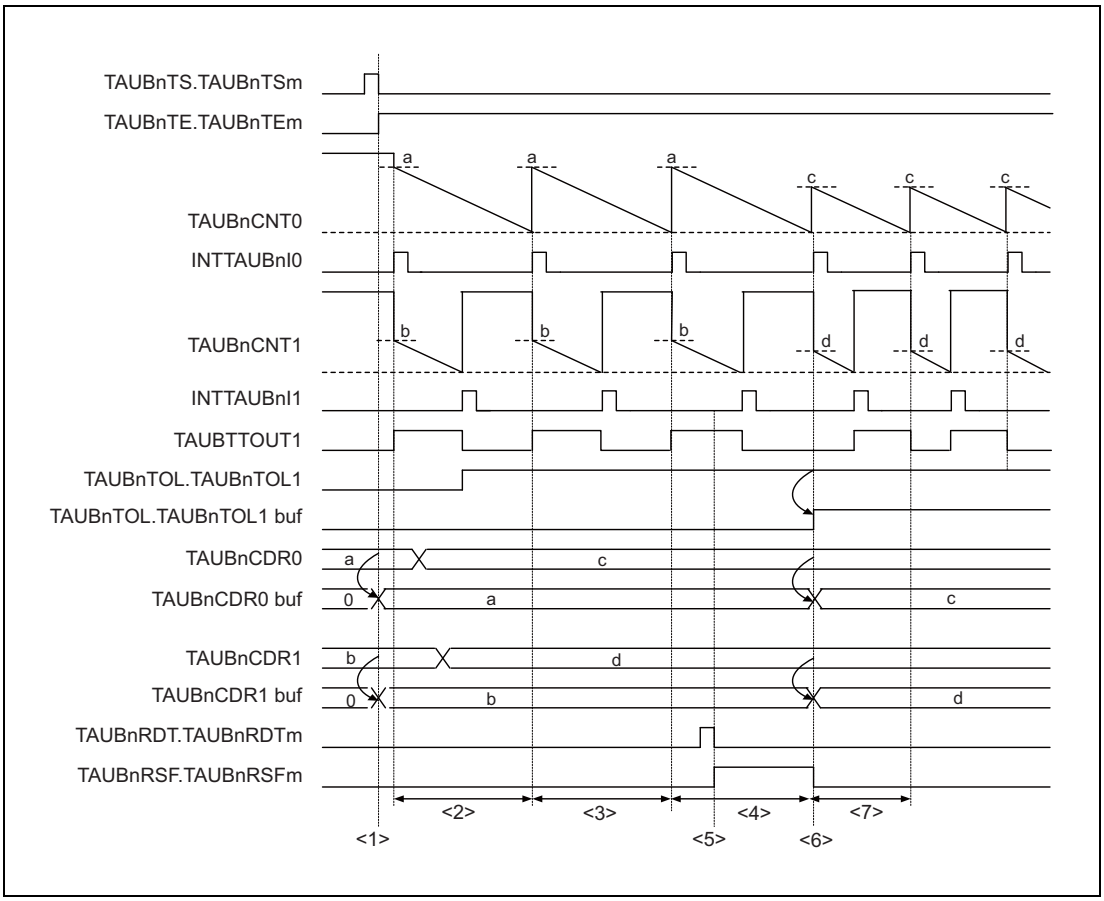


Figure 22.5 Simultaneous Rewrite when the Master Channel (Re)Starts Counting

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.

22.6.4.2 Simultaneous Rewrite at the Peak of a Triangular Wave of the Slave Channel (Method B)

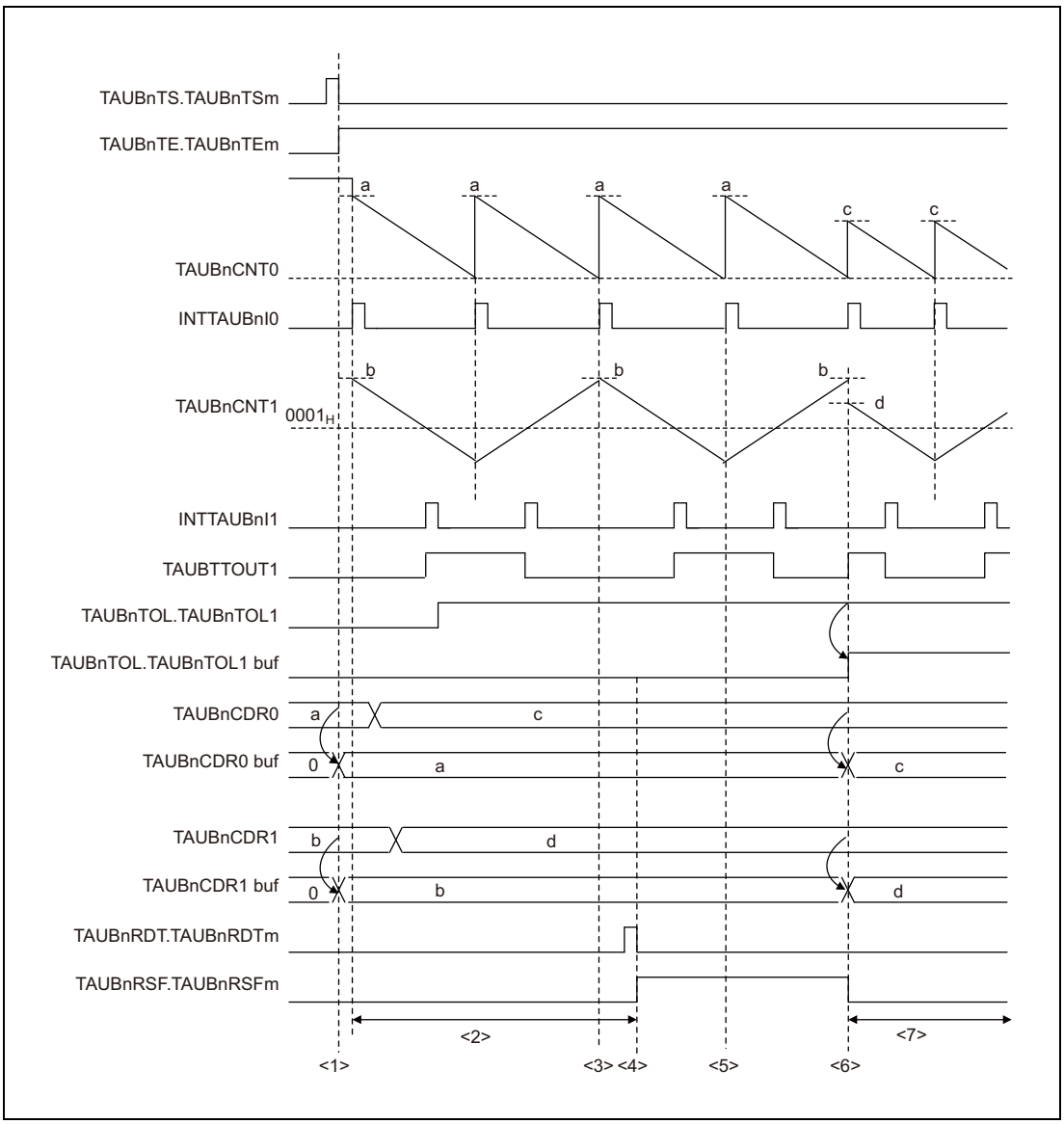


Figure 22.6 Simultaneous Rewrite at the Peak of a Triangular Wave of the Slave Channel

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.

22.6.4.3 Simultaneous Rewrite when INTTAUBnIm is Generated on an Upper Channel Specified by TAUBnRDC.TAUBnRDCm (Method C1)

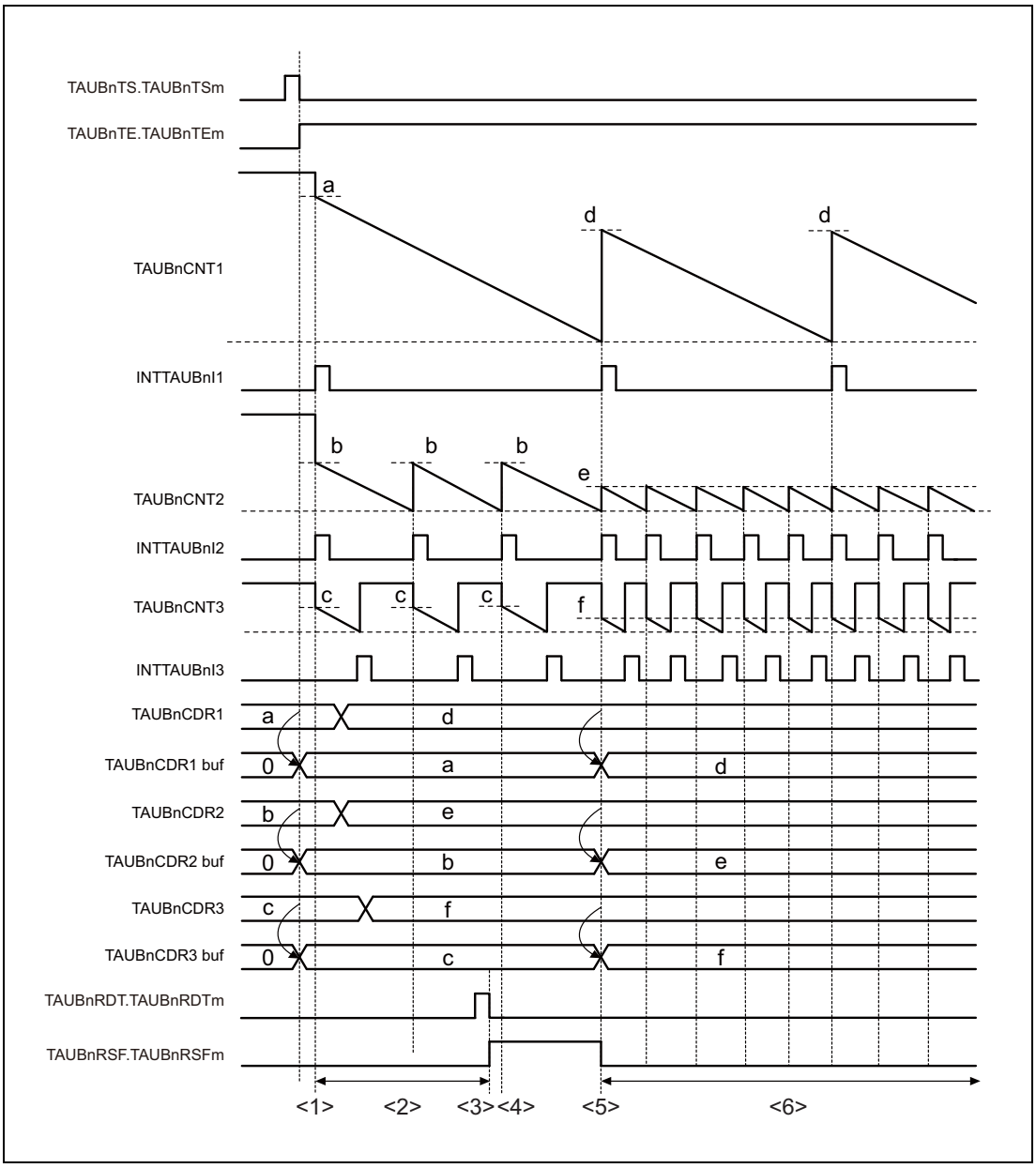


Figure 22.7 Simultaneous Rewrite when INTTAUBnIm is Generated on an Upper Channel Specified by TAUBnRDC.TAUBnRDCm

Setting:

CH1 is an upper channel used counting down, CH2 is a master channel, and CH3 is the slave channel. The simultaneous rewrite method C1 is applied. The TAUBnRDC register specifies a channel which generates simultaneous rewrite triggers.

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.

22.7 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 22.38, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 22.7.2, Channel Output Modes Controlled Independently by TAUBn Signals**
- **Section 22.7.3, 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

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 triangle PWM output function. Otherwise, changes to TAUBnTOL.TAUBnTOLm result in an invalid TAUBTTOUTm signal output.

See **Section 22.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 22.38**.

Table 22.38 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):

- TAUBnTOE.TAUBnTOEm
- TAUBnTOM.TAUBnTOMm
- TAUBnTOC.TAUBnTOCm
- TAUBnTDE.TAUBnTDEm

22.7.1 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 22.38, Channel Output Modes**, and the output logic using the TAUBnTOL.TAUBnTOLm bit.

(3) Start the counter (TAUBnTS.TAUBnTSm = 1).

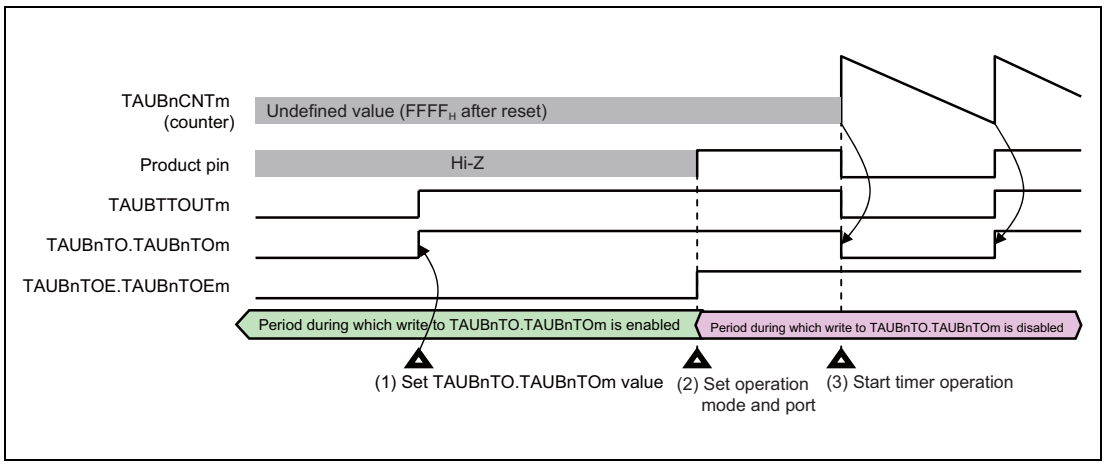


Figure 22.8 General Procedure for Specifying a TAUBTTOUTm Channel Output Mode

22.7.2 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 22.38, Channel Output Modes**.

22.7.2.1 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 22.38, Channel Output Modes**.

22.7.2.2 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 22.38, Channel Output Modes**.

22.7.3 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 22.38, Channel Output Modes**.

22.7.3.1 Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUBnIm of master channel serves as a set signal and INTTAUBnIm of the slave channel as a reset signal. If INTTAUBnIm of 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 master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 22.38, Channel Output Modes**.

22.7.3.2 Synchronous Channel Output Mode 2

In this output mode, the operating mode should be set to count-up/-down mode. The result is a triangle PWM wave at TAUBTTOUTm. For details, see **Section 22.14.5, Triangle 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 triangle PWM output. TAUBTTOUTm should be set to 0 before the function starts.

22.7.3.3 Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUBnTTOUTm. The set/reset conditions are shown in **Figure 22.9**.

Set/reset conditions

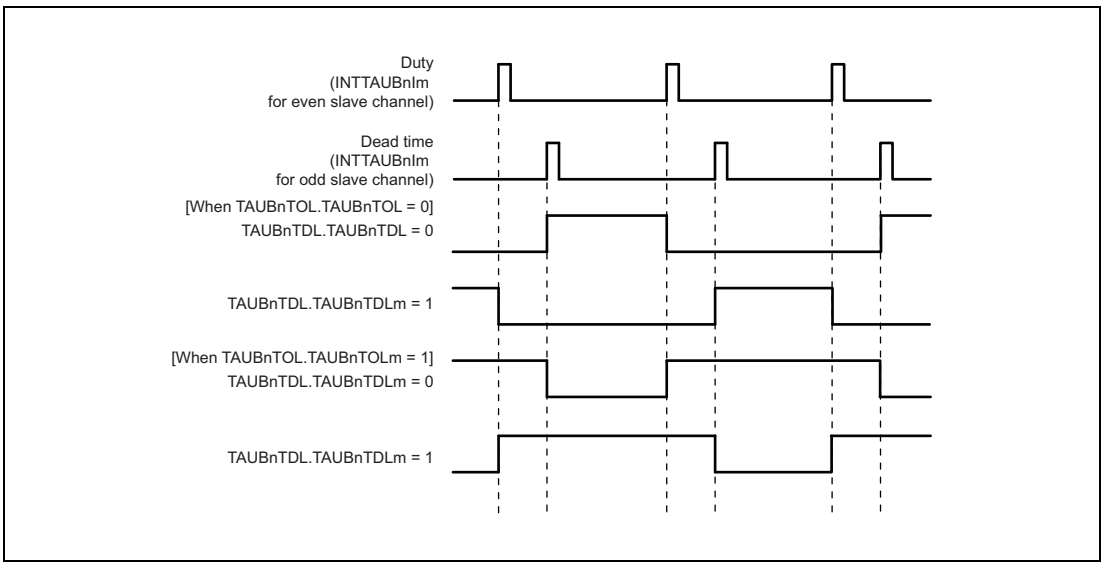


Figure 22.9 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output

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
- TAUBnTDM.TAUBnTDMm

22.8 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.

22.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-Up/-Down Mode, and Count Capture Mode

The counter starts operating with the next count clock after TAUBnTS.TAUBnTSm is set to 1. The value of data register is also loaded when the counter starts.

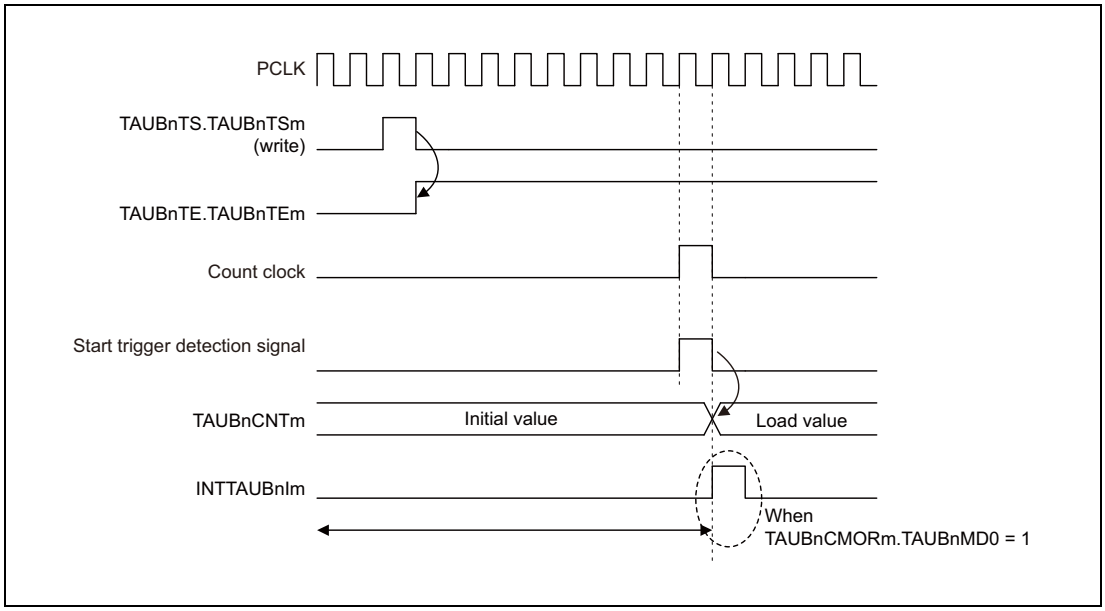


Figure 22.10 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, Up/Down Count Mode, and Count Capture Mode

NOTE

Make sure to set TAUBnCMORm.TAUBnMD0 to 0 when using the count-up/-down mode.

22.8.2 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 increments with subsequent count clocks.

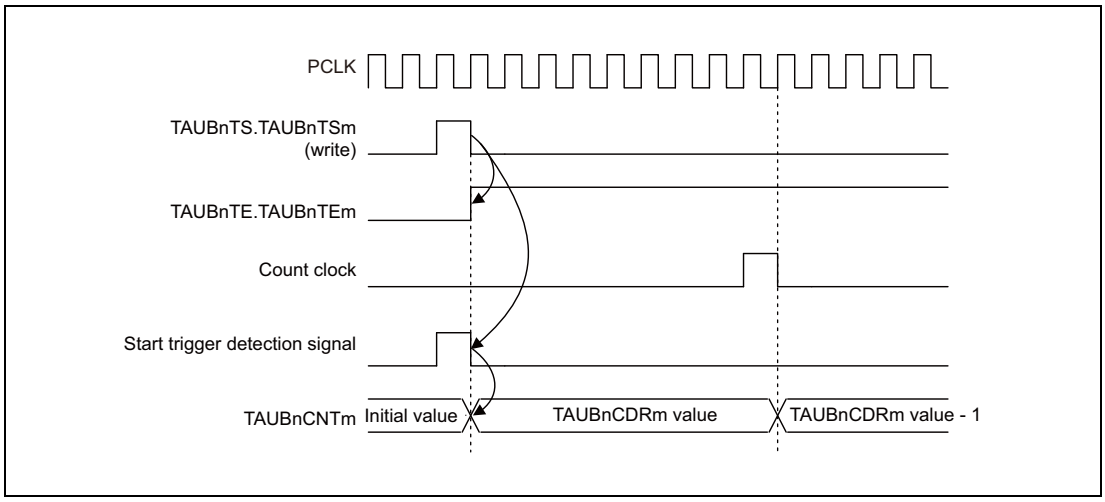


Figure 22.11 Start Timing in Event Count Mode

22.8.3 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 is irrelevant to start of counter operation, determine the frequency with which all operations take place.

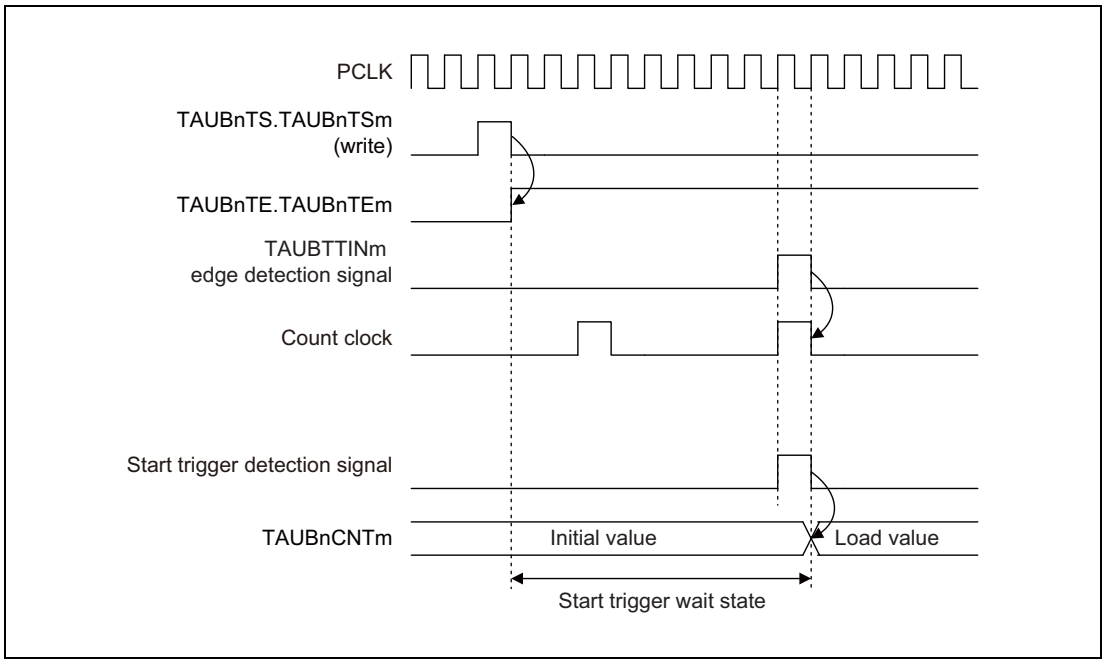


Figure 22.12 Start Timing in Other Operating Modes

22.9 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 INTTAUJnIm 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.

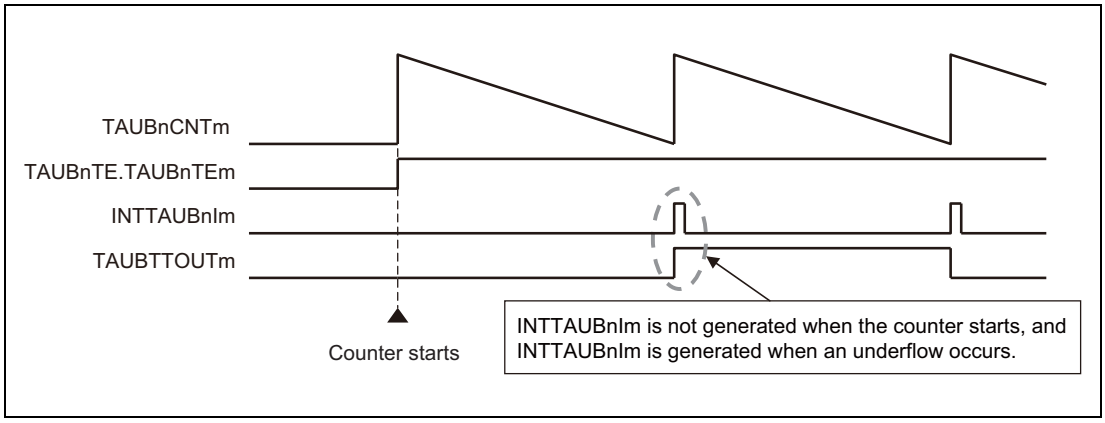


Figure 22.13 INTTAUBnIm Generation Timing (TAUBnCMORm.TAUBnMD0=0)

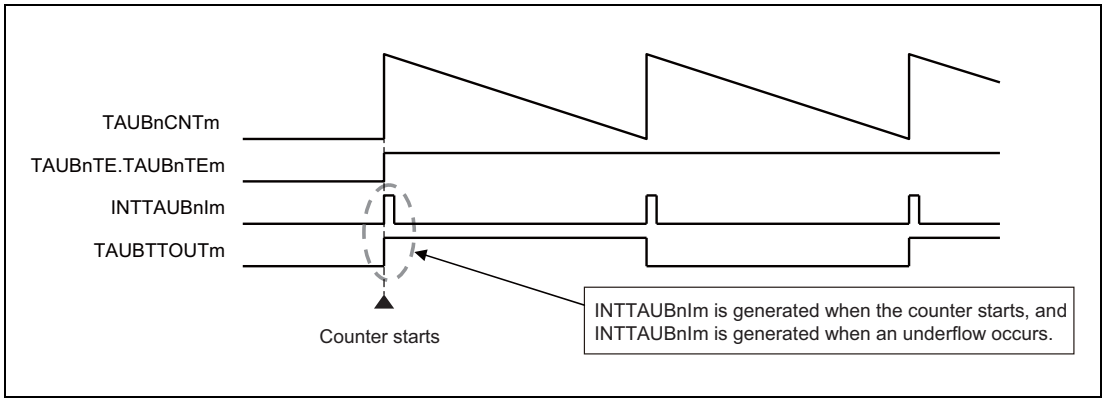


Figure 22.14 INTTAUBnIm Generation Timing (TAUBnCMORm.TAUBnMD0=1)

22.10 Interrupt Generation upon Overflow

Certain independent functions that count up, overflow without generating an interrupt when they reach $FFFF_H$. This section describes how it is possible to generate an interrupt, by combining a channel operating in one of these modes with a channel in a different operation mode which 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 (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.

22.10.1 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 input interval timer function can detect the overflow when TAUBnCNTm of the TAUBTTINm input pulse interval measurement function exceeds $FFFF_H$.

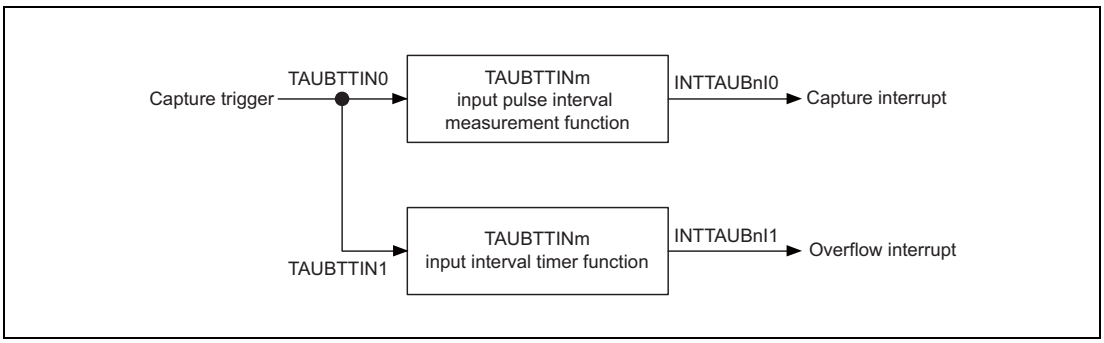


Figure 22.15 Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

Timing diagram

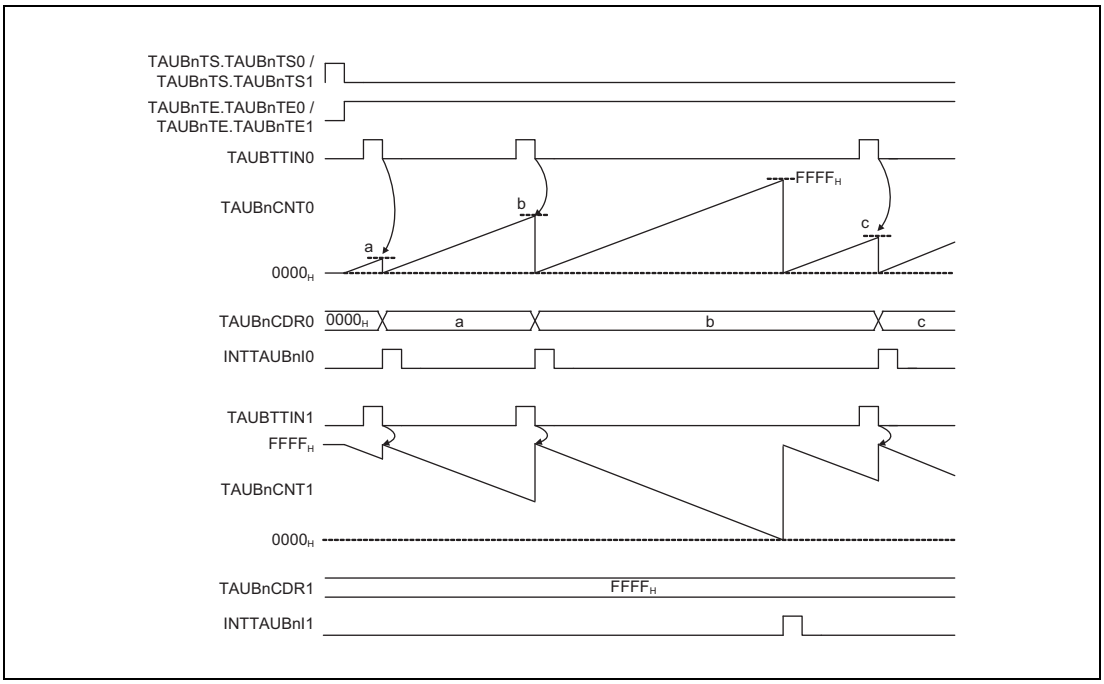


Figure 22.16 Interrupt Generation by Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

22.10.2 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 TAUDTTINm width measurement) can detect the overflow when TAUBnCNTm of the TAUBTTINm input signal width measurement function exceeds $FFFF_H$.

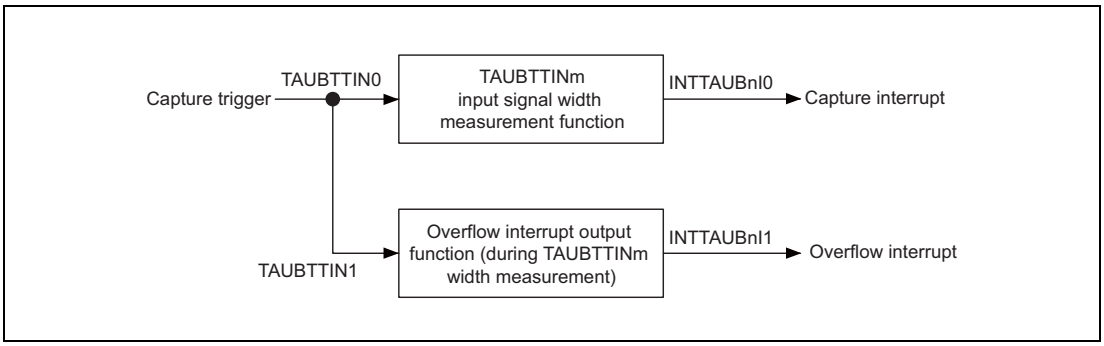


Figure 22.17 Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Timing diagram

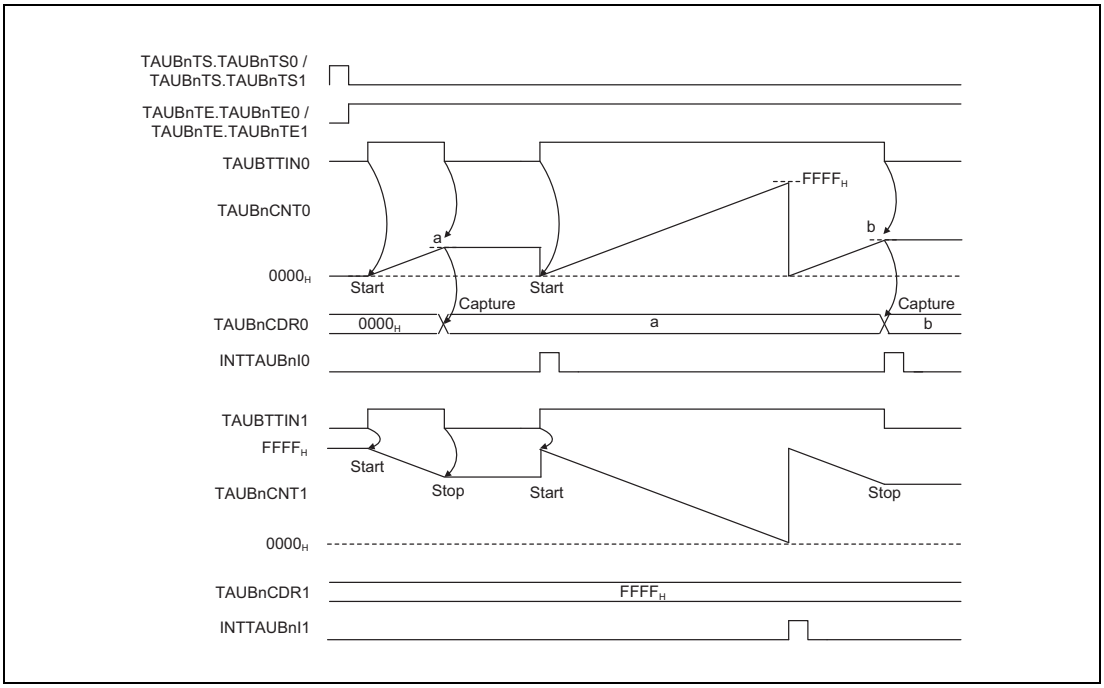


Figure 22.18 Interrupt Generation by Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

22.10.3 Example of Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

When the counters of both channels are started simultaneously, INTTAUBnIm of the interval timer function can detect the overflow when TAUBnCNTm of the TAUBTTINm input position detection function exceeds $FFFF_H$.

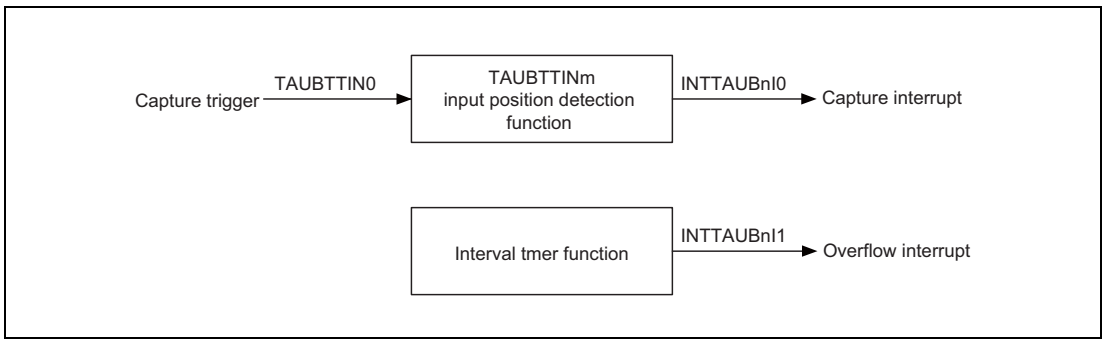


Figure 22.19 Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

Timing diagram

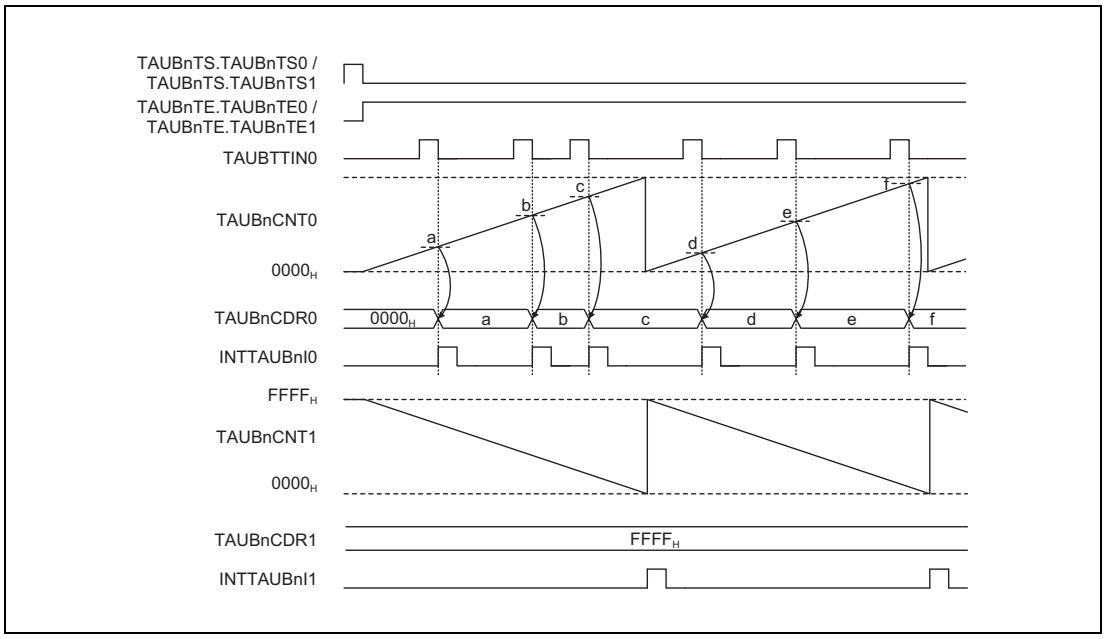


Figure 22.20 Interrupt Generation by Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

22.10.4 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 $FFFF_H$.

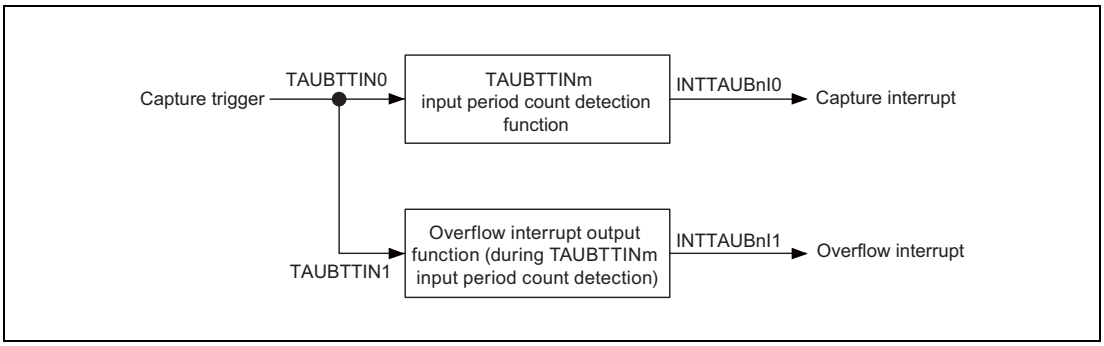


Figure 22.21 Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (TAUBTTINm Input Period Count Detection)

Timing diagram

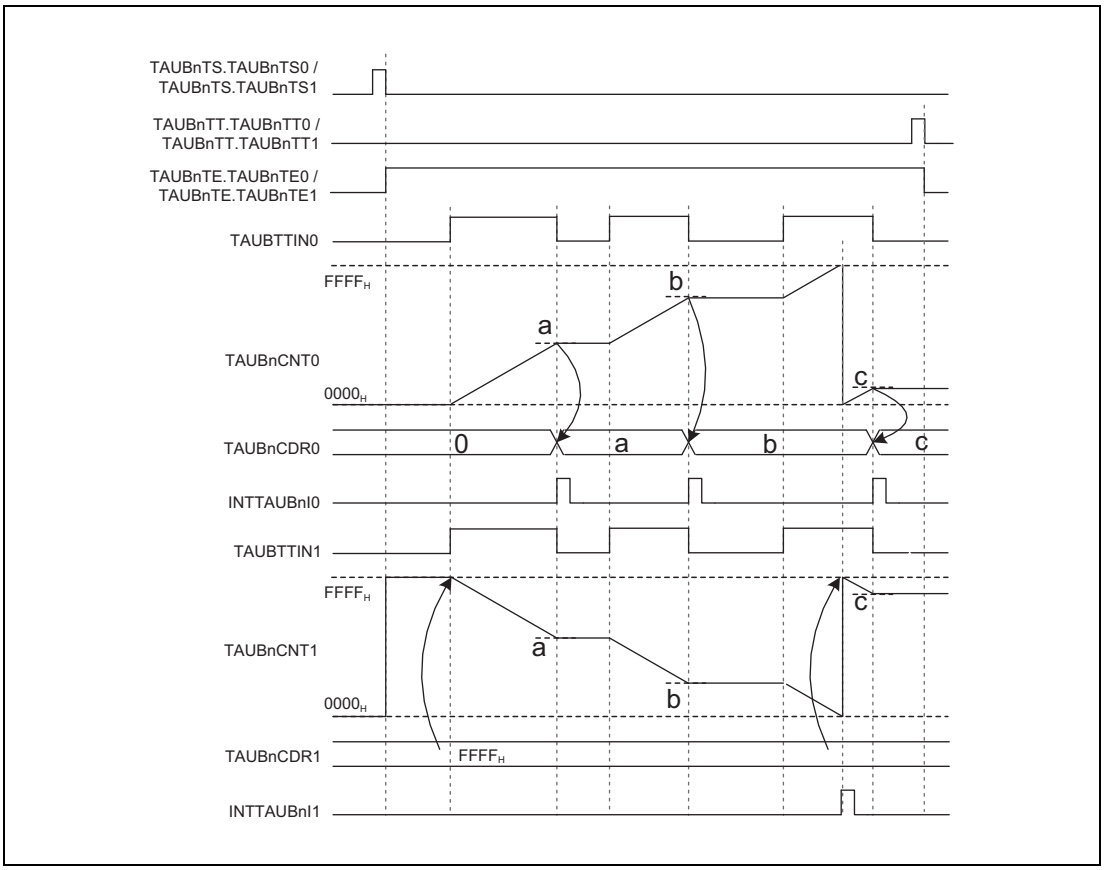


Figure 22.22 Interrupt Generation by Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

22.11 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.

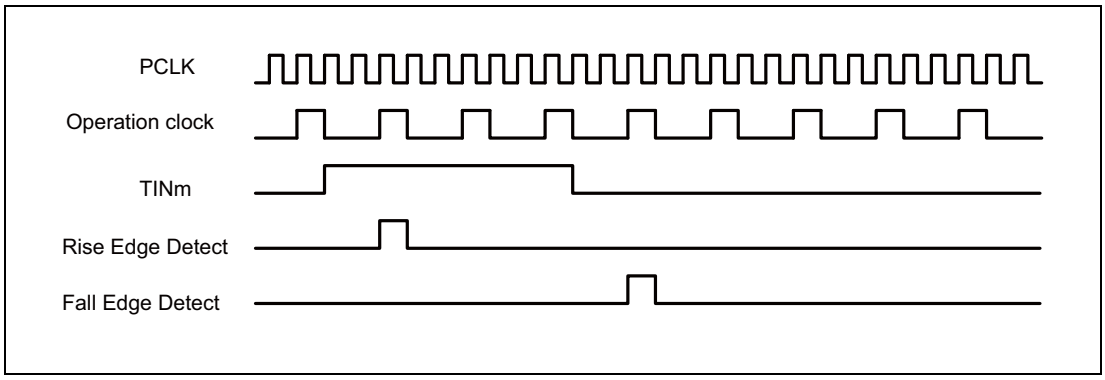


Figure 22.23 Basic Edge Detection Timing

Figure 22.23 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.

22.12 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 22.2, Overview**.

22.12.1 Interval Timer Function

22.12.1.1 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 started 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 reset 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 compared to when TAUBnCMORm.TAUBnMD0 is set to 1.

22.12.1.2 Equations

INTTAUBnIm cycle = count clock cycle × (TAUBnCDRm + 1)

TAUBTTOUTm square wave cycle = count clock cycle × (TAUBnCDRm + 1) × 2

22.12.1.3 Block Diagram and General Timing Diagram

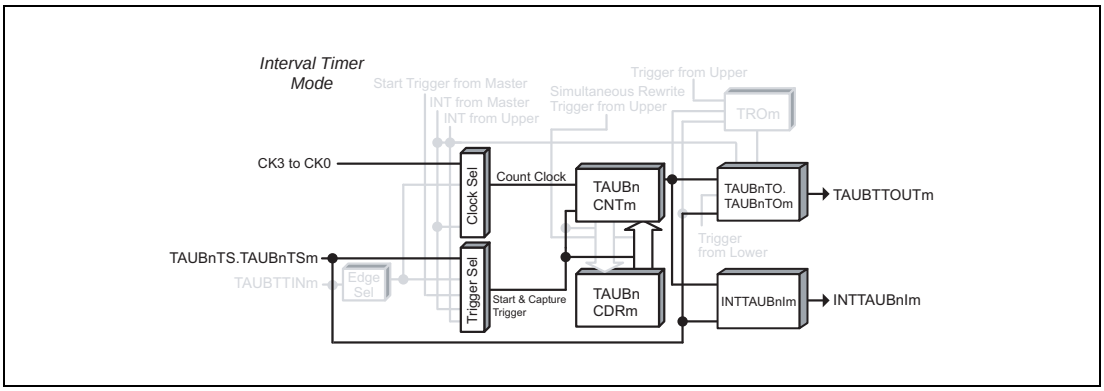


Figure 22.24 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

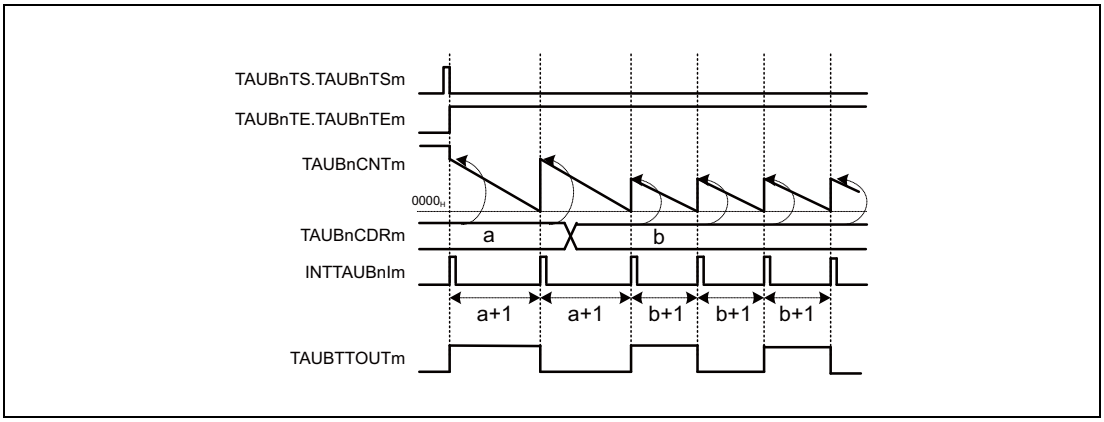


Figure 22.25 General Timing Diagram for Interval Timer Function

22.12.1.4 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 22.39 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 to this bit, 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 to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: Generates INTTAUBnIm and toggles TAUBTTOUTm at operation start. 1: INTTAUBnIm not generated and TAUBTTOUTm does not toggle 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 22.40 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 to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode

Table 22.41 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 22.42 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	

22.12.1.5 Operating Procedure for Interval Timer Function

Table 22.43 Operating Procedure for Interval Timer Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.39, Contents of the TAUBnCMORm Register for Interval Timer Function** and **Table 22.40, 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 22.41, 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

22.12.1.6 Specific Timing Diagrams

(1) TAUBnCDRm = 0000_H, count clock = PCLK/2

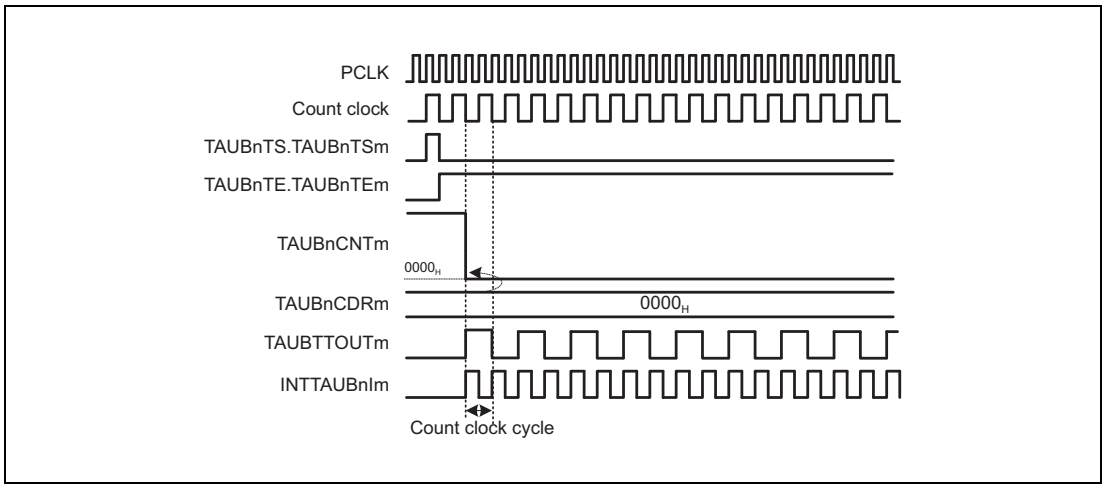


Figure 22.26 TAUBnCDRm = 0000_H, Count Clock = PCLK/2

- 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

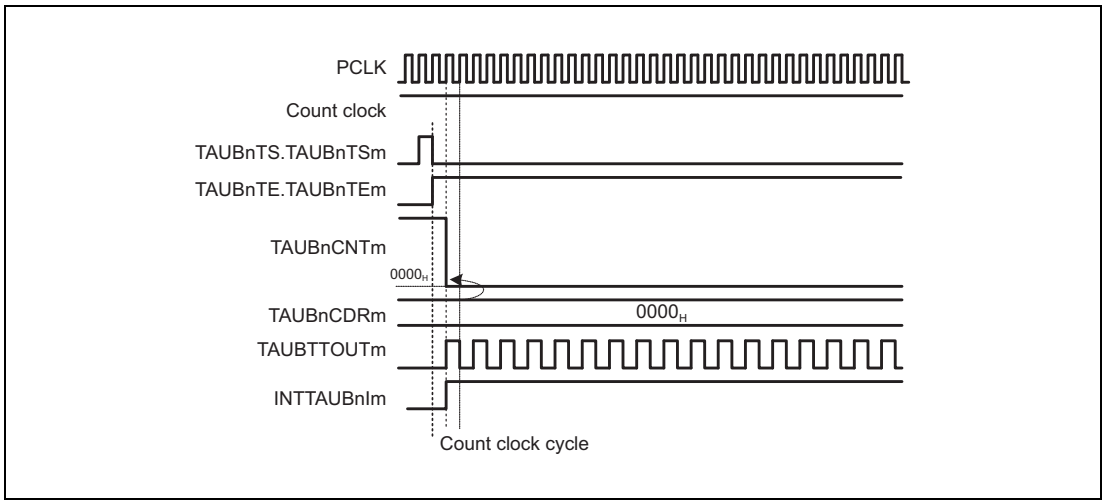


Figure 22.27 TAUBnCDRm = 0000_H, Count Clock = PCLK

- 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.
- INTTAUBnlm is fixed to the high level, resulting in TAUBTTOUTm toggling every PCLK clock.

(3) Operation stop and restart (TAUBnCMORm TAUBnMDO = 1)

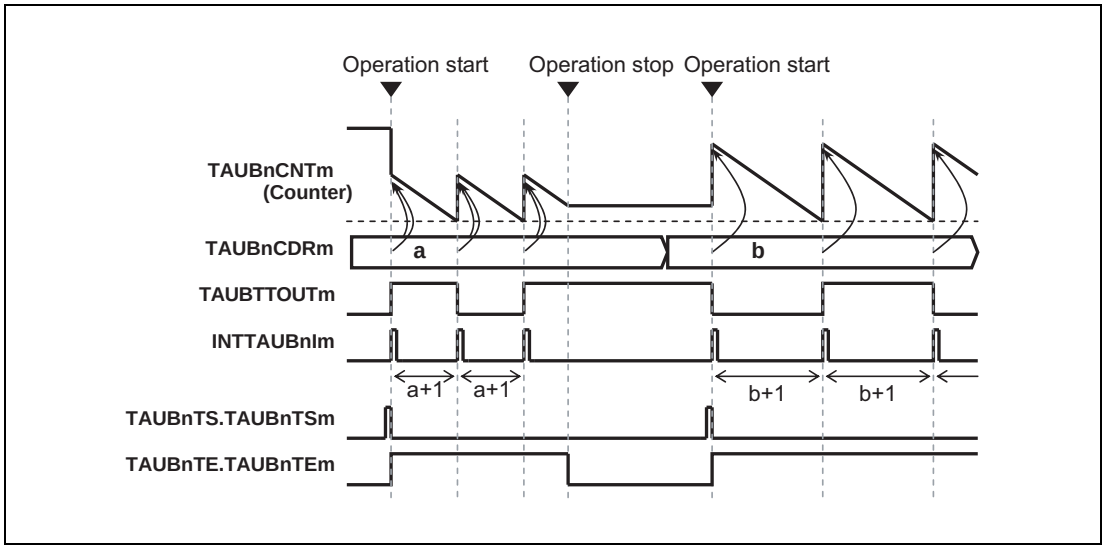


Figure 22.28 Operation Stop and Restart, TAUBnCMORm.TAUBnMD0 = 1

- 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)

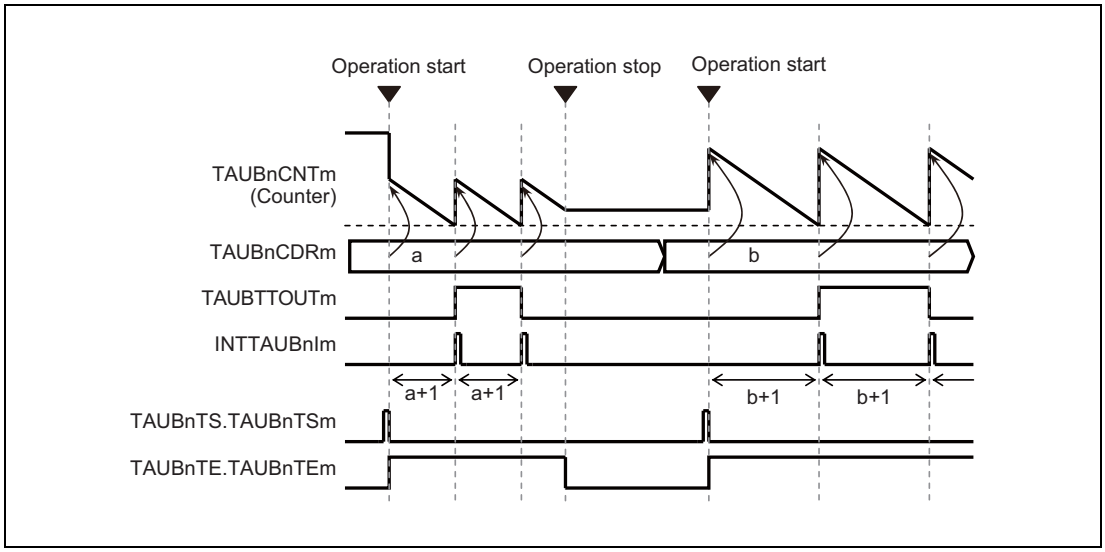


Figure 22.29 Operation Stop and Restart, TAUBnCMORm.TAUBnMD0 = 0

(5) Forced restart (TAUBnCMORm.TAUBnMD0 = 1)

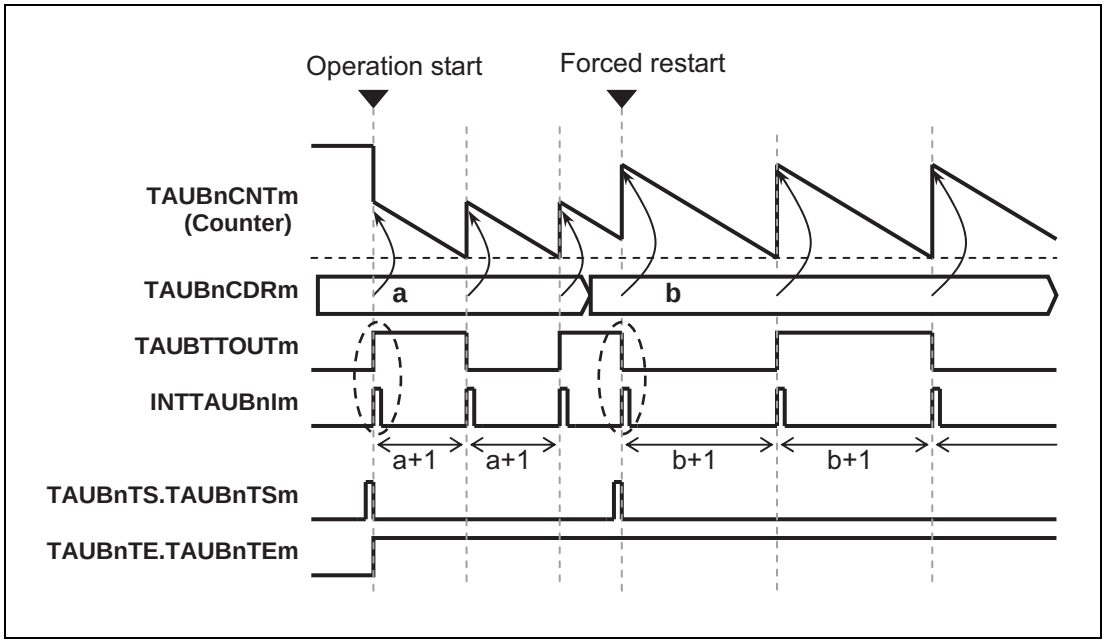


Figure 22.30 Forced Restart Operation, TAUBnCMORm.TAUBnMD0 = 1

- 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 at start or restart is generated and the output TAUBTTOUTm toggles.

(6) Forced restart (TAUBnCMORm.TAUBnMD0 = 0)

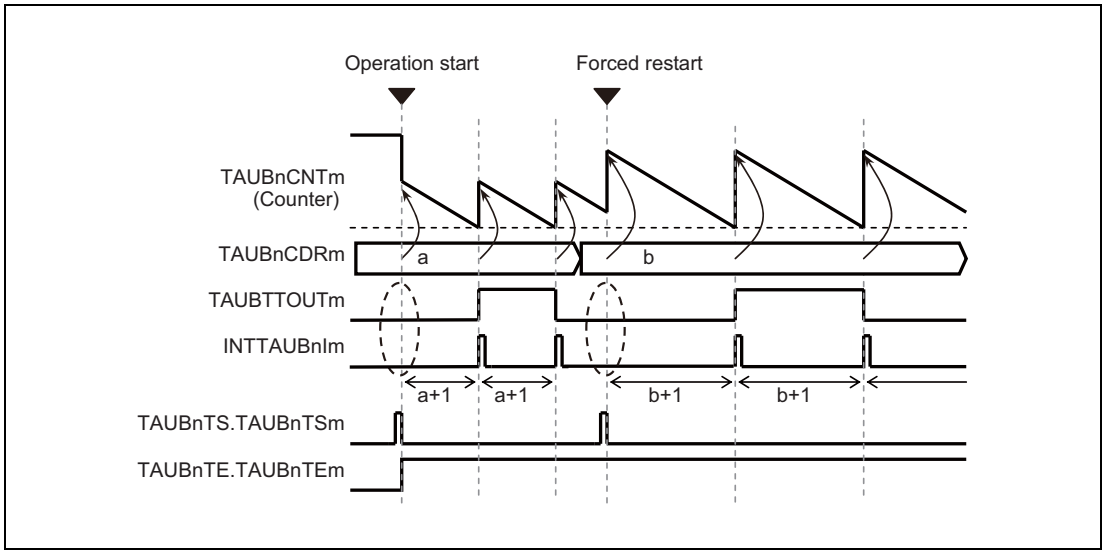


Figure 22.31 Forced Restart Operation (TAUBnCMORm.TAUBnMD0 = 0)

- 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.

22.12.2 TAUBTTINm Input Interval Timer Function

22.12.2.1 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. When an interrupt is generated, the TAUBTTOUTm signal toggles, resulting in a square wave.

Description

This function operates in an identical manner to the Interval Timer Function (see **Section 22.12.1, Interval Timer Function**), except that this function is restarted by a valid TAUBTTINm input edge. 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.

22.12.2.2 Equations

INTTAUBnIm cycle = count clock cycle × (TAUBnCDRm + 1)

TAUBTTOUTm square wave cycle = count clock cycle × (TAUBnCDRm + 1) × 2

22.12.2.3 Block Diagram and General Timing Diagram

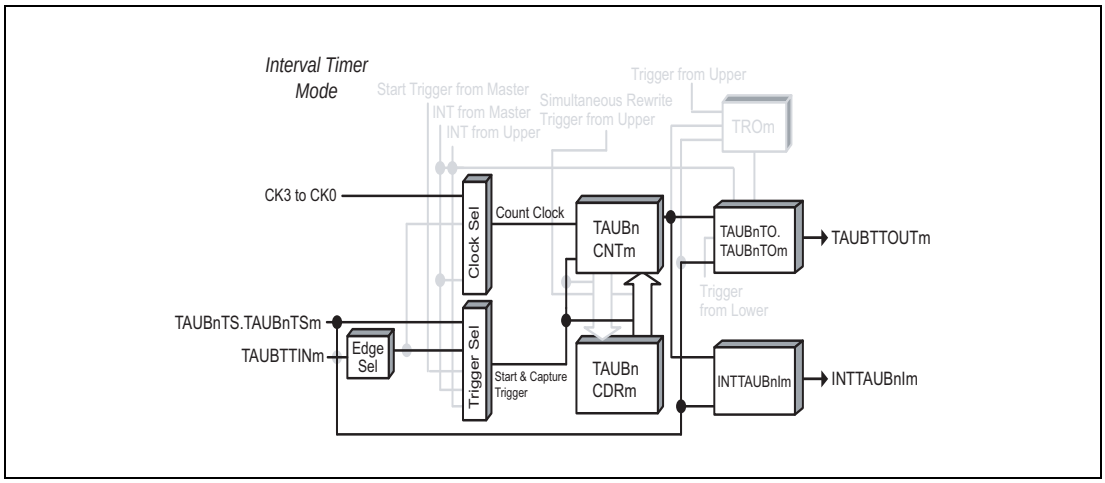


Figure 22.32 Block Diagram for TAUBTTINm Input Interval Timer Function

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)

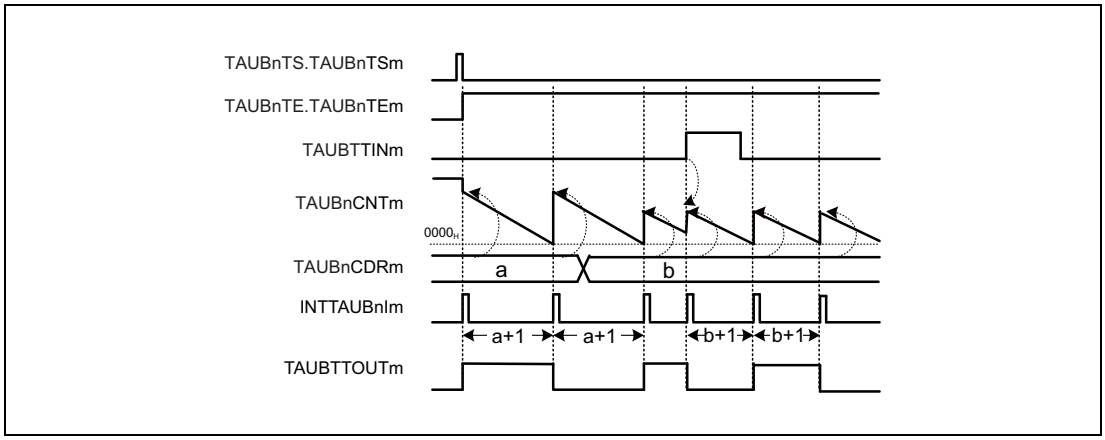


Figure 22.33 General Timing Diagram for TAUBTTINm Input Interval Timer Function

22.12.2.4 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 22.44 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 to this bit, 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 to this bit, 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

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 22.45 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 to these bits, 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

Table 22.46 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 TAUBTTINm Input Interval Timer Function. Therefore, these registers must be set to 0.

Table 22.47 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	

22.12.2.5 Operating Procedure for TAUBTTINm Input Interval Timer Function

Table 22.48 Operating Procedure for TAUBTTINm Input Interval Timer Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.44, Contents of the TAUBnCMORm Register for TAUBTTINm Input Interval Timer Function** and **Table 22.45, Contents of the TAUBnCMURm Register for TAUBTTINm Input Interval Timer Function** Set the value of the TAUBnCDRm register Set the channel output mode by setting the control bits as described in **Table 22.46, 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 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 and TAUBTTOUTm toggles 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 and TAUBTTOUTm stop and retain their current values.

Restart operation

22.12.2.6 Specific Timing Diagrams

The timing diagrams in **Section 22.12.1, Interval Timer Function** also apply, except for this function the counter can also be restarted by a valid TAUBTTINm input edge.

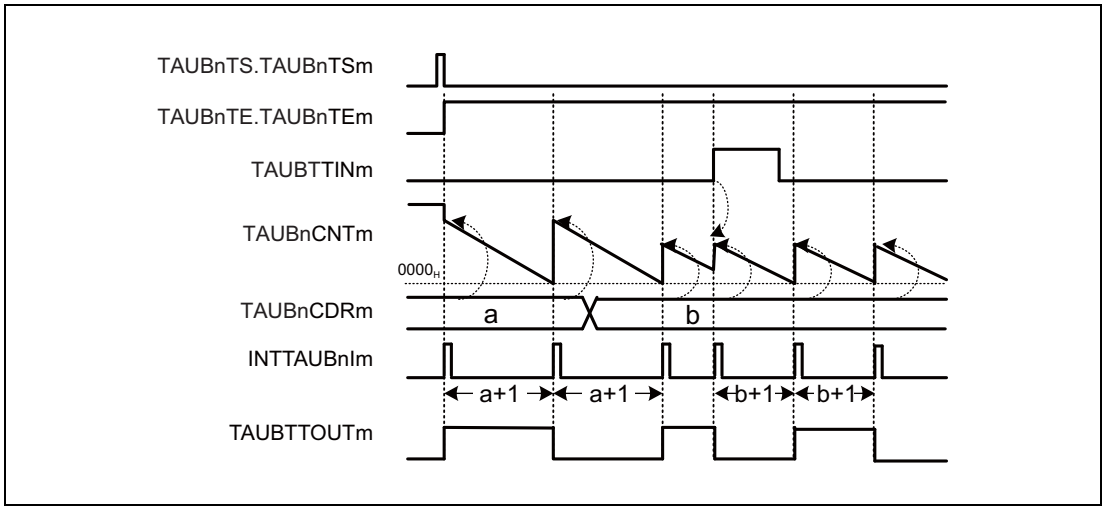


Figure 22.34 Counter Triggered by Rising TAUBTTINm Input Edge (TAUBnCMURm.TAUBnTIS[1:0] = 01_B), TAUBnCMORm.TAUBnMD0 = 1

- If a valid TAUBTTINm input edge is detected, an interrupt is generated which causes TAUBTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUBnCMURm.TAUBnTIS[1:0] = 01_B).

22.12.3 Clock Divide Function

22.12.3.1 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 the resulting signal is output to TAUBTTOUTm.

Prerequisites

- TAUBTTINm must have a fixed frequency
- The operation mode must be set to interval timer mode, see **Table 22.49, Contents of the TAUBnCMORm Register for Clock Divide Function**
- The channel output mode must be set to independent channel output mode 1.

Description

The counter is started 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 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 function starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm = 1, which in turn sets TAUBnTE.TAUBnTEm = 0. TAUBnCNTm and TAUBTTOUTm stop but retain their values. 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, and therefore TAUBTTOUTm does not toggle. This results in an inverted TAUBTTOUTm signal compared to when TAUBnCMORm.TAUBnMD0 is set to 1.

NOTE

The input TAUBTTINm is sampled at the frequency of the operation clock, specified by TAUBnCMORm.TAUBnCKS[1:0] bits. As a result, the output cycle of TAUBTTOUTm has an error of ± 1 operation clock cycle.

22.12.3.2 Equations

- When rising edge detection is selected:
  TAUBTTOUTm frequency = TAUBTTINm frequency /
  [(TAUBnCDRm + 1) × 2]
- When falling edge detection is selected:
  TAUBTTOUTm frequency = TAUBTTINm frequency /
  [(TAUBnCDRm + 1) × 2]
- When rising and falling edge detection is selected:
  TAUBTTOUTm frequency = TAUBTTINm frequency / (TAUBnCDRm + 1)

22.12.3.3 Block Diagram and General Timing Diagram

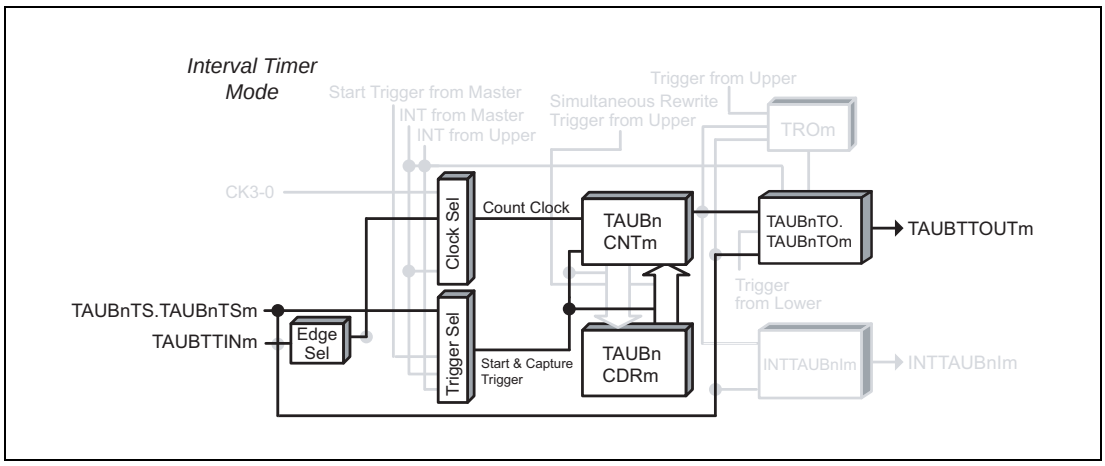


Figure 22.35 Block Diagram for Clock Divide Function

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)

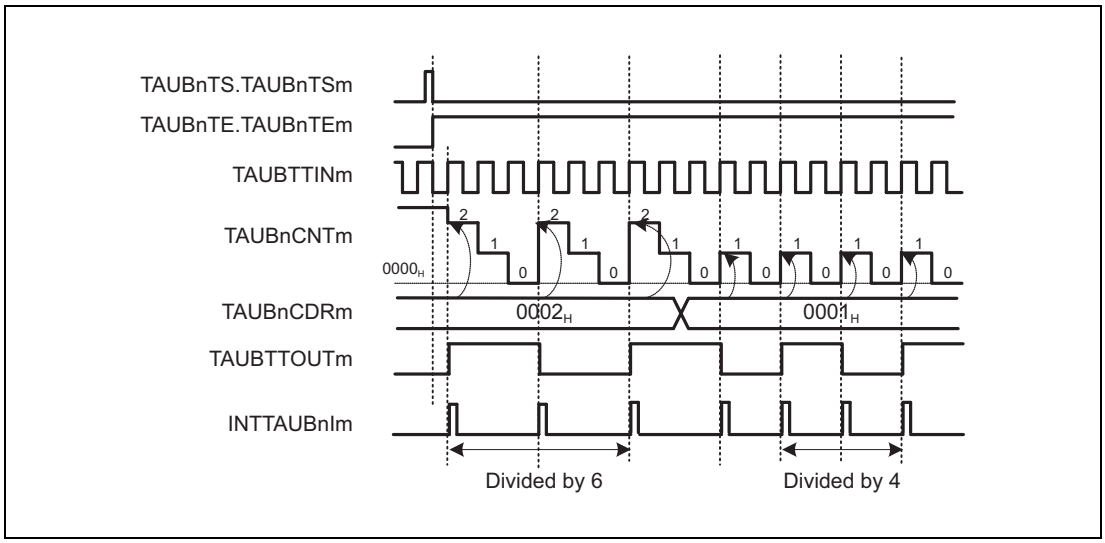


Figure 22.36 General Timing Diagram for Clock Divide Function

22.12.3.4 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 22.49 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 to this bit, 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 to this bit, 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

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 22.50 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 to these bits, 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

Table 22.51 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 Clock Divide Function. Therefore, these registers must be set to 0.

Table 22.52 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	

22.12.3.5 Operating Procedure for Clock Divide Function

Table 22.53 Operating Procedure for Clock Divide Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.49, Contents of the TAUBnCMORm Register for Clock Divide Function** and **Table 22.50, Contents of the TAUBnCMURm Register for Clock Divide Function** Set the value of the TAUBnCDRm register Set the channel output mode by setting the control bits as described in **Table 22.50, 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 and TAUBTTOUTm toggles.
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: • TTAUBnCNTm loads the TAUBnCDRm value and continues count operation • INTTAUBnIm is generated • TAUBTTOUTm toggles. 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 TAUBTTOUTm retain their current values.

Restart operation

22.12.3.6 Specific Timing Diagrams

(1) TAUBnCDRm = 0000_H

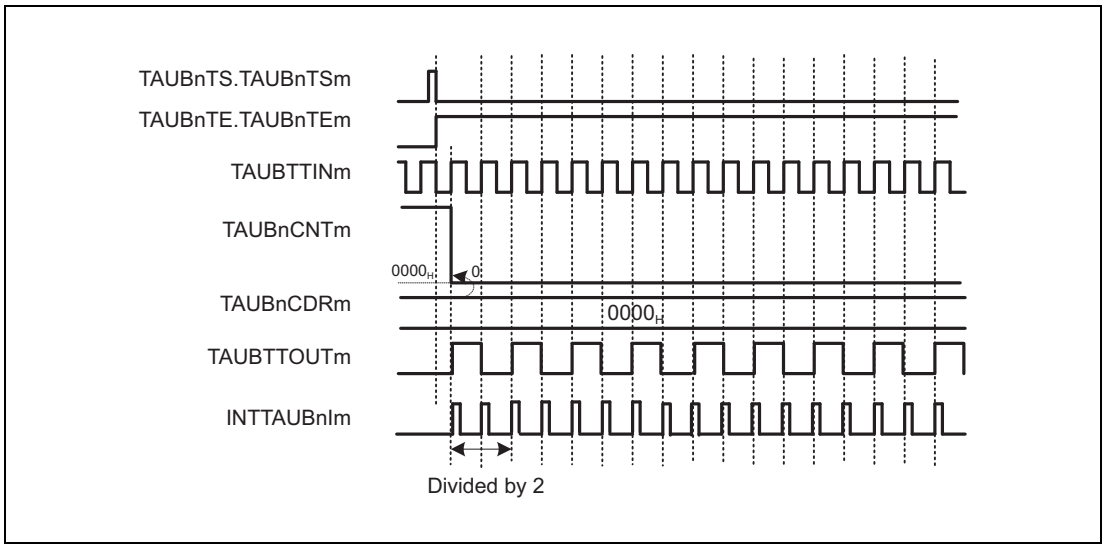


Figure 22.37 TAUBnCDRm = 0000_H, TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01_B

- If TAUBnCDRm is 0000_H, TAUBnCNTm is also always 0000_H.
- INTTAUBnIm is generated every count clock, resulting in TAUBTTOUTm toggling every count clock.

Figure 22.37 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 cause a delay from the TINm detection to the TOUTm output.

(2) Restart

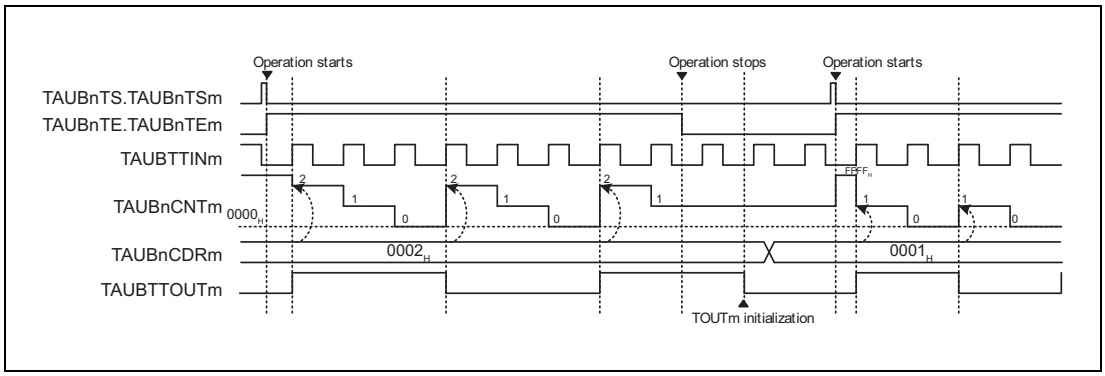


Figure 22.38 Restart (TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

To reset the value of TAUBTTOUTm:

- Set TAUBnTOE.TAUBnTOEm = 0 when the counter is stopped (TAUBnTE.TAUBnTEm = 0)
- Then write either 0 or 1 to TAUBnTO.TAUBnTOm to set the new start value of TAUBTTOUTm

(3) Forced restart

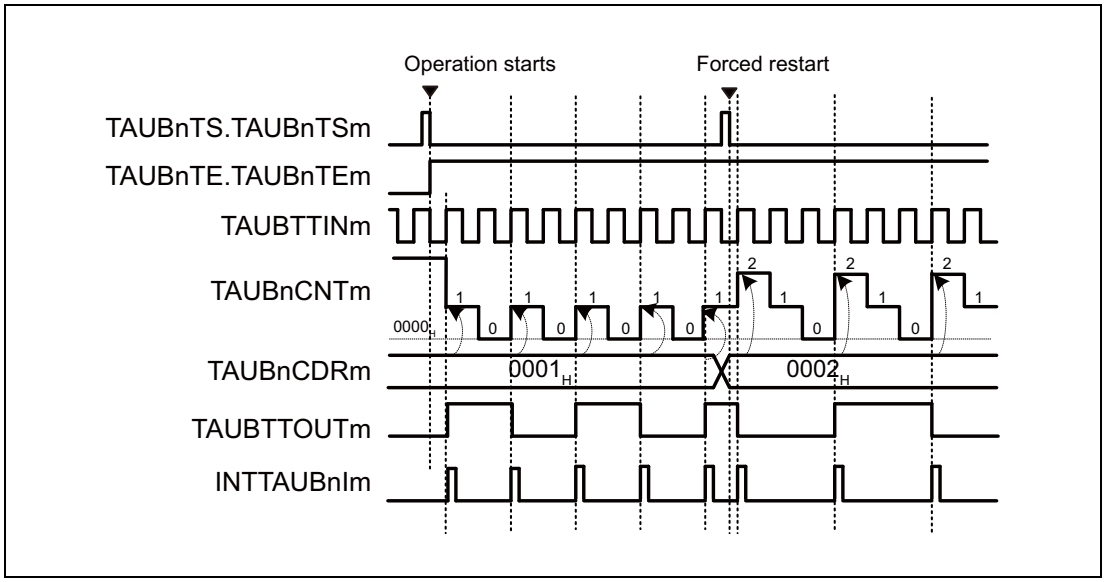


Figure 22.39 Forced Restart, (TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

To reset the value of TAUBTTOUTm.

- 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.
- TAUBTTOUTm restarts at the same level as before the forced restart.

22.12.4 External Event Count Function

22.12.4.1 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 22.54, 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 a valid TAUBTTINm input edge is detected, the value of TAUBnCNTm reduces by 1. TAUBnCNTm retains this value until a valid TAUBTTINm input edge is detected or the counter is restarted.

When the counter value reaches 0000_H, INTTAUBnIm is generated. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm stops and retains its value. 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.

22.12.4.2 Equations

Number of valid edges,
detected before INTTAUBnIm is generated = TAUBnCDRm + 1

22.12.4.3 Block Diagram for External Event Count Function

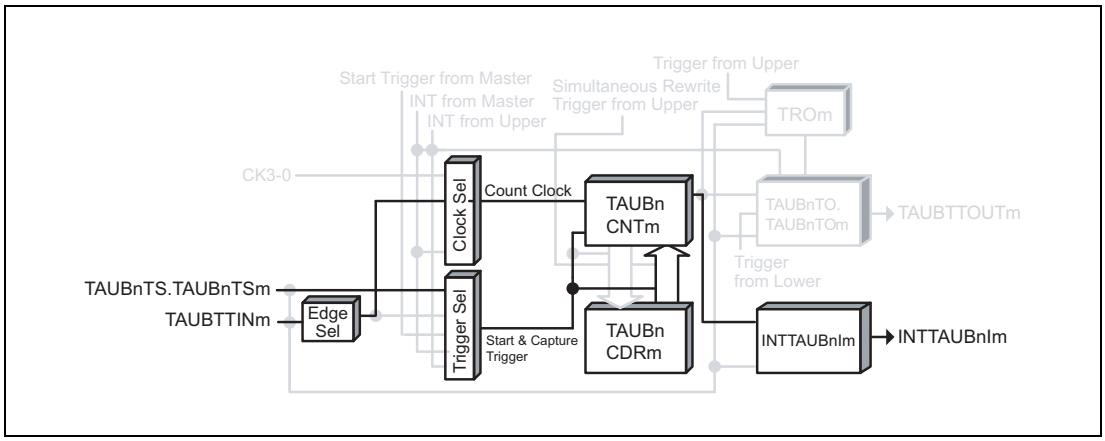


Figure 22.40 Block Diagram for External Event Count Function

The following settings apply to the general timing diagram.

- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

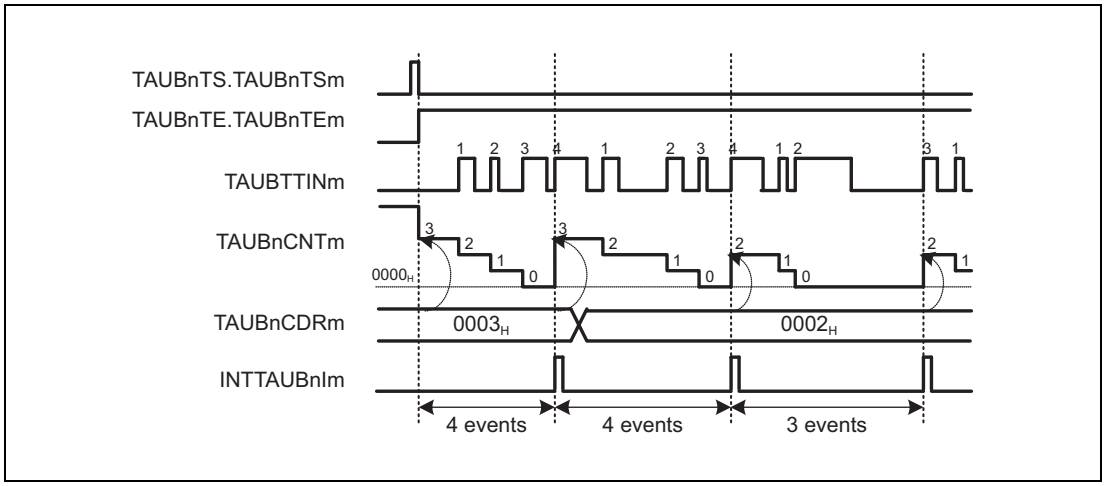


Figure 22.41 General Timing Diagram for External Event Count Function

22.12.4.4 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 22.54 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 to this bit, 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 to this bit, 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 22.55 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 to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge 01: Rising edge 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, and TAUBnRDC) cannot be used with the External Event Count Function.

Therefore, these registers must be set to 0.

Table 22.56 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	

22.12.4.5 Operating Procedure for External Event Count Function

Table 22.57 Operating Procedure for External Event Count Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.54, Contents of the TAUBnCMORm Register for External Event Count Function** and **Table 22.55, 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 the counter reaches 0000_H: • 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

22.12.4.6 Specific Timing Diagrams

(1) TAUBnCDRm = 0000_H

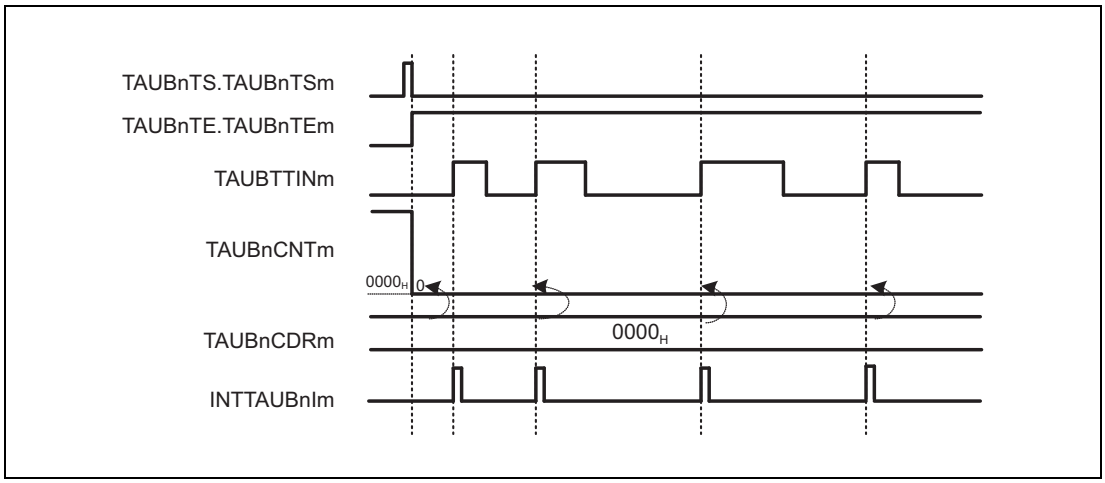


Figure 22.42 TAUBnCDRm = 0000_H, TAUBnCMURm.TAUBnTIS[1:0] = 01_B

- 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

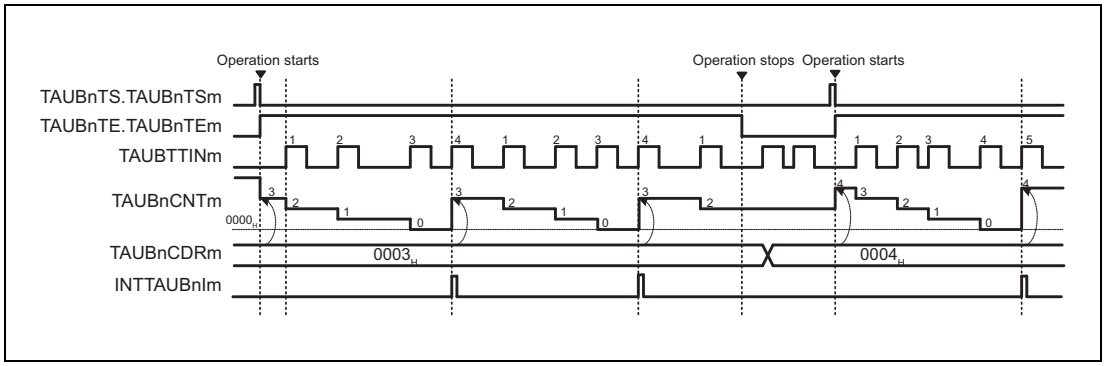


Figure 22.43 Operation Stop and Restart, (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

- 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

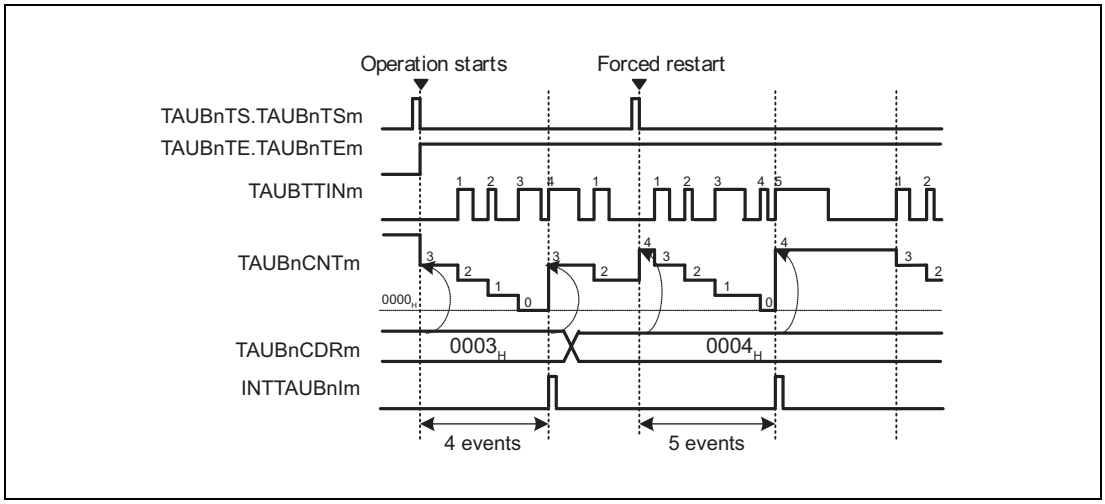


Figure 22.44 Forced Restart, (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

A forced restart applies the new TAUBnCDRm value to TAUBnCNTm immediately.

- 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.

22.12.5 One-Pulse Output Function

22.12.5.1 Overview

Summary

This function generates an interrupt (INTTAUBnIm) when a valid TAUBTTINm input edge is detected and also a specific interval later. TAUBTTINm input signal pulses that occur within the defined interval are ignored. When an interrupt is generated, the TAUBTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode must be set to pulse one-count mode. (See **Table 22.58, Contents of the TAUBnCMORm Register for One-Pulse Output Function**).
- The channel output mode must be set to independent channel output mode 2. (See **Table 22.60, Control Bit Settings for Independent Channel Output Mode 2**.)
- 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. An interrupt is generated and TAUBTTOUTm is set to the active level.

When the counter reaches 0001_H an interrupt is generated and TAUBTTOUTm is set to the active level. The counter stops at 0000_H and awaits the next valid 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.

22.12.5.2 Equations

Interval between TAUBTTINm and INTTAUBnIm = TAUBTTOUTm (timer output) width = count clock cycle × TAUBnCDRm

22.12.5.3 Block Diagram and General Timing Diagram

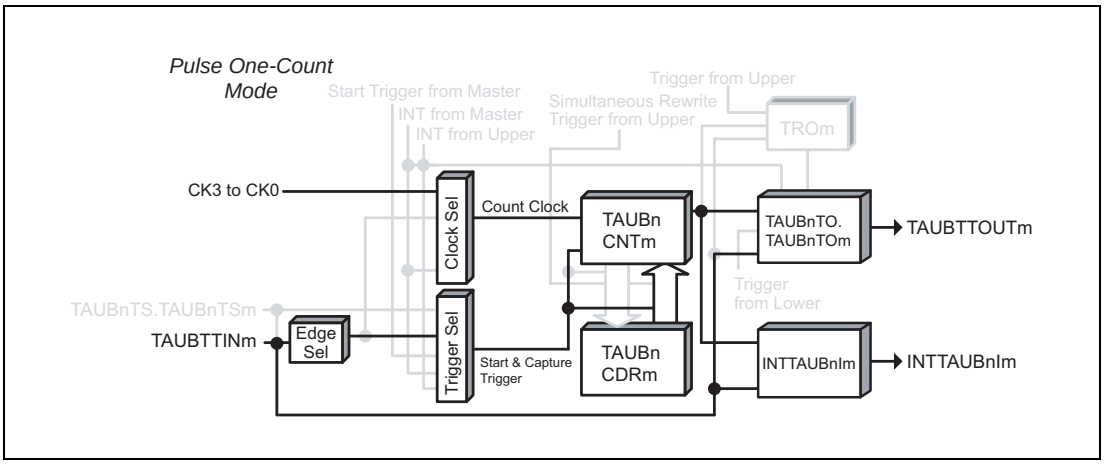


Figure 22.45 Block Diagram for One-Pulse Output Function

The following settings apply to the general timing diagram.

- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

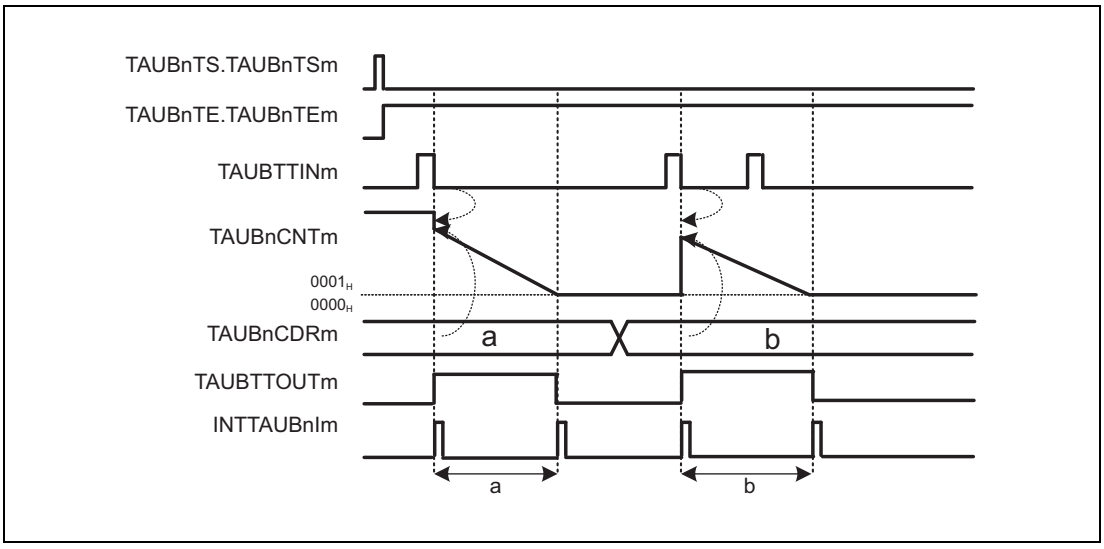


Figure 22.46 General Timing Diagram for One-Pulse Output Function

22.12.5.4 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 22.58 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 to this bit, 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 to this bit, 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 22.59 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 to these bits, 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

Table 22.60 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.

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 One-Pulse Output Function.
Therefore, these registers must be set to 0.

Table 22.61 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	

22.12.5.5 Operating Procedure for One-Pulse Output Function

Table 22.62 Operating Procedure for One-Pulse Output Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in**Table 22.58, Contents of the TAUBnCMORm Register for One-Pulse Output Function** and **Table 22.59, Contents of the TAUBnCMURm Register for One-Pulse Output Function** Set the value of the TAUBnCDRm register Set the channel output mode by setting the control bits as described in **Table 22.60, Control Bit Settings for Independent Channel Output Mode 2**	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 and TAUBTTOUTm is set to its active level. TAUBnCNTm counts down. When the counter reaches 0001_H: • INTTAUBnIm is generated • TAUBTTOUTm is set to its inactive level. 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 and TAUBTTOUTm stop and retain their current values.

Restart operation

22.12.6 TAUBTTINm Input Pulse Interval Measurement Function

22.12.6.1 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 started 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 22.63 Effects of an Overflow

	When Overflow Occurs		When a Valid TAUBTTINm Input is then Detected	
TAUBnCMORm.COS[1:0]	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 a CPU command that sets 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.

22.12.6.2 Equations

TAUBTTINm input pulse interval = count clock cycle ×

[(TAUBnCSRm.TAUBnOVF × ($FFFF_H$ + 1)) + TAUBnCDRm capture value + 1]

22.12.6.3 Block Diagram and General Timing Diagram

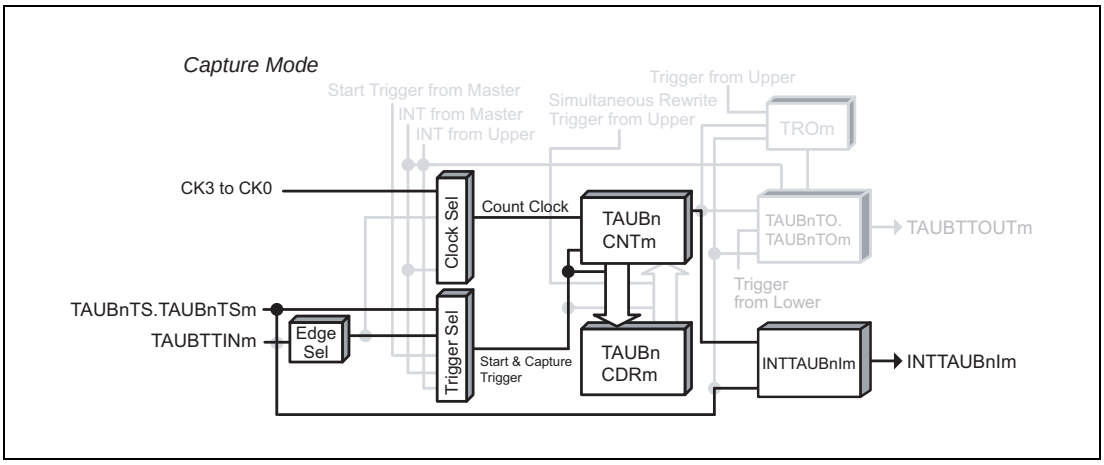


Figure 22.47 Block Diagram for TAUBTTINm Input Pulse Interval Measurement Function

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)

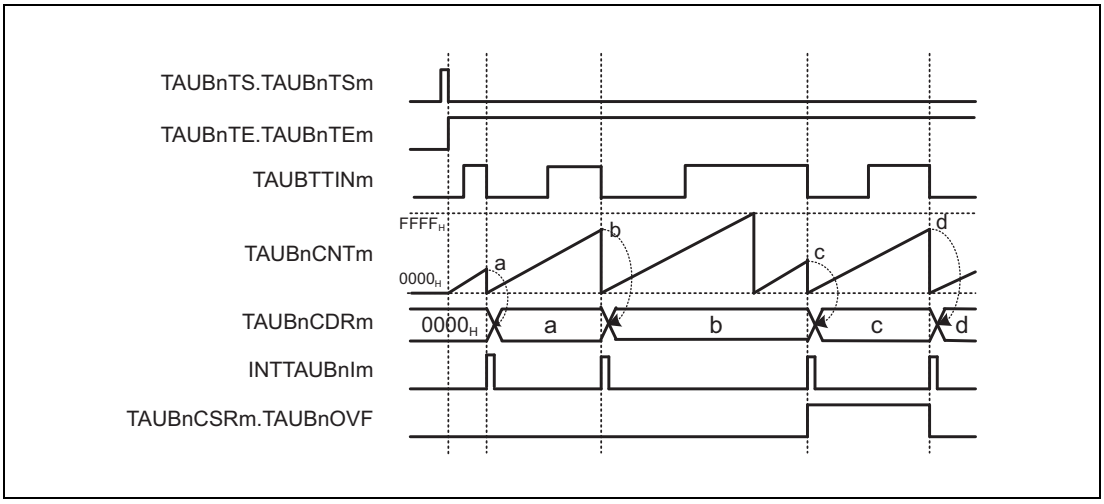


Figure 22.48 General Timing Diagram for TAUBTTINm Input Pulse Interval Measurement Function

22.12.6.4 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 22.64 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 to this bit, 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 22.63, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing to this bit, 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 22.65 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 to these bits, 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 22.66 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	

22.12.6.5 Operating Procedure for TAUBTTINm Input Pulse Interval Measurement Function

Table 22.67 Operating Procedure for TAUBTTINm Input Pulse Interval Measurement Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.64, Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Measurement Function** and **Table 22.65, 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

22.12.6.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUBnCMORm.TAUBnCOS[1:0] = 00_B

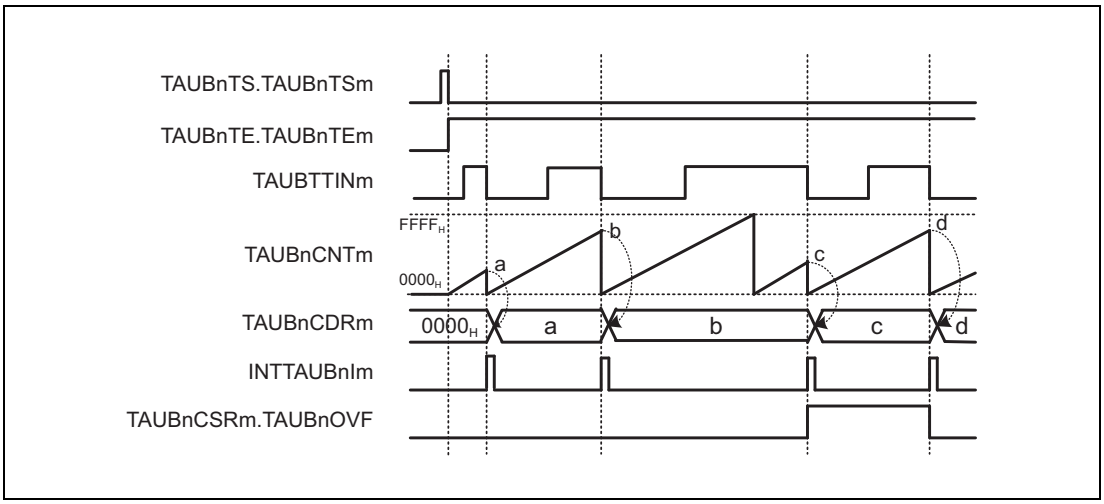


Figure 22.49 TAUBnCMORm.TAUBnCOS[1:0] = 00_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- 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

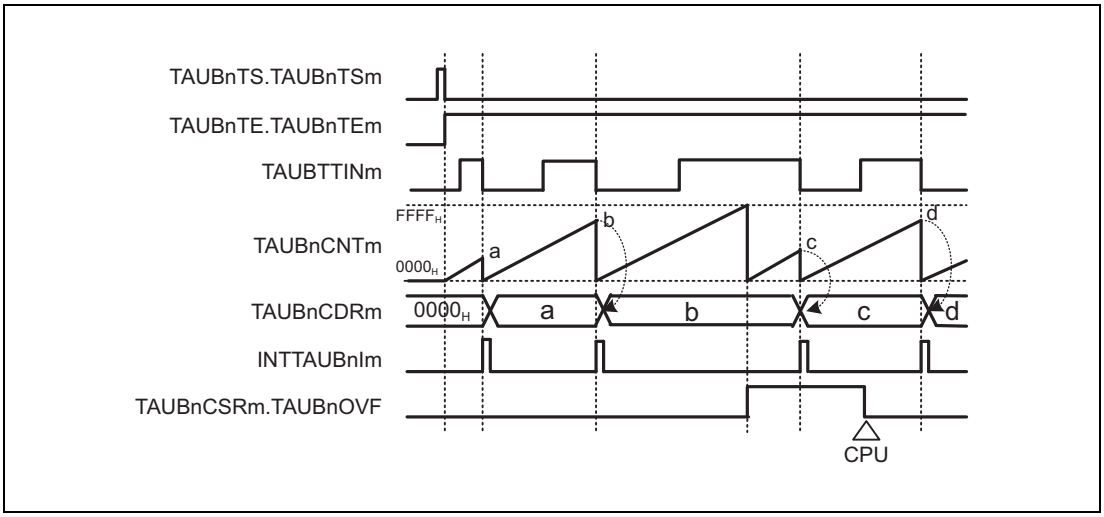


Figure 22.50 TAUBnCMORm.TAUBnCOS[1:0] = 01_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- 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

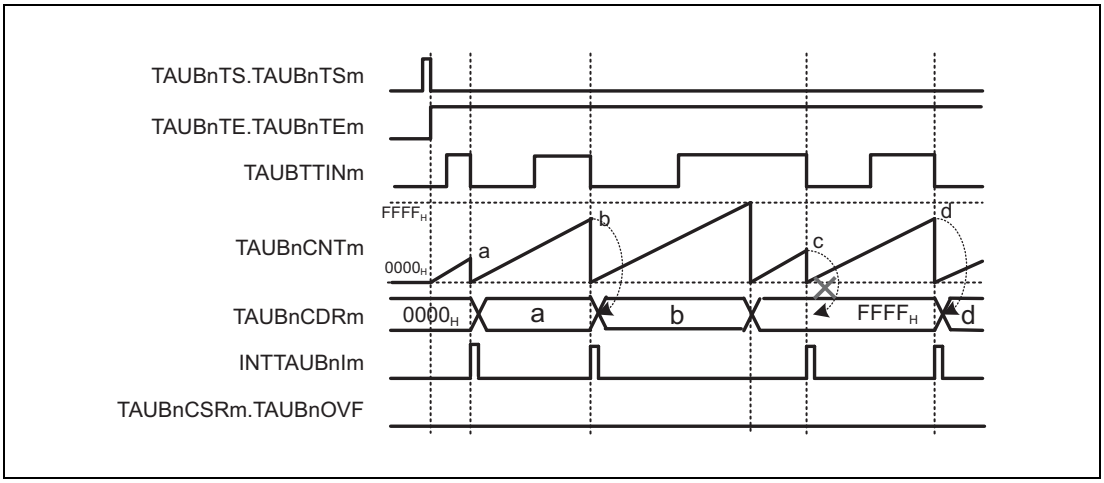


Figure 22.51 TAUBnCMORm.TAUBnCOS[1:0] = 10_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- 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

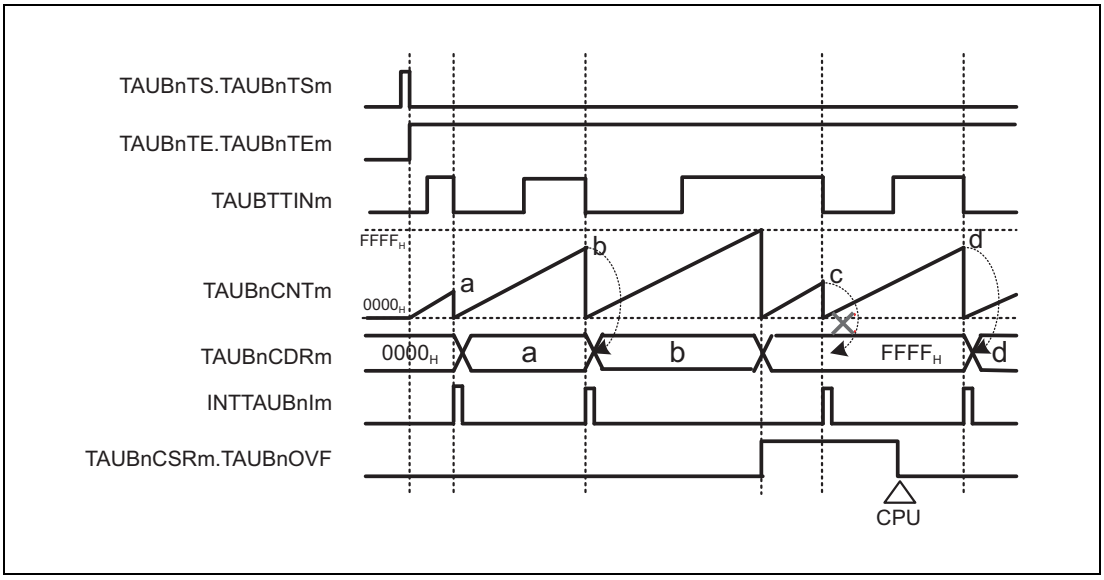


Figure 22.52 TAUBnCMORm.TAUBnCOS[1:0] = 11_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- 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 (CMORn:MD0 = 1)

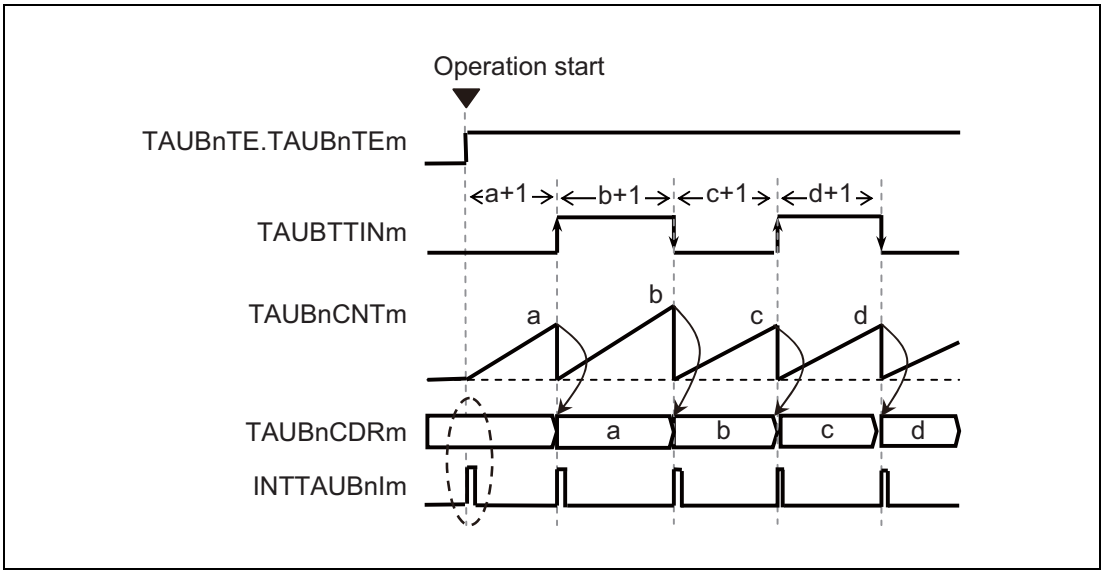


Figure 22.53 TAUBnCMORn: TAUBnMD0 = 1

Setting TAUBnCMURm.TAUBnTIS1 and TAUBnCMURm.TAUBnTIS0 to 10_B (rising and falling edge detection selected) measures the TAUBTTINm rising and falling edge intervals.

If TS: TAUBnTS.TAUBnTSm is set to 1 while MD0 is 1 and TE: TAUBnTE.TAUBnTEm is 0, INTTAUBnIm and TAUBnCDRm is a)

= Count clock cycle × ((10000$_H$ × CSRn: OVF) + (TAUBnCDRm capture value + 1))

= Count clock cycle × ((10000$_H$ × 0) + (a+1))

= Count clock cycle × (a+1)

(6) Operation stop and operation restart (CMORn:MD0 = 0)

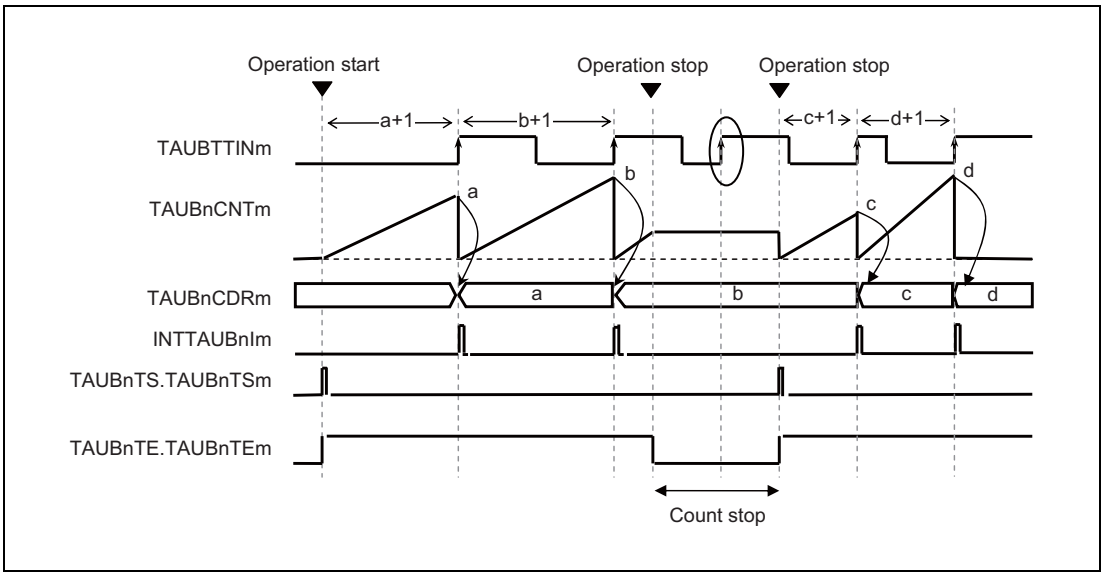


Figure 22.54 Operation Stop and Operation Restart (CMORn:MD0 = 0)

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.

22.12.7 TAUBTTINm Input Signal Width Measurement Function

22.12.7.1 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 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 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 22.68 Effects of an Overflow

	When Overflow Occurs		When a Valid TAUBTTINm Input Stop Edge is Detected	
TAUBnCMORm.COS[1:0]	**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 a CPU command that sets 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.

22.12.7.2 Equations

TAUBTTINm input signal width = count clock cycle ×

[(TAUBnCSRm.OVF × ($FFFF_H$ + 1)) + TAUBnCDRm capture value + 1]

22.12.7.3 Block Diagram and General Timing Diagram

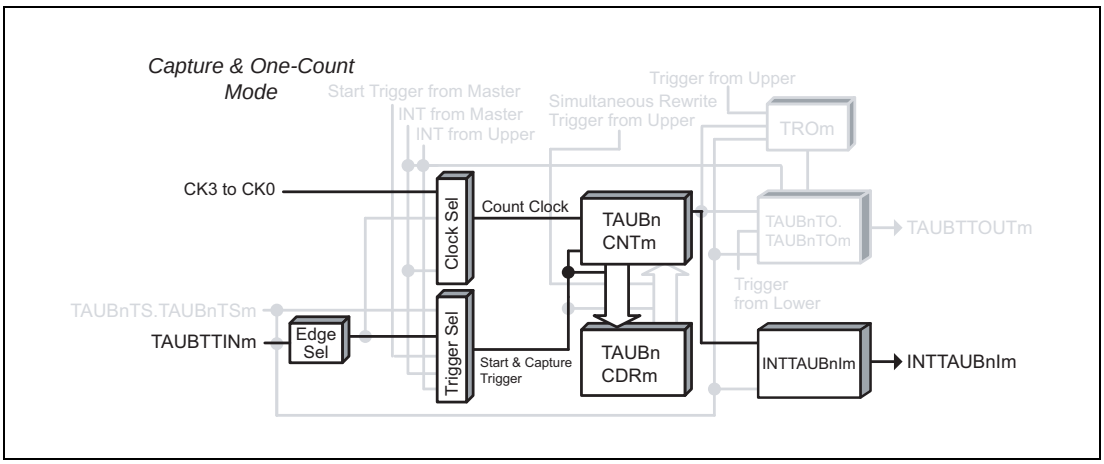


Figure 22.55 Block Diagram for TAUBTTINm Input Signal Width Measurement Function

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.OVF is set to 1 (TAUBnCMORm.TAUBnCOS[1:0] = 00_B)

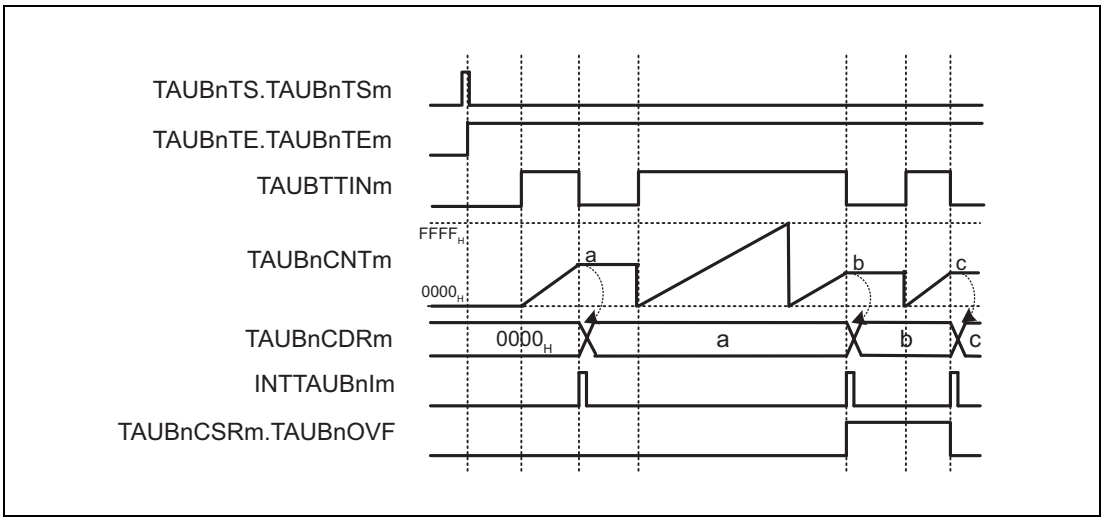


Figure 22.56 General Timing Diagram for TAUBTTINm Input Signal Width Measurement Function

22.12.7.4 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 22.69 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 to this bit, 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 22.68, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing to this bit, 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 22.70 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 to these bits, 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 22.71 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.RDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

22.12.7.5 Operating Procedure for TAUBTTINm Input Signal Width Measurement Function

Table 22.72 Operating Procedure for TAUBTTINm Input Signal Width Measurement Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.69, Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Measurement Function** and **Table 22.70, 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. The TAUBnCSC.CLOV 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

22.12.7.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUBnCMORm.TAUBnCOS[1:0] = 00_B

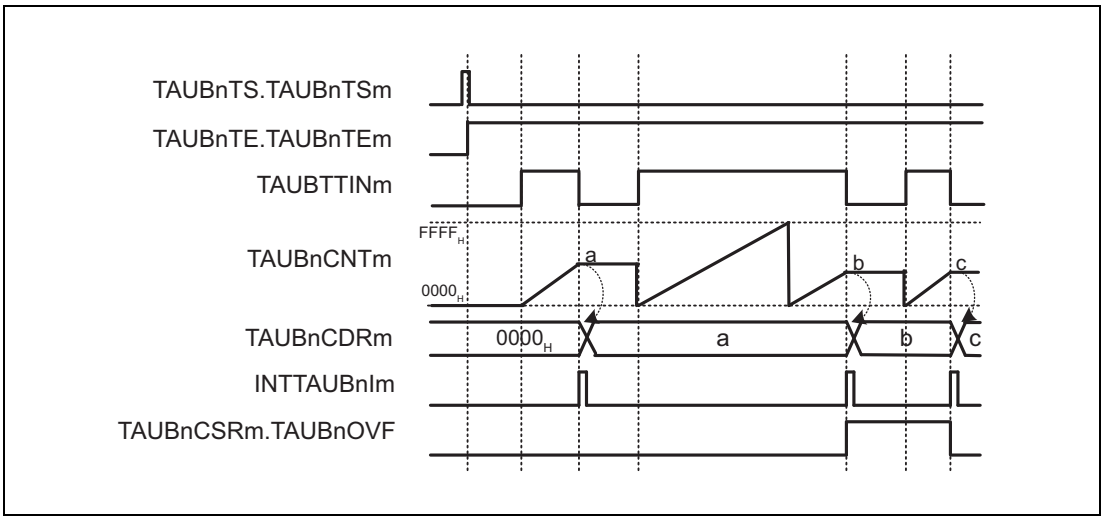


Figure 22.57 TAUBnCMORm.TAUBnCOS[1:0] = 00_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- 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 written 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

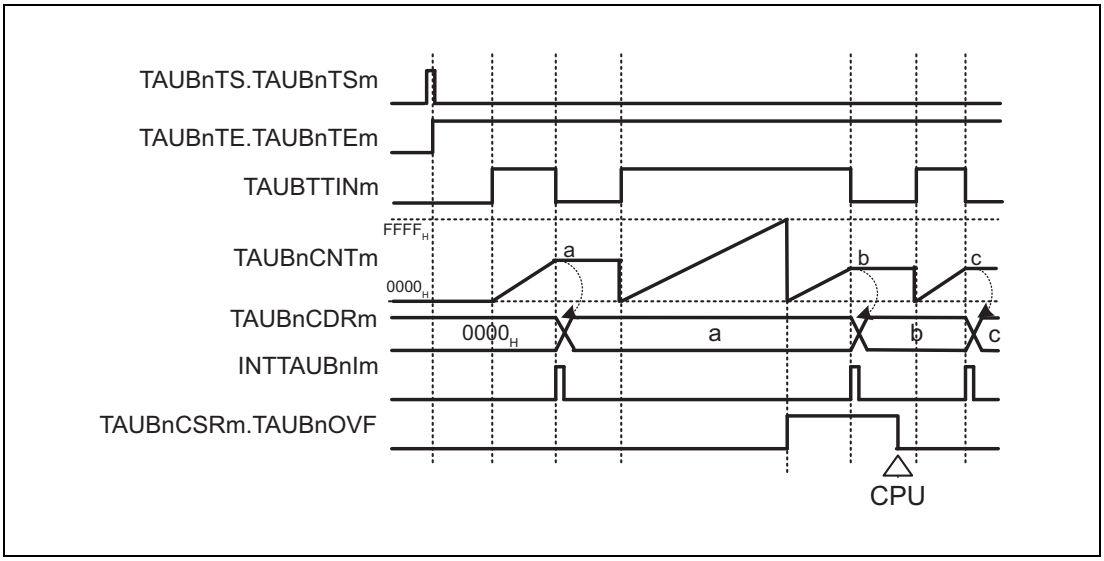


Figure 22.58 TAUBnCMORm.TAUBnCOS[1:0] = 01_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- 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

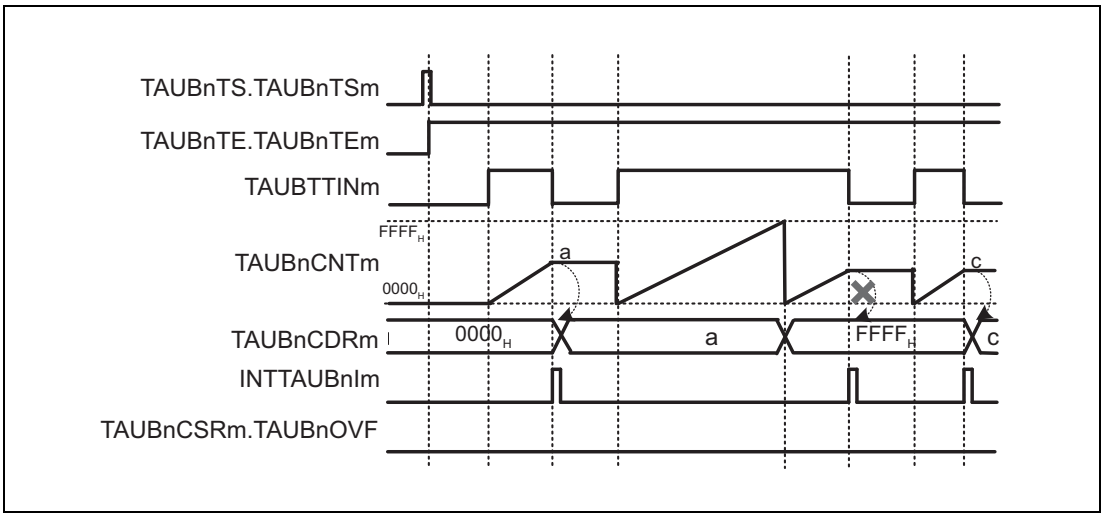


Figure 22.59 TAUBnCMORm.TAUBnCOS[1:0] = 10_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- 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

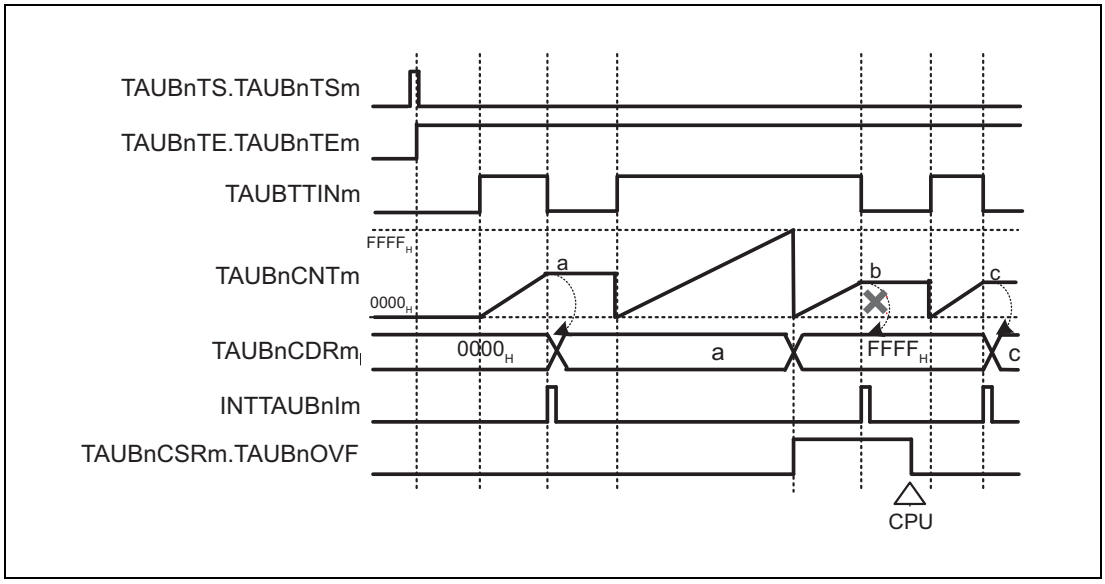


Figure 22.60 TAUBnCMORm.TAUBnCOS[1:0] = 11_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- 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)

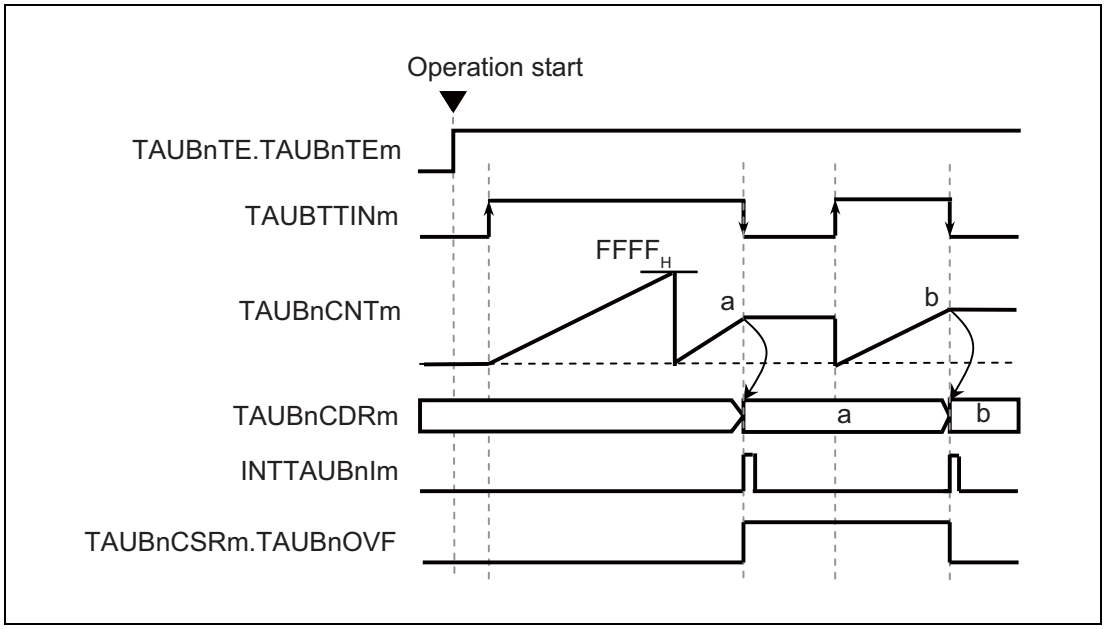


Figure 22.61 When an Overflow Occurs

When a capture trigger is input after the counter value has overflowed, the counter value is transferred to TAUBnCDRm and at the same time CSRn: OVF is set to 1.

CSRn: OVF is kept at 1 until the next capture trigger occurs.

If the next capture trigger is not accompanied by an overflow, CSRn: OVF is cleared to 0.

TAUBTTINm input signal width (example when CSRn: OVF is 1 and TAUBnCDRm is a)
= count clock cycle × ((10000_H × CSRn: OVF) + (TAUBnCDRm capture value + 1))
= count clock cycle × ((10000_H × 1) + (a+1))
= count clock cycle × (1000_H + a+1)

22.12.8 TAUBTTINm Input Position Detection Function

22.12.8.1 Overview

Summary

This function measures the interval of an input signal 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. The counter starts to count from 0000_H. When a valid TAUBTTINm input stop edge is detected, the current TAUBnCNTm value is written to TAUBnCDRm and an interrupt (INTTAUBnIm) is generated. The counter continues to count from the current value until the next valid TAUBnTTINm input edge is detected.

When the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTE

The input TAUBTTINm 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.

22.12.8.2 Equations

Function duration at a TAUBTTINm input pulse =

count clock cycle × (TAUBnCDRm capture value + 1)

22.12.8.3 Block Diagram and General Timing Diagram

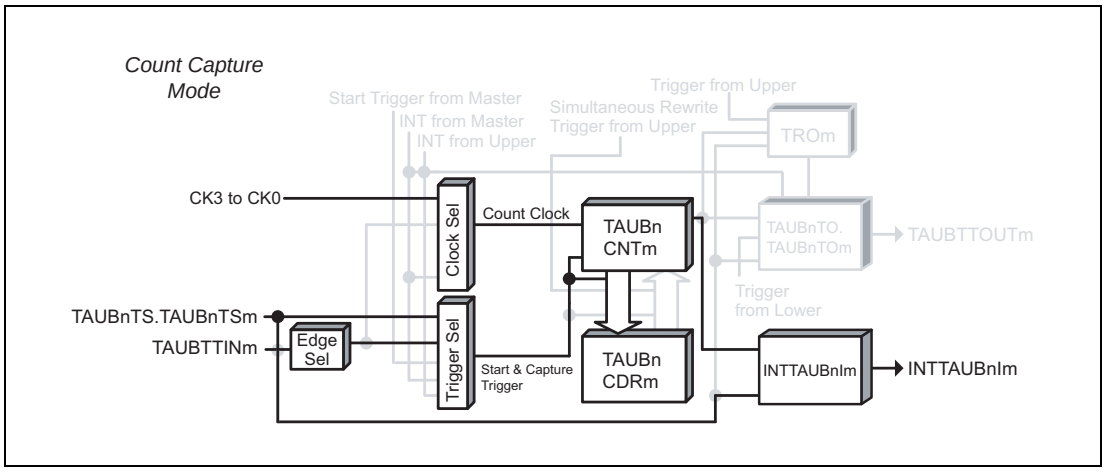


Figure 22.62 Block Diagram for TAUBTTINm Input Position Detection Function

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)

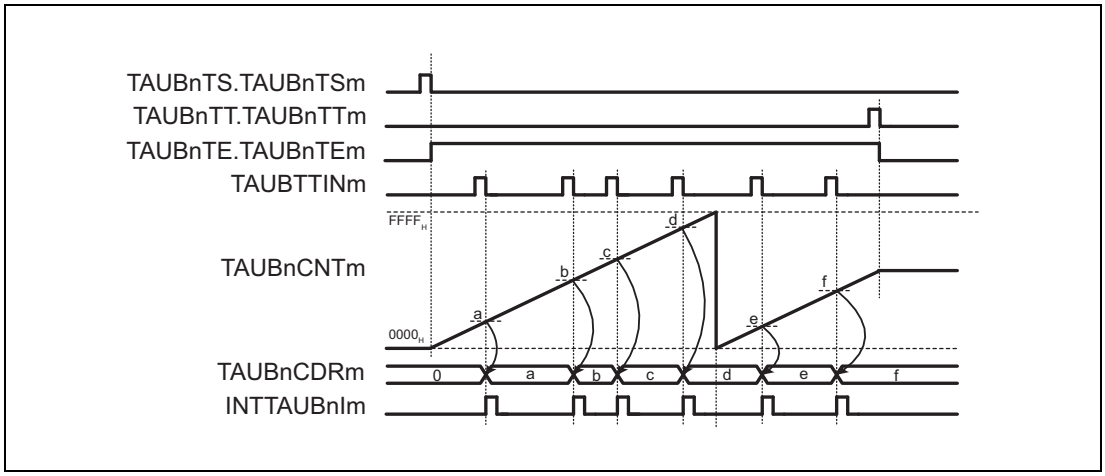


Figure 22.63 General Timing Diagram for TAUBTTINm Input Position Detection Function

22.12.8.4 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 22.73 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 to this bit, 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 to this bit, 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 22.74 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 to these bits, 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 22.75 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	

22.12.8.5 Operating Procedure for TAUBTTINm Input Position Detection Function

Table 22.76 Operating Procedure for TAUBTTINm Input Position Detection Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.73, Contents of the TAUBnCMORm Register for TAUBTTINm Input Position Detection Function** and **Table 22.74, 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

22.12.8.6 Specific Timing Diagrams

(1) Operation stop and restart

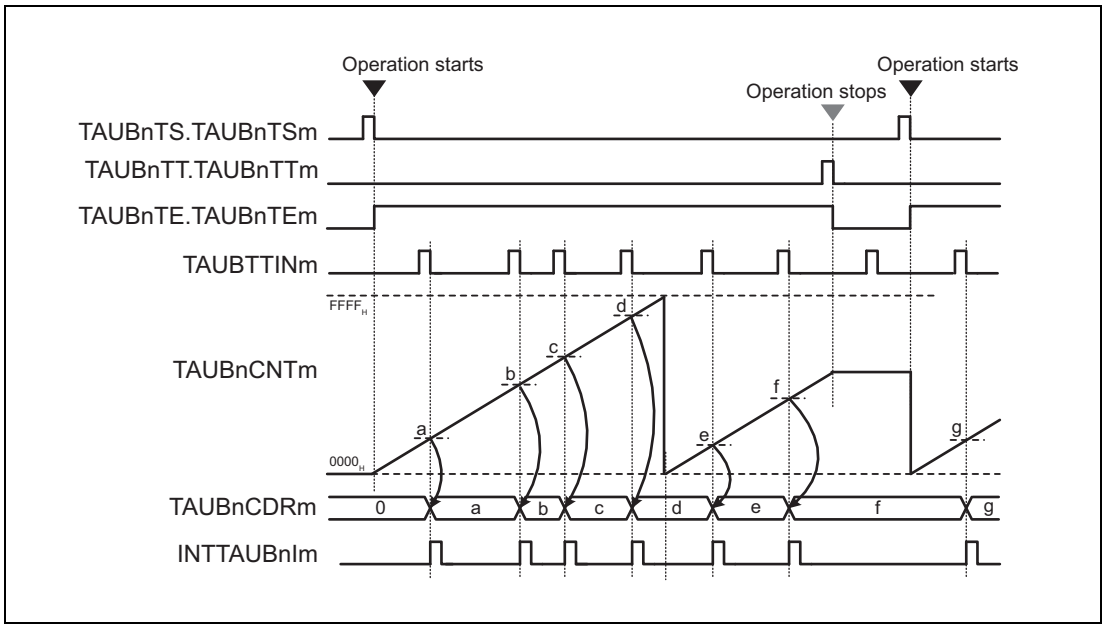


Figure 22.64 Operation Stop and Restart, (TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

- 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.

22.12.9 TAUBTTINm Input Period Count Detection Function

22.12.9.1 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 written to TAUBnCDRm and an interrupt (INTTAUBnIm) is generated. The counter stops and retains its value (TAUBnCDRn + 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 input TAUBTTINm 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 disabled 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.

22.12.9.2 Equations

Cumulative TAUBTTINm input width =

count clock cycle × (TAUBnCDRm capture value + 1)

22.12.9.3 Block Diagram and General Timing Diagram

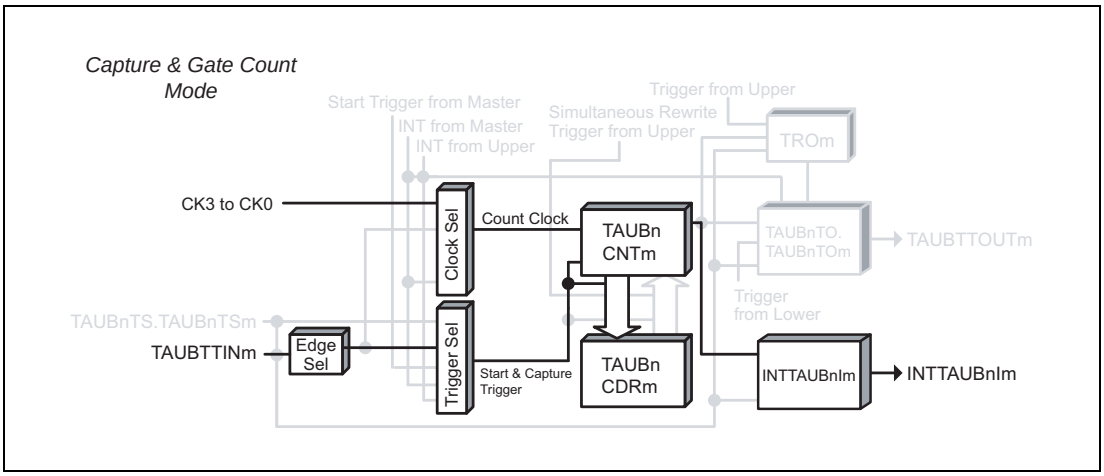


Figure 22.65 Block Diagram for TAUBTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

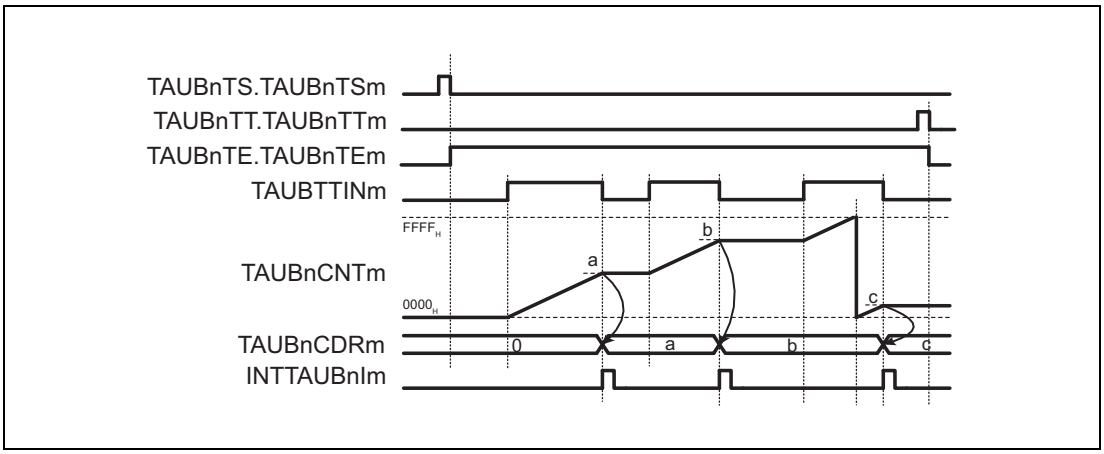


Figure 22.66 General Timing Diagram for TAUBTTINm Input Period Count Detection Function

22.12.9.4 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 22.77 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 to this bit, 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 to this bit, 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 22.78 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 to these bits, 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 22.79 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	

22.12.9.5 Operating Procedure for TAUBTTINm Input Period Count Detection Function

Table 22.80 Operating Procedure for TAUBTTINm Input Period Count Detection Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.77, Contents of the TAUBnCMORm Register for TAUBTTINm Input Period Count Detection Function** and **Table 22.78, 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 TAUBnTTINm 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 capture 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

22.12.9.6 Specific Timing Diagrams

(1) Operation stop and restart

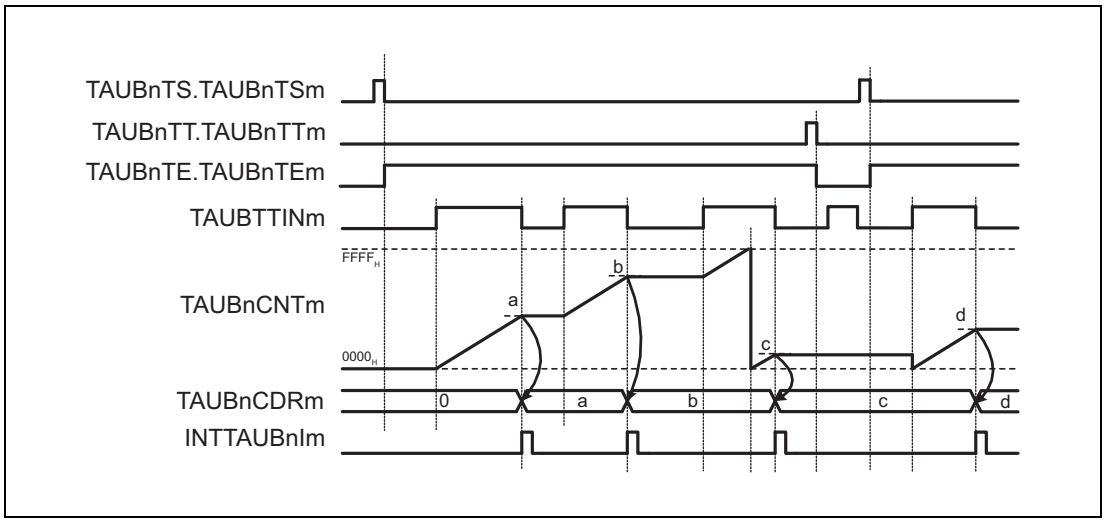


Figure 22.67 Operation Stop and Restart, (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

- 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.

22.12.10 TAUBTTINm Input Pulse Interval Judgment Function

22.12.10.1Overview

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 started 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 written 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.

22.12.10.2Block Diagram and General Timing Diagram

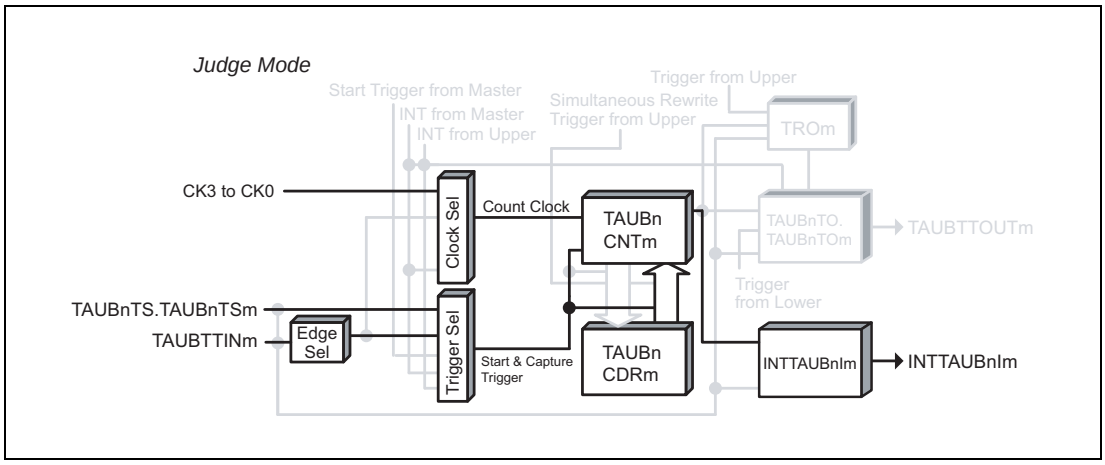


Figure 22.68 Block Diagram for TAUBTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

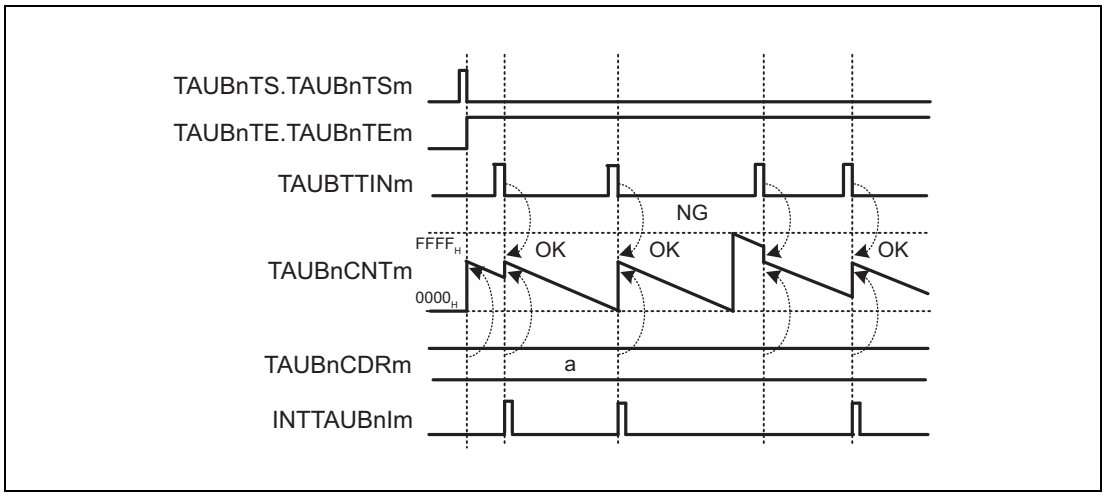


Figure 22.69 General Timing Diagram for TAUBTTINm Input Pulse Interval Judgment Function

22.12.10.3Register 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 22.81 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 to this bit, 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 to this bit, 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 22.82 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 to these bits, 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 22.83 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), RDM.RDMm set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

22.12.10.4 Operating Procedure for TAUBTTINm Input Pulse Interval Judgment Function

Table 22.84 Operating Procedure for TAUBTTINm Input Pulse Interval Judgment Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.81, Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Judgment Function** and **Table 22.82, 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

22.12.11 TAUBTTINm Input Signal Width Judgment Function

22.12.11.1Overview

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.TSm) 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 written 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, (When 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.

22.12.11.2Block Diagram and General Timing Diagram

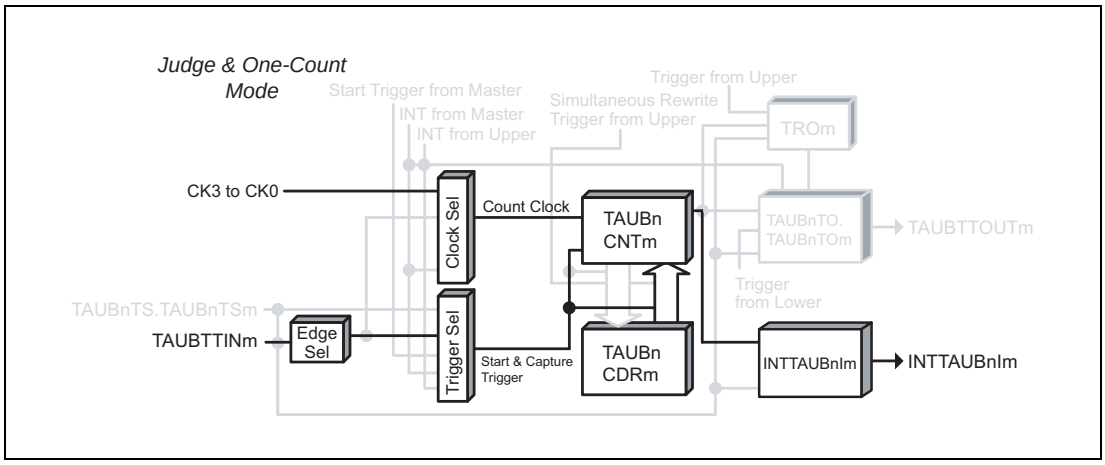


Figure 22.70 Block Diagram for TAUBTTINm Input Signal Width Judgment Function

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)

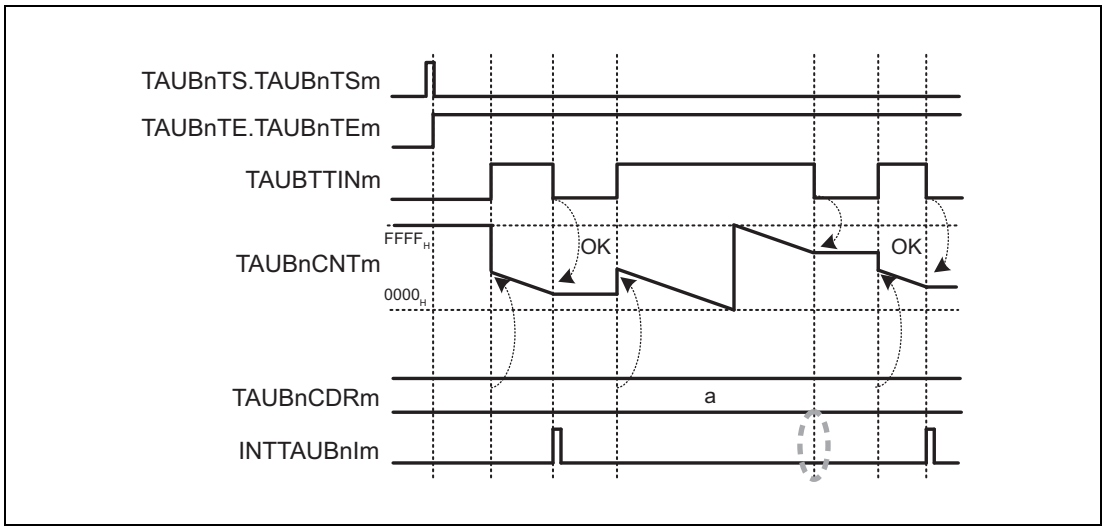


Figure 22.71 General Timing Diagram for TAUBTTINm Input Signal Width Judgment Function

22.12.11.3Register 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 22.85 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 to this bit, 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 to this bit, 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 22.86 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 to these bits, 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 22.87 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), RDM.RDMm set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

22.12.11.4 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function

Table 22.88 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.85, Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Judgment Function** and **Table 22.86, 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

22.12.12 Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

22.12.12.1 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 written 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.

22.12.12.2Block Diagram and General Timing Diagram

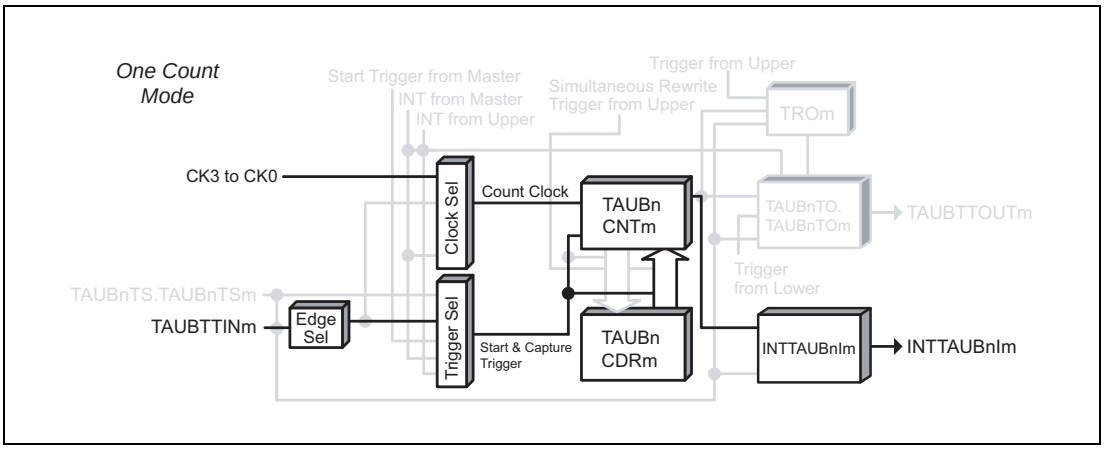


Figure 22.72 Block Diagram for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

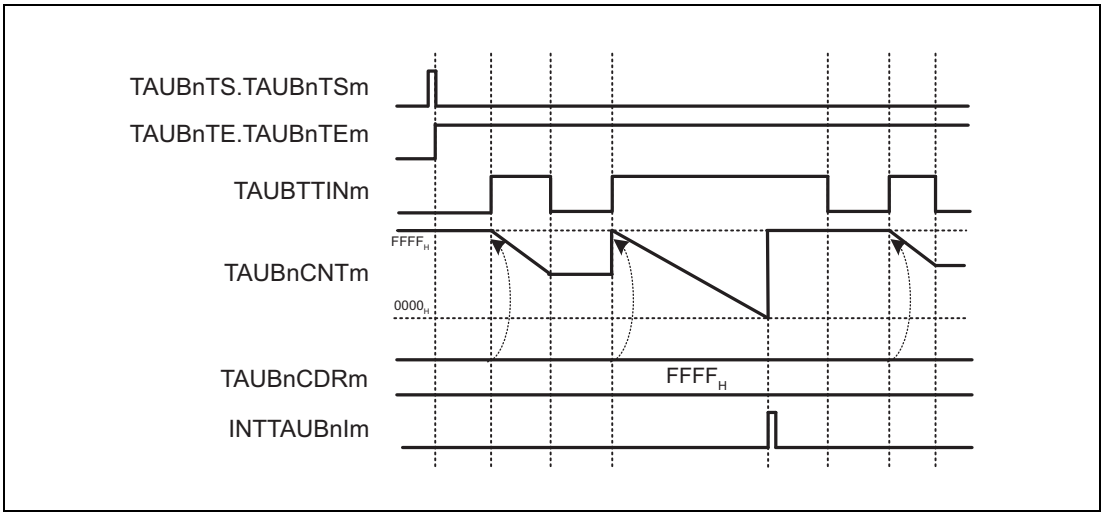


Figure 22.73 General Timing Diagram for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

22.12.12.3Register 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 22.89 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 to this bit, 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 to this bit, 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 22.90 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 to these bits, 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 22.91 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	

22.12.12.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Table 22.92 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement) (1/2)

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.89, Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)** and **Table 22.90, 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.

Table 22.92 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement) (2/2)

	Operation	Status of TAUBn
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

22.12.13 Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

22.12.13.1 Overview

Summary

This function measures the cumulative width of a TAUBTTINm input signal. An interrupt is 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 written 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 written 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.TIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, the TAUBTTINm input low period 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 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.

22.12.13.2Block Diagram and General Timing Diagram

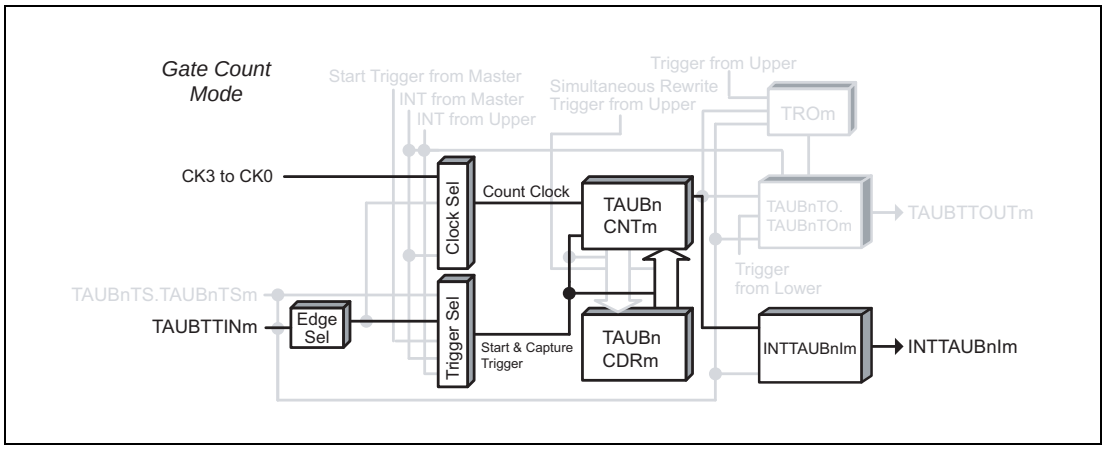


Figure 22.74 Block Diagram for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

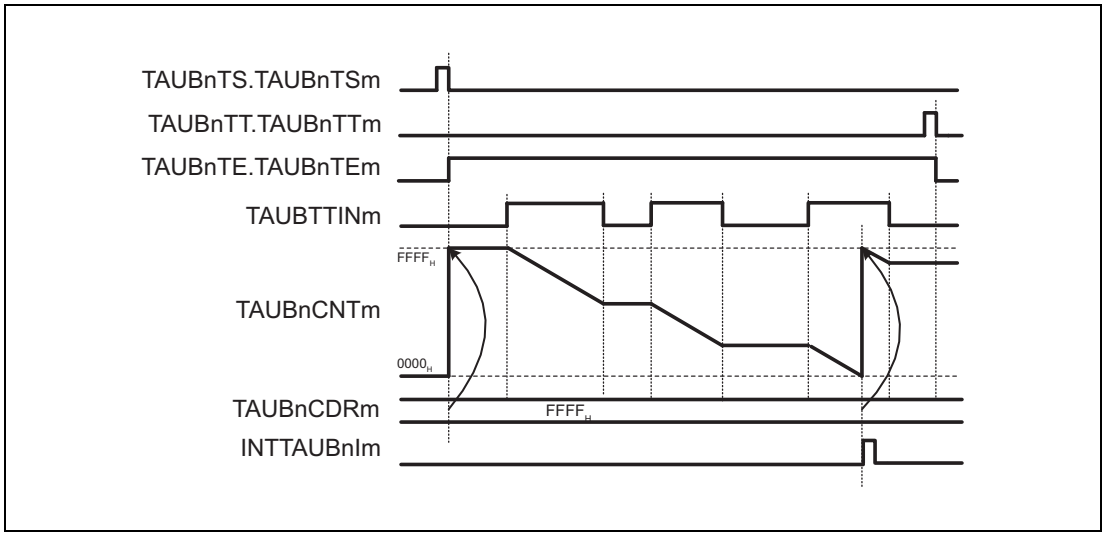


Figure 22.75 General Timing Diagram for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

22.12.13.3Register 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 22.93 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 to this bit, 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 to this bit, 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 22.94 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 to these bits, 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 22.95 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	

22.12.13.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Table 22.96 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 22.93, Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)** and **Table 22.94, 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 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 reloads 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 stops counting and waits for a trigger. 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

22.13 Independent Channel Simultaneous Rewrite Functions

The following describes functions that carry out simultaneous rewrite:

22.13.1 Simultaneous Rewrite Trigger Generation Function Type 1

22.13.1.1 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 22.97, 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 22.37, Simultaneous Rewrite Methods and when They are Triggered**
- 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 written 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 reloads 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.

Conditions

- 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.
- If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated.

22.13.1.2 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 TAUBnCDRm_master + 1 must be an integer. This integer corresponds to the number of interrupts.

Note that the cycle for the triangle PWM is twice the cycle for the PWM

22.13.1.3 Block Diagram and General Timing Diagram

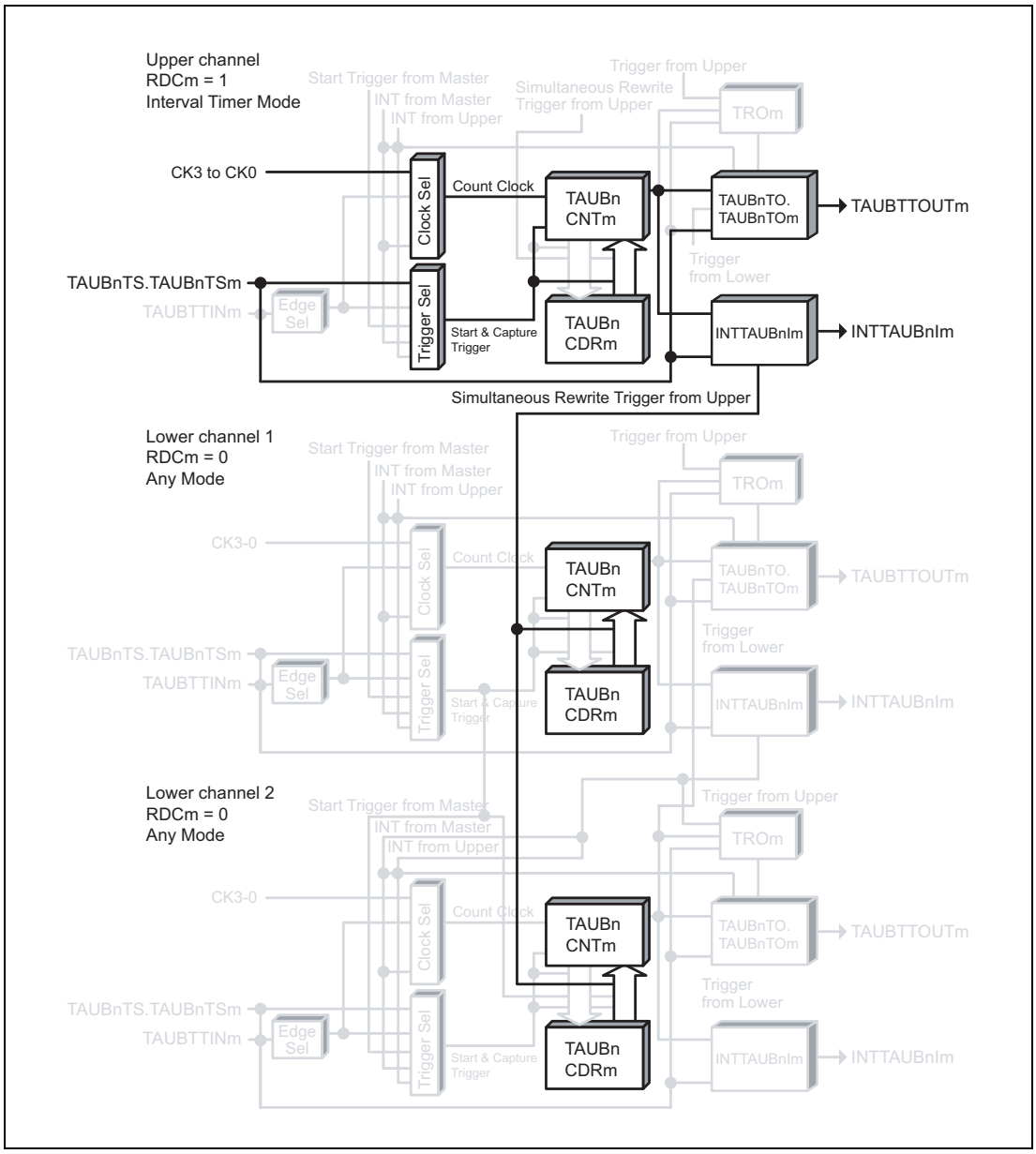


Figure 22.76 Block Diagram for Simultaneous Rewrite Trigger Generation Function Type 1

The following settings apply to the general timing diagram.

- INTTAUBnIm generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

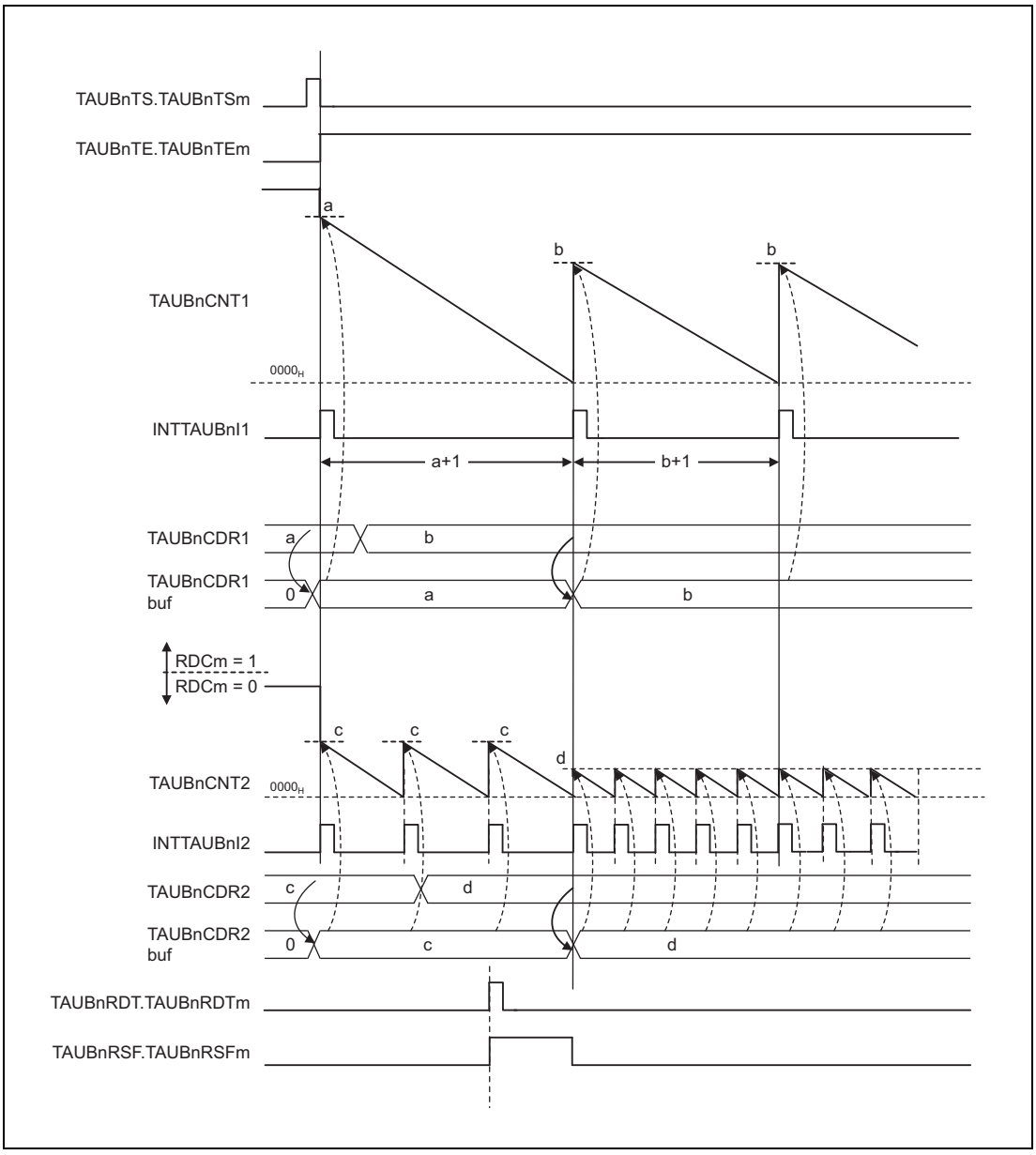


Figure 22.77 General Timing Diagram for Simultaneous Rewrite Trigger Generation Function Type 1

22.13.1.4 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 22.97 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 to this bit, 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 to this bit, 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 22.98 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 to these bits, 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 22.99 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

22.13.1.5 Register Settings for the Lower Channel(s)

(1) Register settings 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 22.37, Simultaneous Rewrite Methods and when They are Triggered**)

(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 22.37, Simultaneous Rewrite Methods and when They are Triggered**)

(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 22.37, Simultaneous Rewrite Methods and when They are Triggered**.

(4) Simultaneous rewrite for the lower channel(s)

Table 22.100 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.

22.13.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Table 22.101 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers for the upper channel as described in **Table 22.97, Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1** and **Table 22.98, Contents of the TAUBnCMURm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1** Set the TAUBnCMORm register and TAUBnCMURm registers for the lower channel as described in **Section 22.13.1.5, 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.MD0 = 1, INTTAUBnIm is generated.
During operation	TAUBnRDT.TAUBnRDTm, TAUBnCDR.CDRm 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 both it and TAUBnTTOUTm retain their current values.

Restart operation

22.14 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 22.2, Overview**.

22.14.1 PWM Output Function

22.14.1.1 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 22.102, 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 22.105, 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 started 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 written to TAUBnCNTm and the counters start to count down from these values. INTTAUBnIm is generated on the master channel and TAUBTTOUTm (slave) toggles, which realizes a PWM output.

- Master channel:

  When the counter of the master channel reaches 0000_H, pulse cycle time has elapsed and INTTAUBnIm is generated. The counter loads the TAUBnCDRm value 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 written 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 22.6, Simultaneous Rewrite**.

22.14.1.2 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

22.14.1.3 Block Diagram and General Timing Diagram

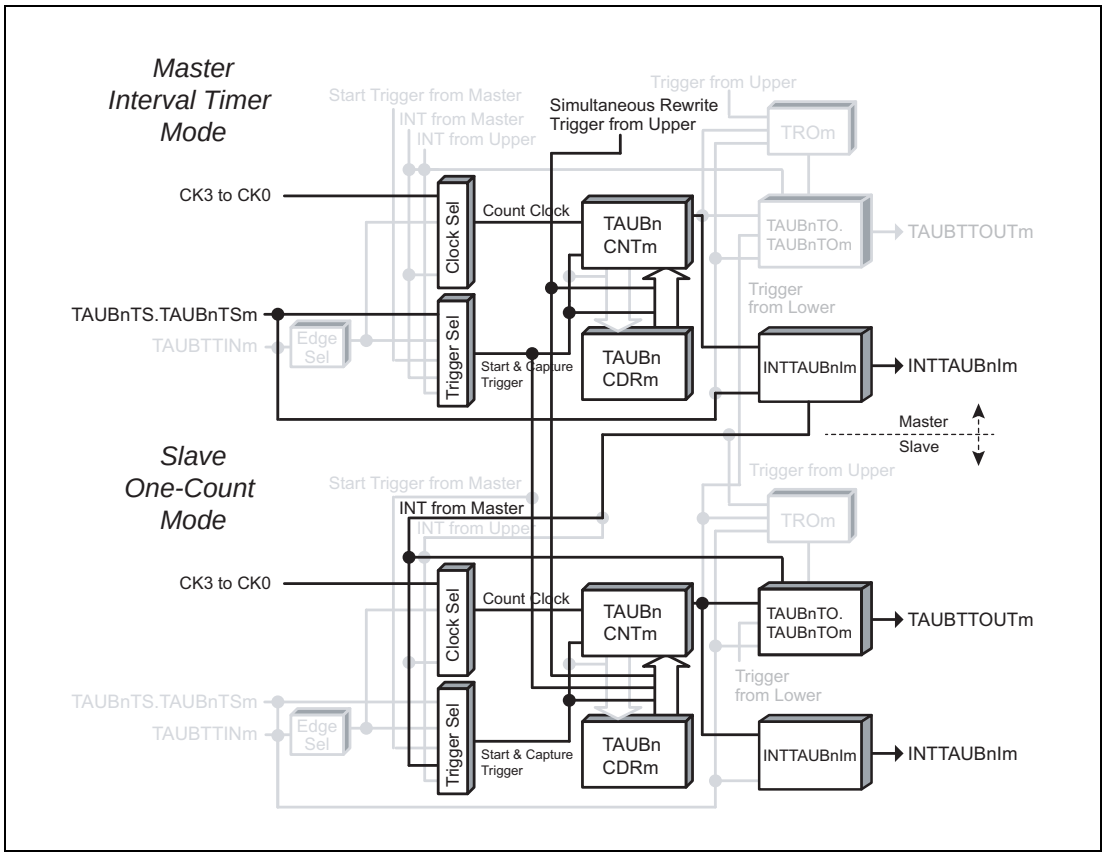


Figure 22.78 Block Diagram for PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUBnTOL.TAUBnTOLm = 0)

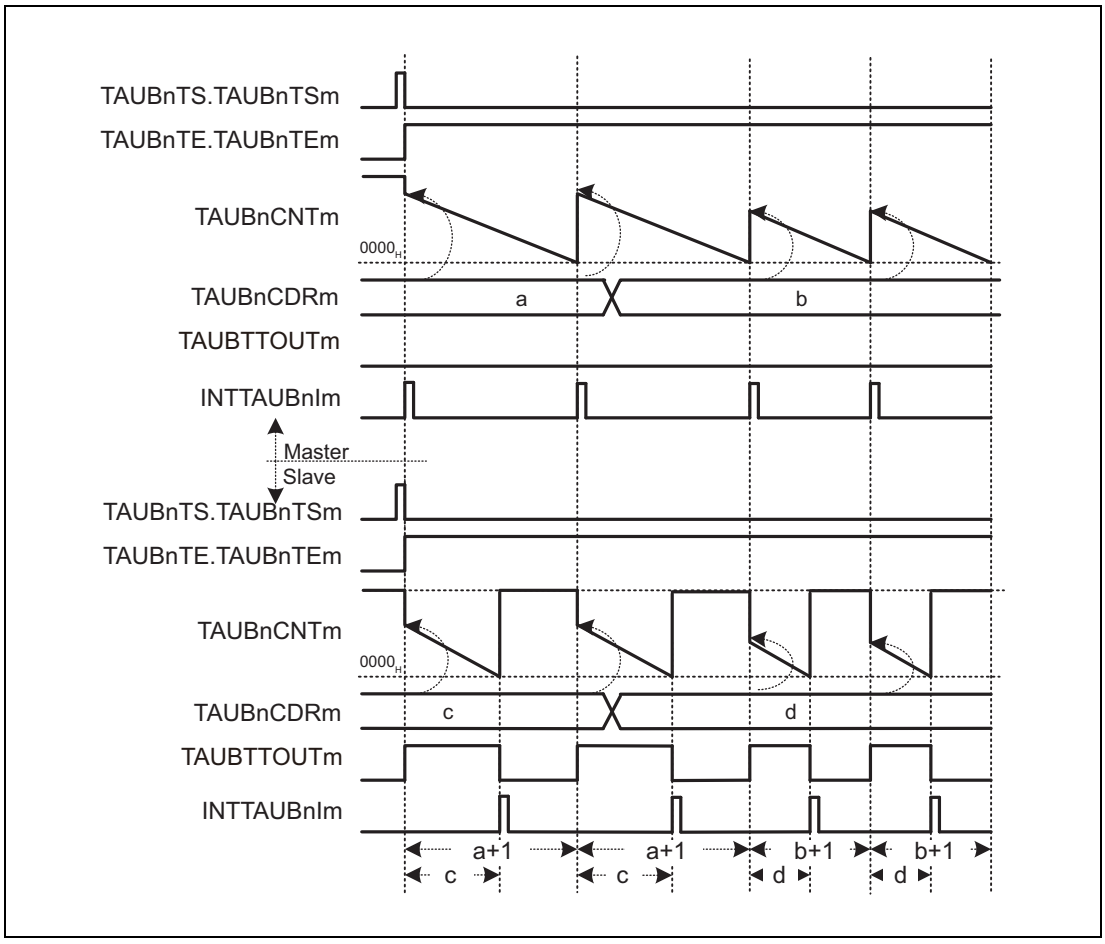


Figure 22.79 General Timing Diagram for PWM Output Function

NOTE

The interval between the slave channel starting to count and an interrupt being generated is the value of corresponding TAUBnCDRm, whereas for the master channel the interval is the corresponding TAUBnCDRm + 1.

22.14.1.4 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 22.102 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 CKS[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 to this bit, 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 to this bit, 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 22.103 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 to these bits, 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 channel must be identical.

Table 22.104 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 22.13.1, 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, TAUBnRDS = 1
  The setting value of TAUBnCDR to this channel is as follows.
  = ((setting value of TAUBnCDR of the master channel subject to simultaneous rewrite + 1) × number of interrupts) – 1
- Master channel: TAUBnRDCm = 0, TAUBnRDS = 1
- Slave channel: TAUBnRDCm = 0, TAUBnRDS = 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%.

22.14.1.5 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 22.105 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 CKS[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 to this bit, 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 to this bit, 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 22.106 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 to these bits, 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 22.107 Control Bit Settings for Independent 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 channel must be identical.

Table 22.108 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.

22.14.1.6 Operating Procedure for PWM Output Function

Table 22.109 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 22.14.1.4, Register Settings for the Master Channel**. Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 22.14.1.5, 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 set When TAUBnCNTm (slave) reaches 0000_H: • The counter of TAUBnCNTm (slave) stops. • INTTAUBnIm (slave) is generated • TAUBTTOUTm (slave) 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.

Restart operation

22.14.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

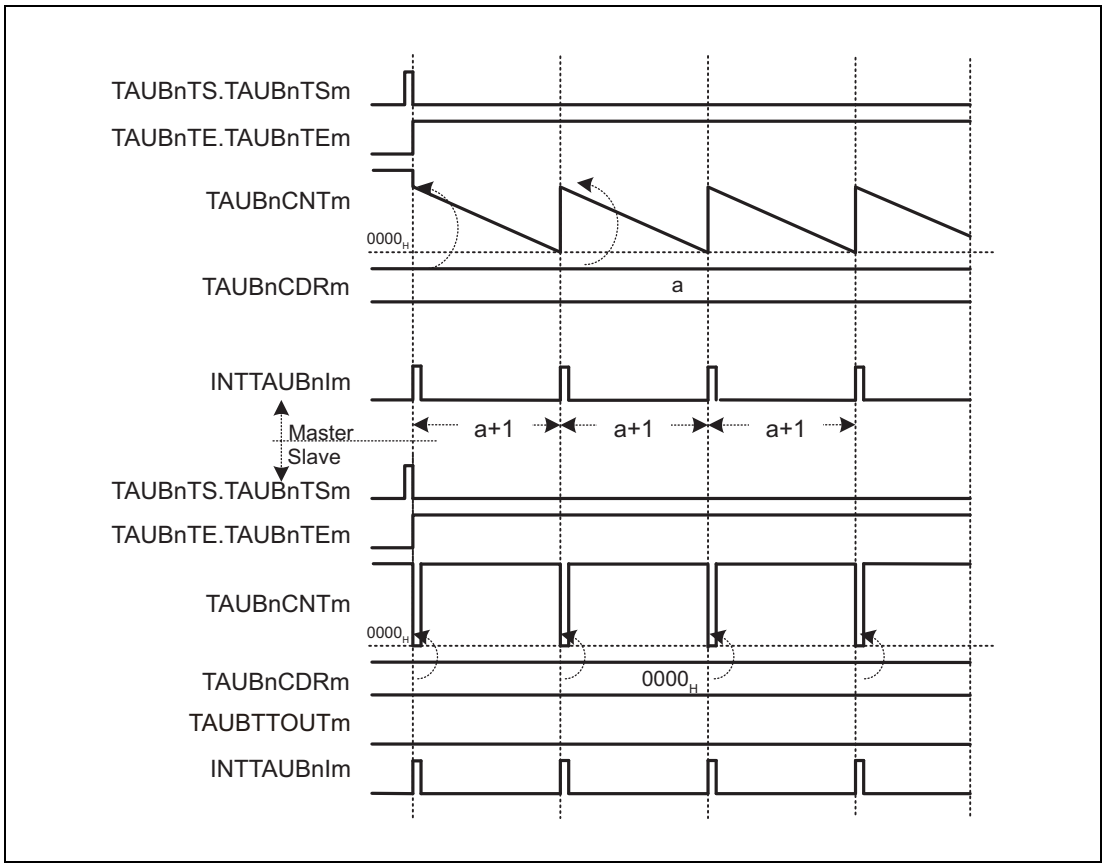


Figure 22.80 TAUBnCDRm (slave) = 0000_H, Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

- Every time the master channel generates an interrupt (INTTAUBnIm), 0000_H is written 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%

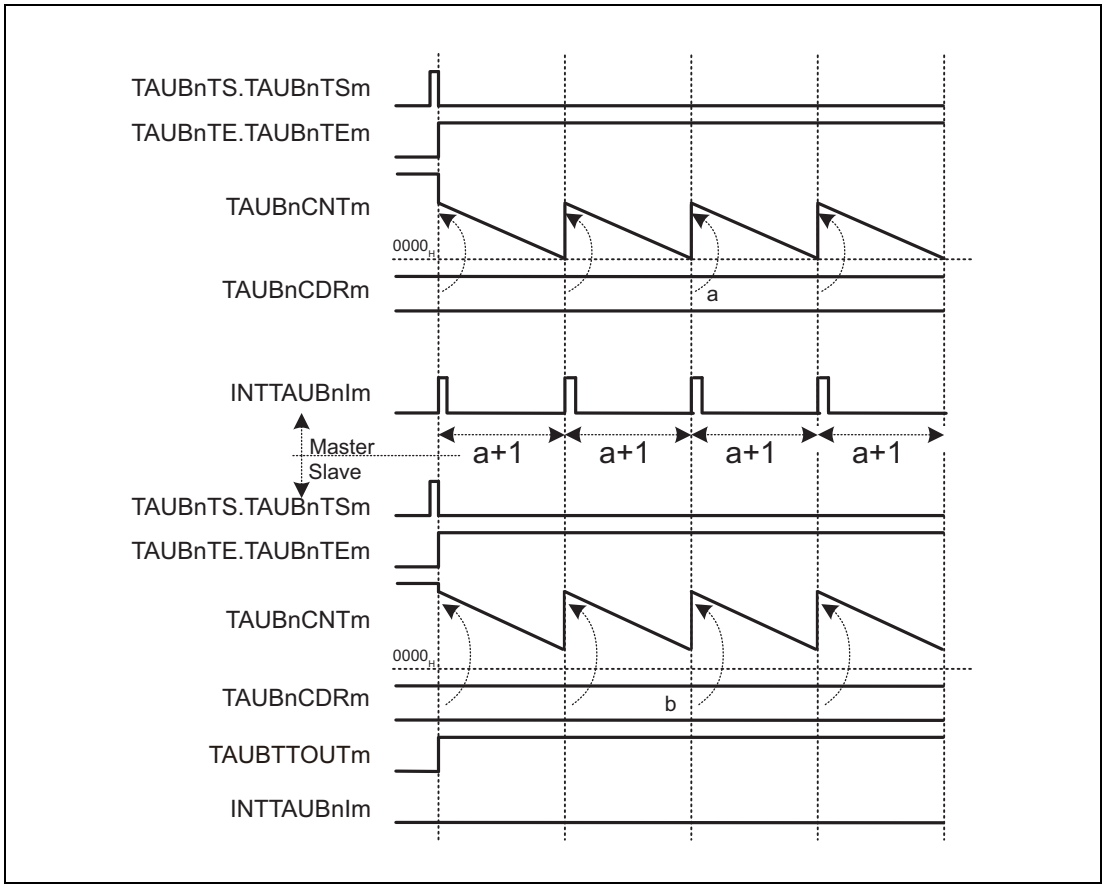


Figure 22.81 TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1, Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

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) Stop and restart operation

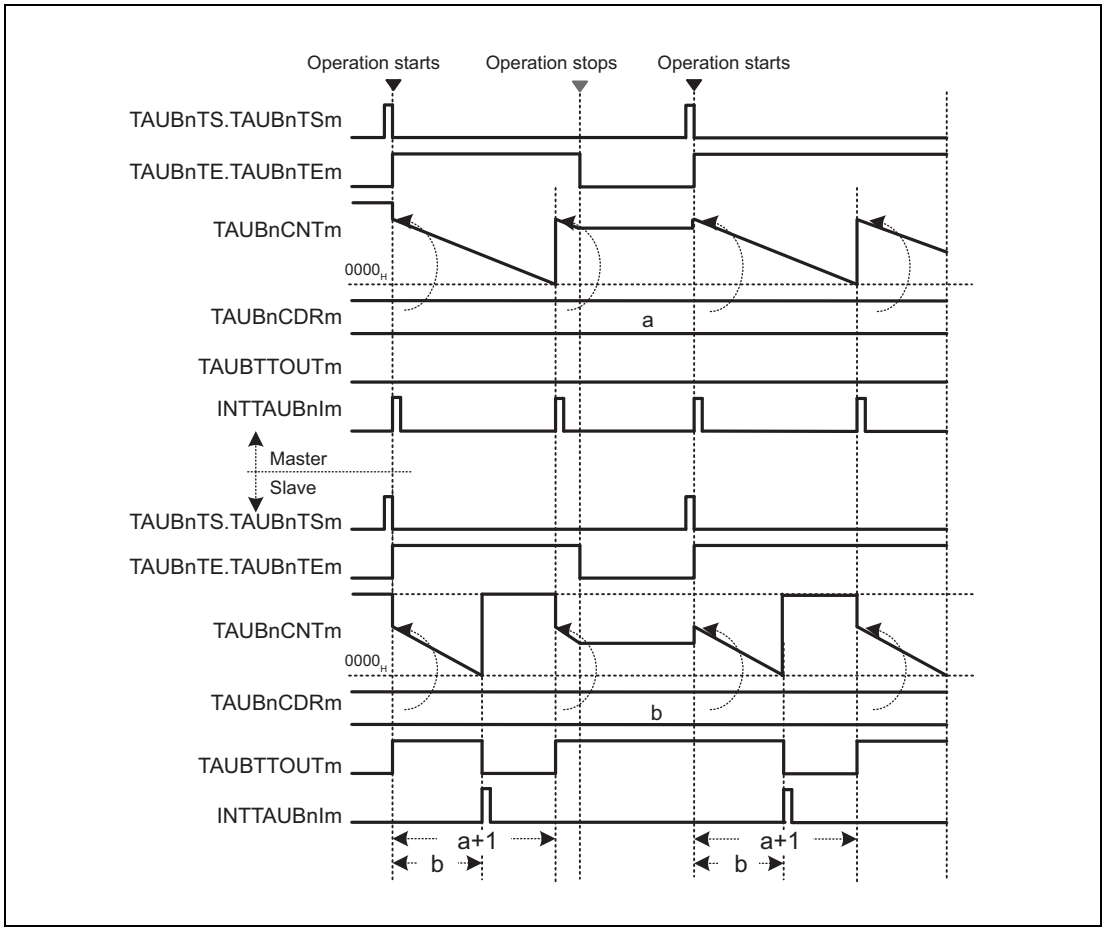


Figure 22.82 Stop and Restart Operation, Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

- 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) Stopping and restarting the operation (Slave output, Initialization)

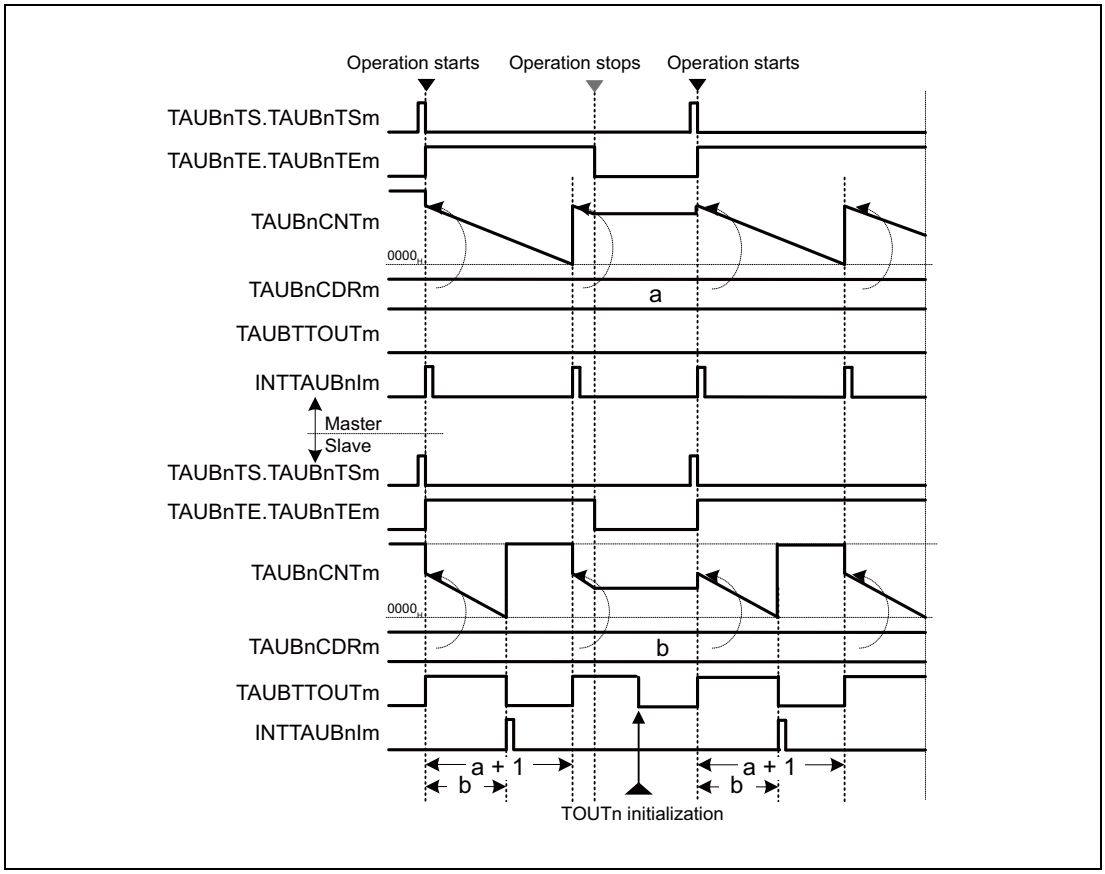


Figure 22.83 Operation Stop and Restart (Slave Output, Initialization)

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 when the count operation is started after restart.

22.14.2 One-Shot Pulse Output Function

22.14.2.1 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 22.112, Simultaneous Rewrite Settings 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 22.116, Simultaneous Rewrite Settings 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) 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 written 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 returns 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 written 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 remains 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 channel, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel 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 slave channel is executed during operation, the width of the output signal does not correspond to the value of TAUBnCDRm (slave).
2. The input TAUBTTINm 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 22.6, Simultaneous Rewrite**.

22.14.2.2 Equations

Delay from trigger input to pulse output
= (TAUBnCDRm (master) + 1) × count clock cycle

Pulse width = (TAUBnCDRm (slave)) × count clock cycle

22.14.2.3 Block Diagram and General Timing Diagram

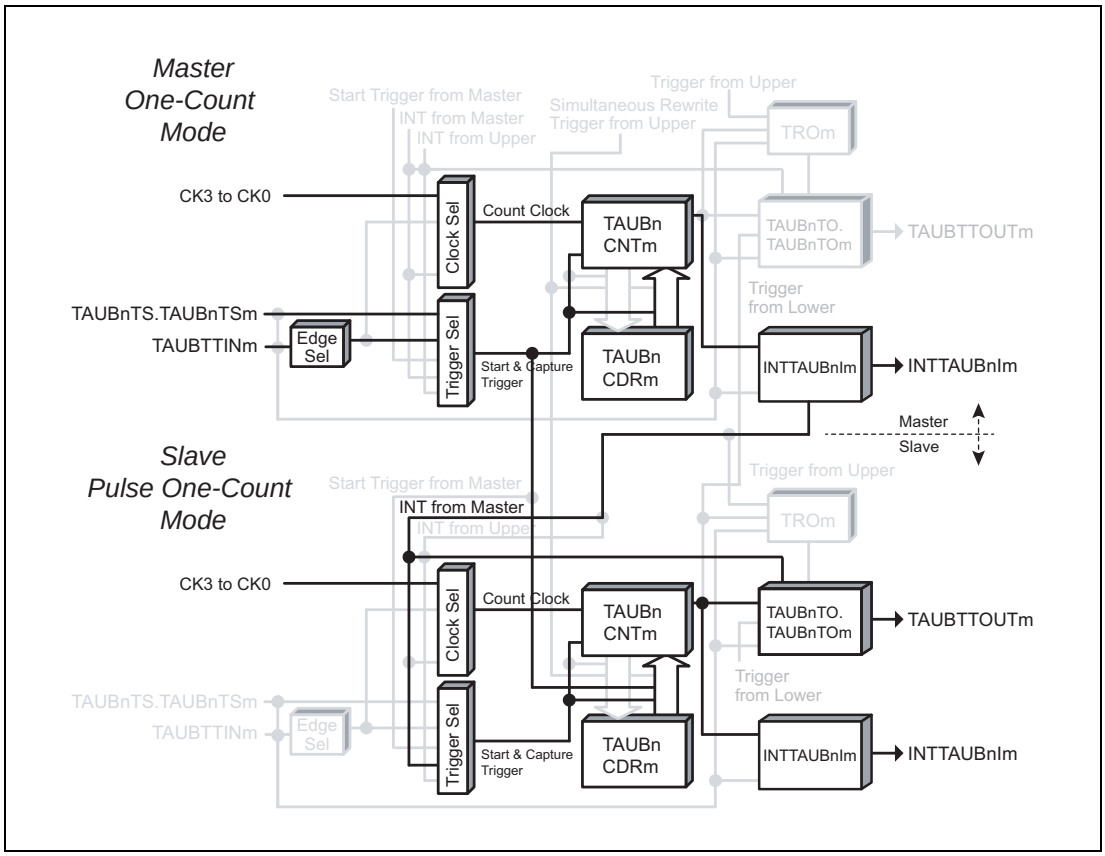


Figure 22.84 Block Diagram for One-Shot Pulse Output Function

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)

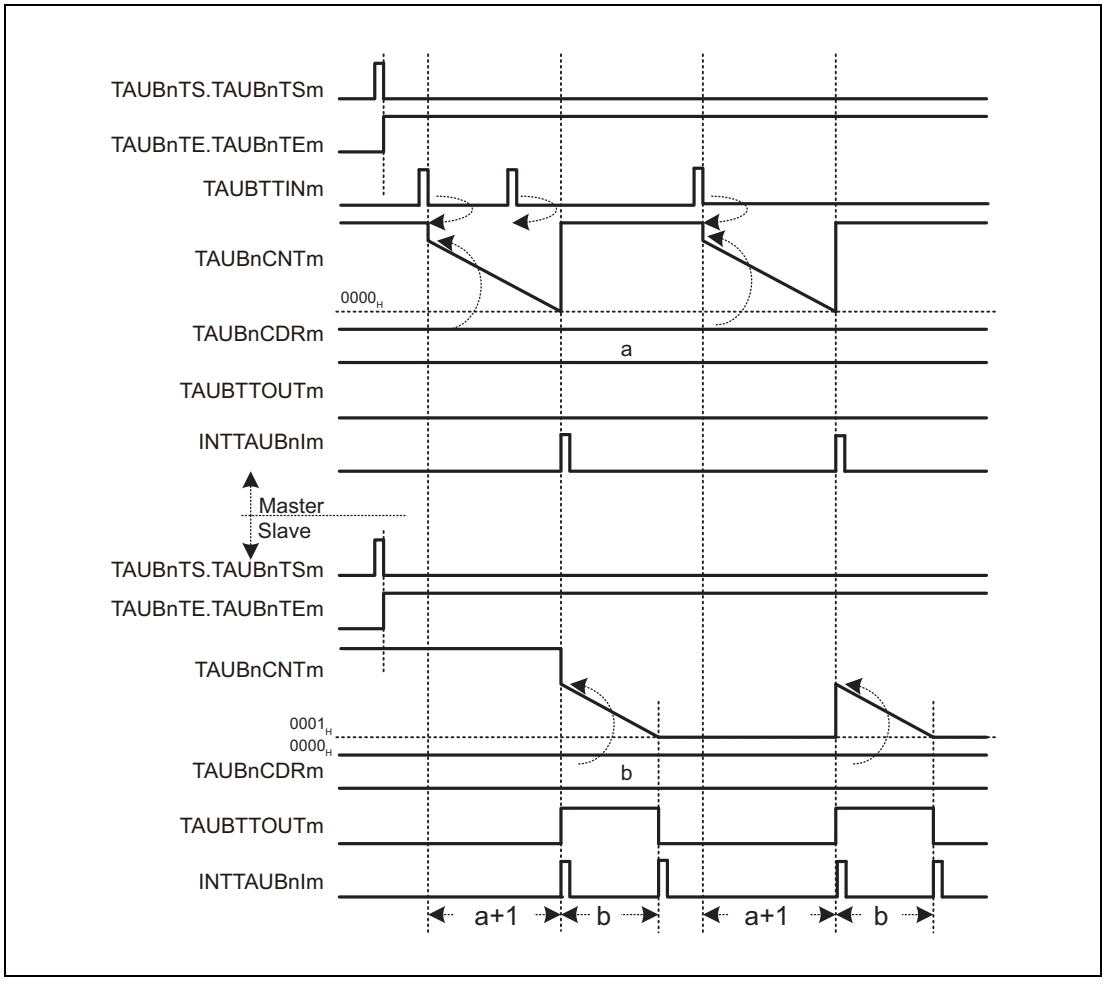


Figure 22.85 General Timing Diagram for One-Shot Pulse Output Function

22.14.2.4 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 22.110 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
13	Reserved	When read, the value after reset is returned. When writing to this bit, 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 to this bit, 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 22.111 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 to these bits, 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 channel must be identical.

Table 22.112 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.

22.14.2.5 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 22.113 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 channel must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, 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 to this bit, 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 channel 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 22.114 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 to these bits, 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 22.115 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 real-time output 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 channel must be identical.

Table 22.116 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.

22.14.2.6 Operating Procedure for One-Shot Pulse Output Function

Table 22.117 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 22.14.2.4, Register Settings for the Master Channel**. Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 22.14.2.5, 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 AUBnCDRm 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 TAUBnTTINm input edge. • TAUBnCNTm (slave) reloads 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

22.14.2.7 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)

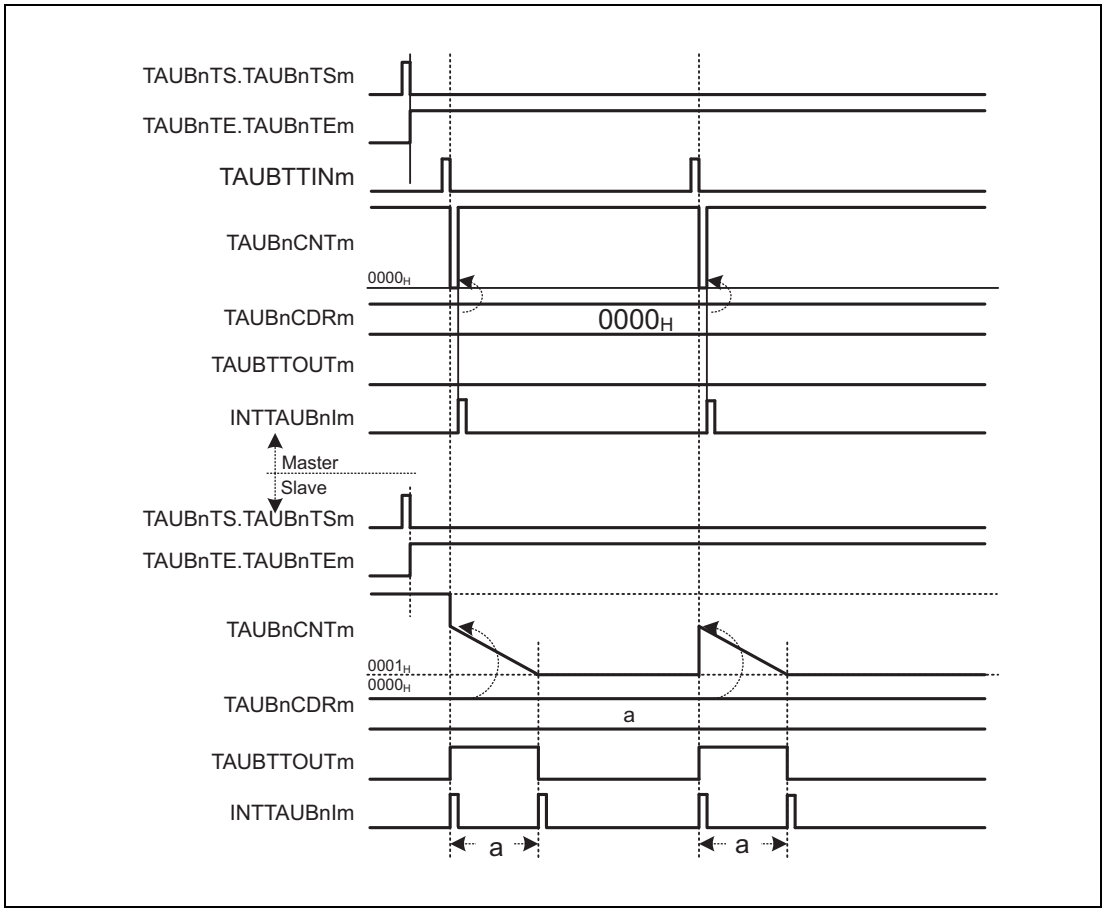


Figure 22.86 TAUBnCDRm (master) = 0000_H

- 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)

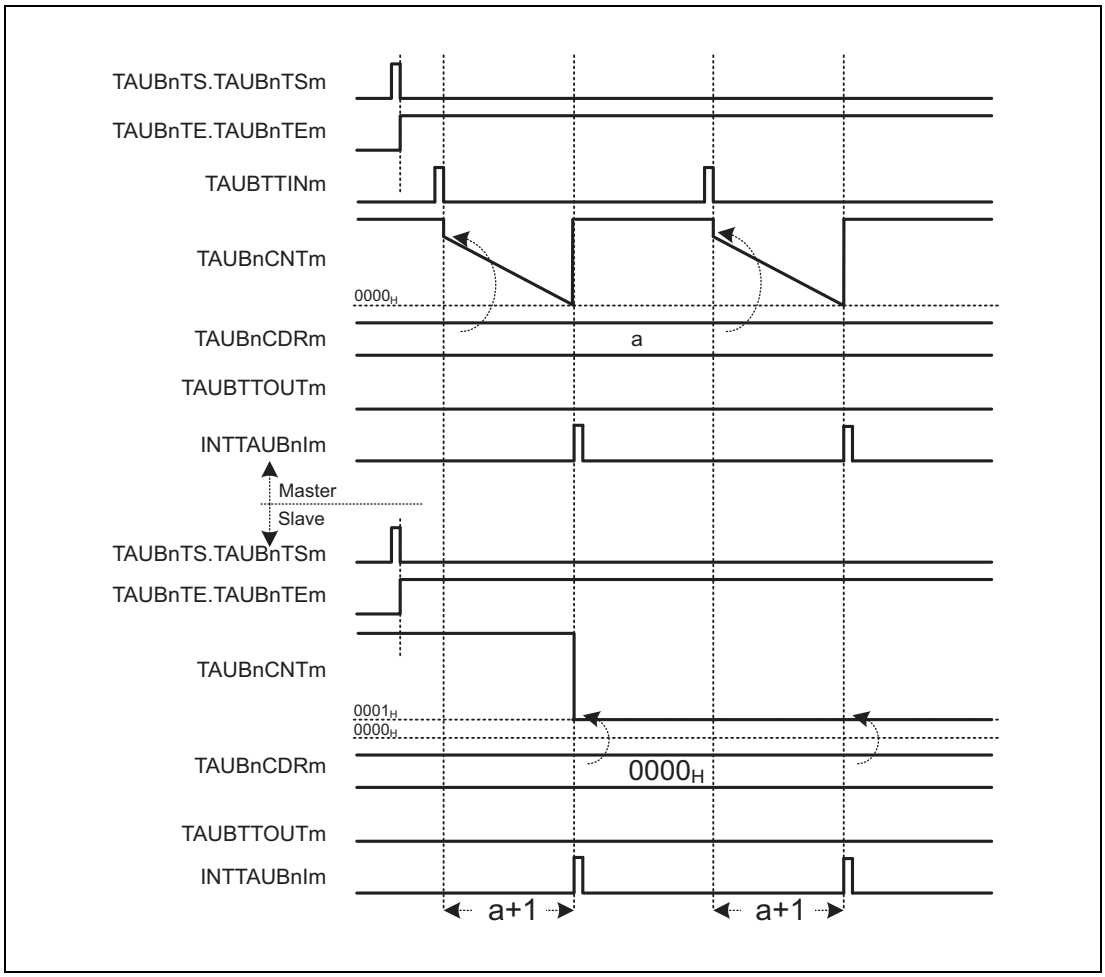


Figure 22.87 TAUBnCDRm (slave) = 0000_H

- TAUBTTOUTm remains at not active state, because the pulse width is zero.

(3) TAUBnCMORm.TAUBnMD0 = 0 (disables start trigger during count operation)

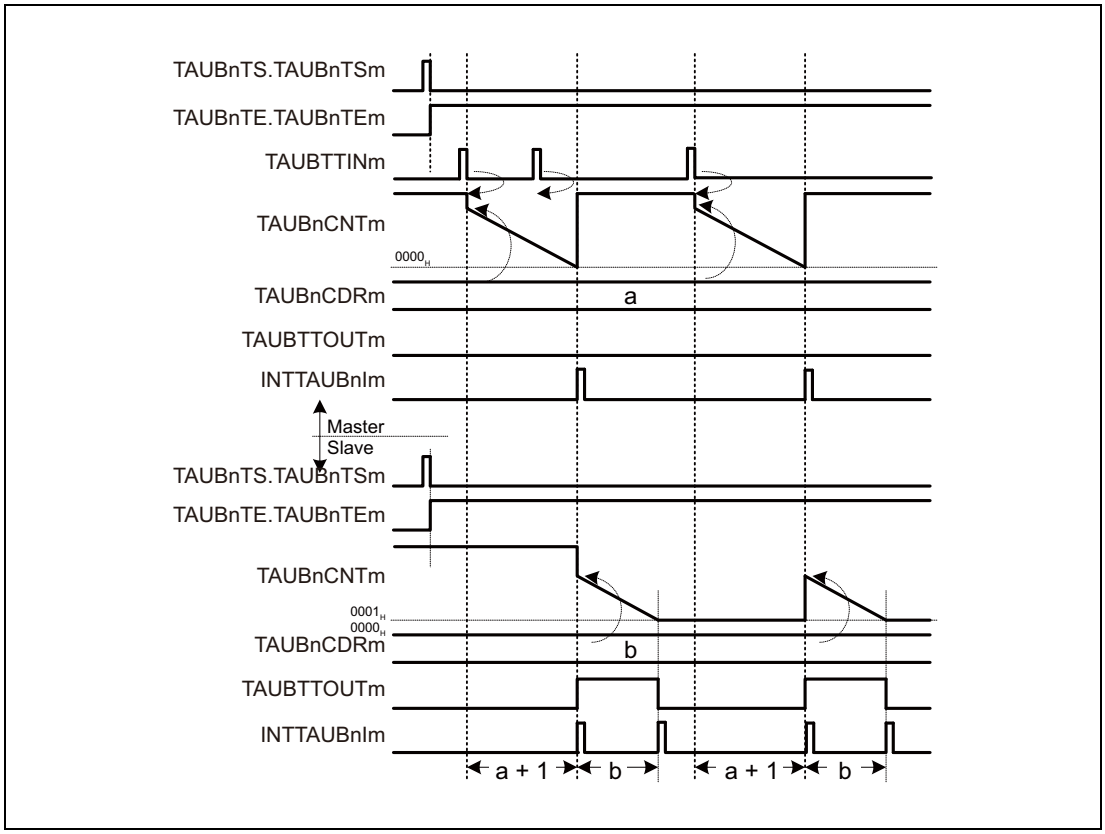


Figure 22.88 TAUBnCMORm.TAUBnMD0 = 0

(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)

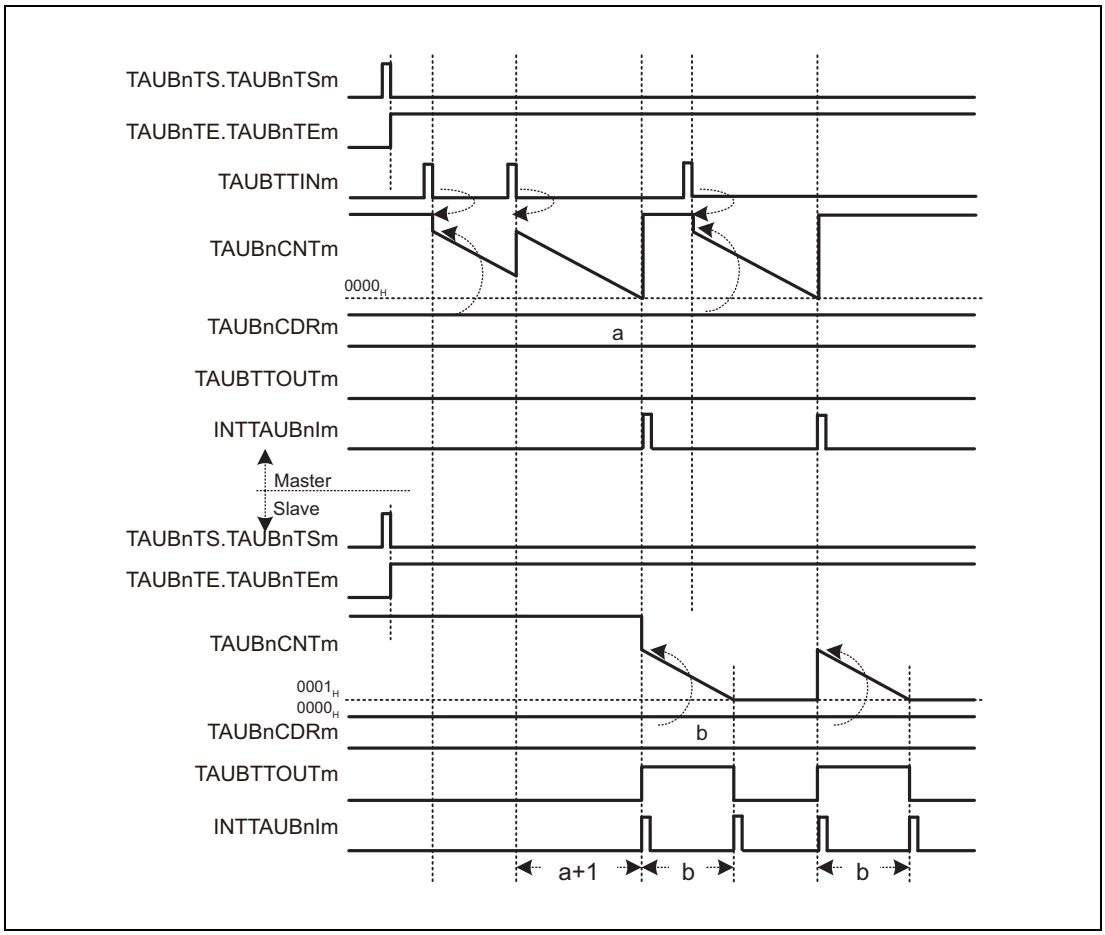


Figure 22.89 TAUBnCMORm.TAUBnMD0 = 1

- 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 is extended by the value of TAUBnCNTm at the time when a valid TAUBTTINm input edge is detected.

(5) Stopping and restarting the operation

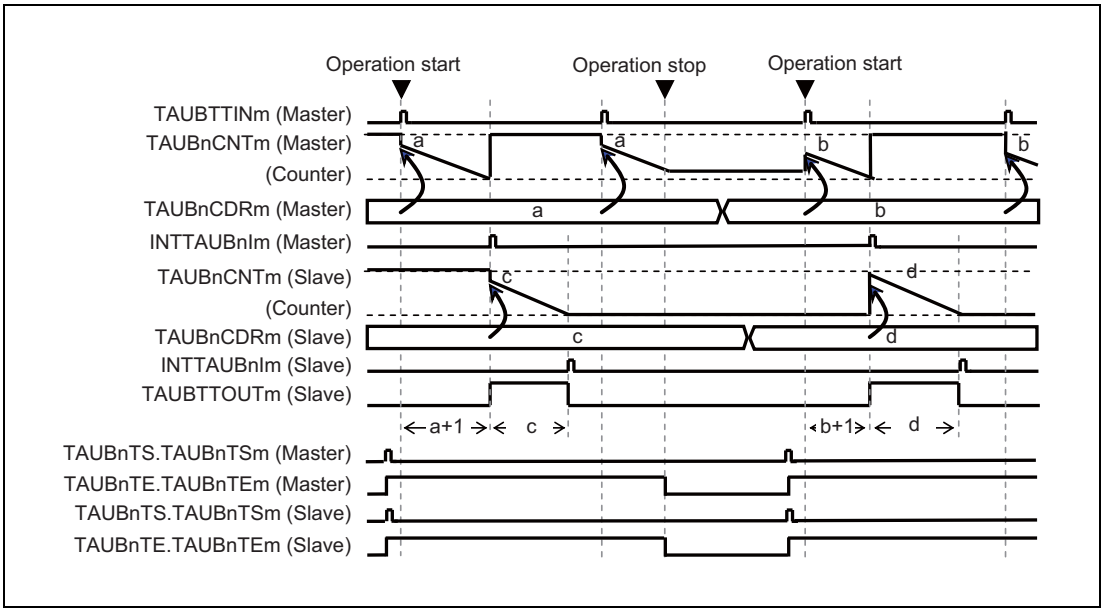


Figure 22.90 Stopping and Restarting the Operation

Setting TTm of the master and slave channels to 1 concurrently sets 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 concurrently 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)

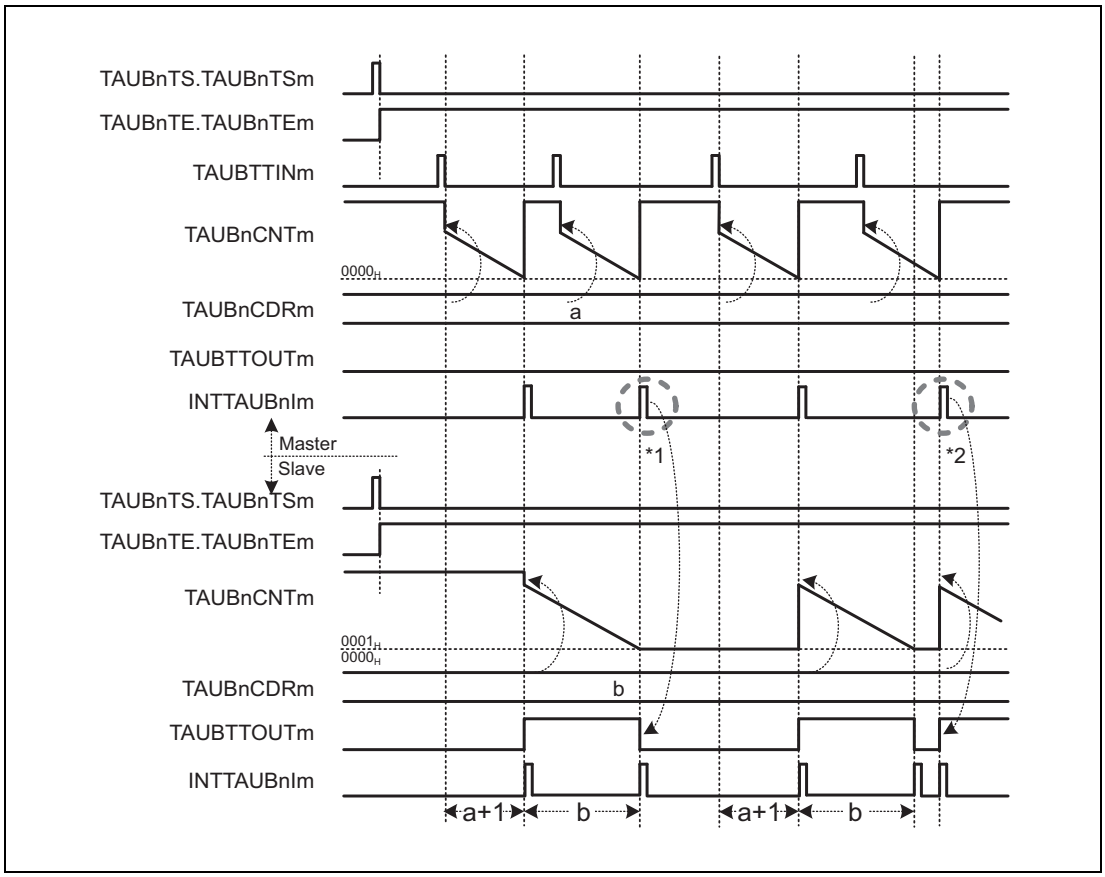


Figure 22.91 Interval of TAUBTTINm ≤ Delay Time + Pulse Width + 1

- 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 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.

22.14.3 Delay Pulse Output Function

22.14.3.1 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 values of TAUBnCDRm of these both channels have to be identical.
- 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 22.118, 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 22.121, Contents of the TAUBnCMORm Register for the Slave Channel 1 of the Delay Pulse Output Function**.
- The operation mode of slave channel 3 must be set to pulse one-count mode, see **Table 22.125, Contents of the TAUBnCMORm Register for the Slave Channel 2 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 started by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm, enabling count operation.

- Master channel:

  The current value of TAUBnCDRm is written 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, pulse cycle time has elapsed and 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 returns 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 returns 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 22.6, Simultaneous Rewrite**.

22.14.3.2 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).)

22.14.3.3 Block Diagram and General Timing Diagram

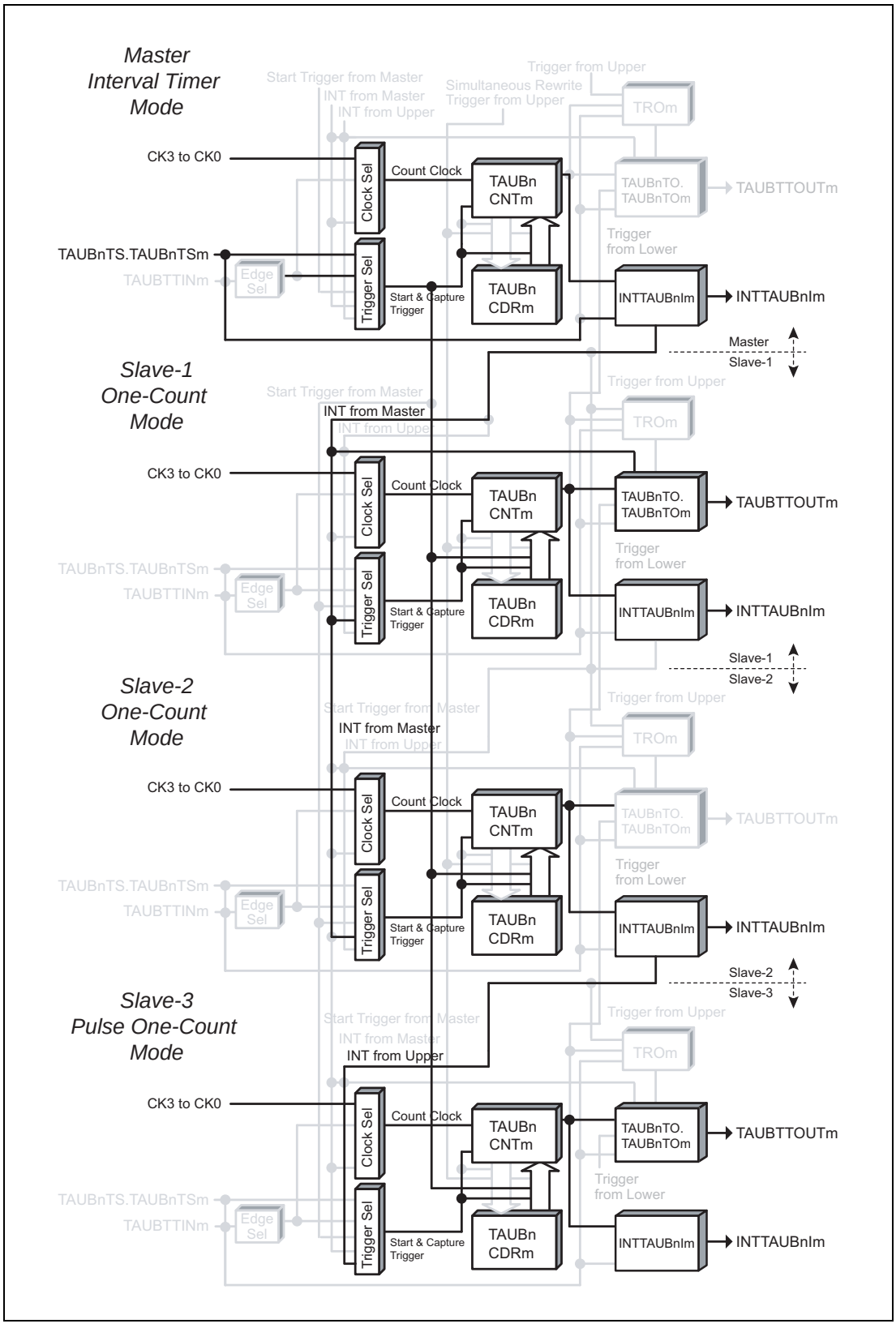


Figure 22.92 Block Diagram for Delay Pulse Output Function

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)

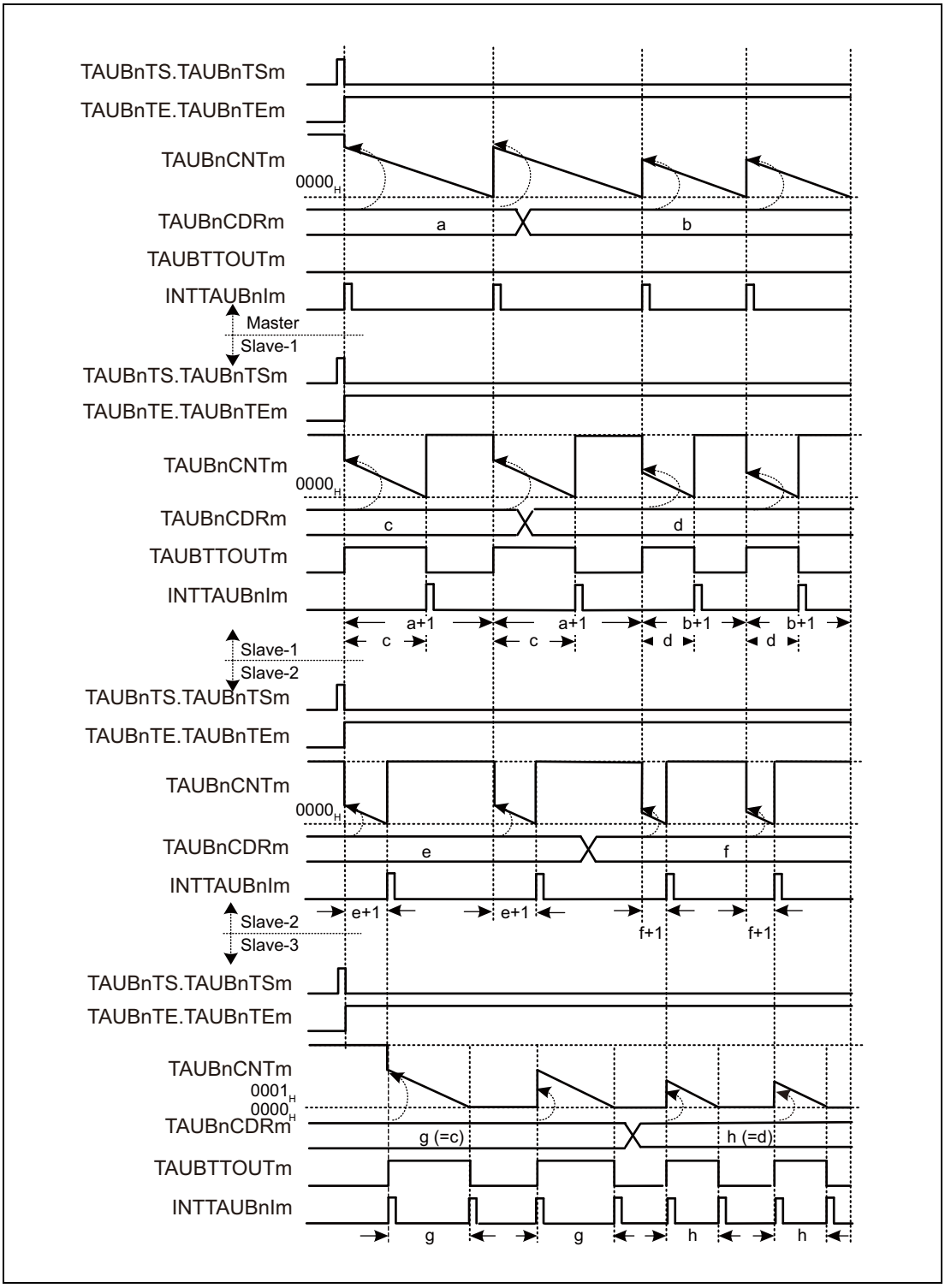


Figure 22.93 General Timing Diagram for Delay Pulse Output Function

22.14.3.4 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 22.118 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 channel must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, 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 to this bit, 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 22.119 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 to these bits, 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 22.120 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.

22.14.3.5 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 22.121 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 channel must be identical.
13	Reserved	When writing to this bit, 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 writing to this bit, 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 22.122 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 writing to these bits, 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 22.123 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 22.124 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.

22.14.3.6 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 22.125 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 channel must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, 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 to this bit, 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 22.126 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 to these bits, 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 22.127 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.

22.14.3.7 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 22.128 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 channel must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, 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 to this bit, 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 22.129 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 to these bits, 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 22.130 Control Bit Settings for Synchronous 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 22.131 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.

22.14.3.8 Operating Procedure for Delay Pulse Output Function

Table 22.132 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 22.14.3.4, Register Settings for the Master Channel**. Slave channel 1: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 22.14.3.5, Register Settings for Slave Channel 1**. Slave channel 2: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 22.14.3.6, Register Settings For Slave Channel 2**. Slave channel 3: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 22.14.3.7, Register Settings for Slave Channel 3**. Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.

Table 22.132 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.TSm 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) reloads 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.

22.14.3.9 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) = $000B_H$

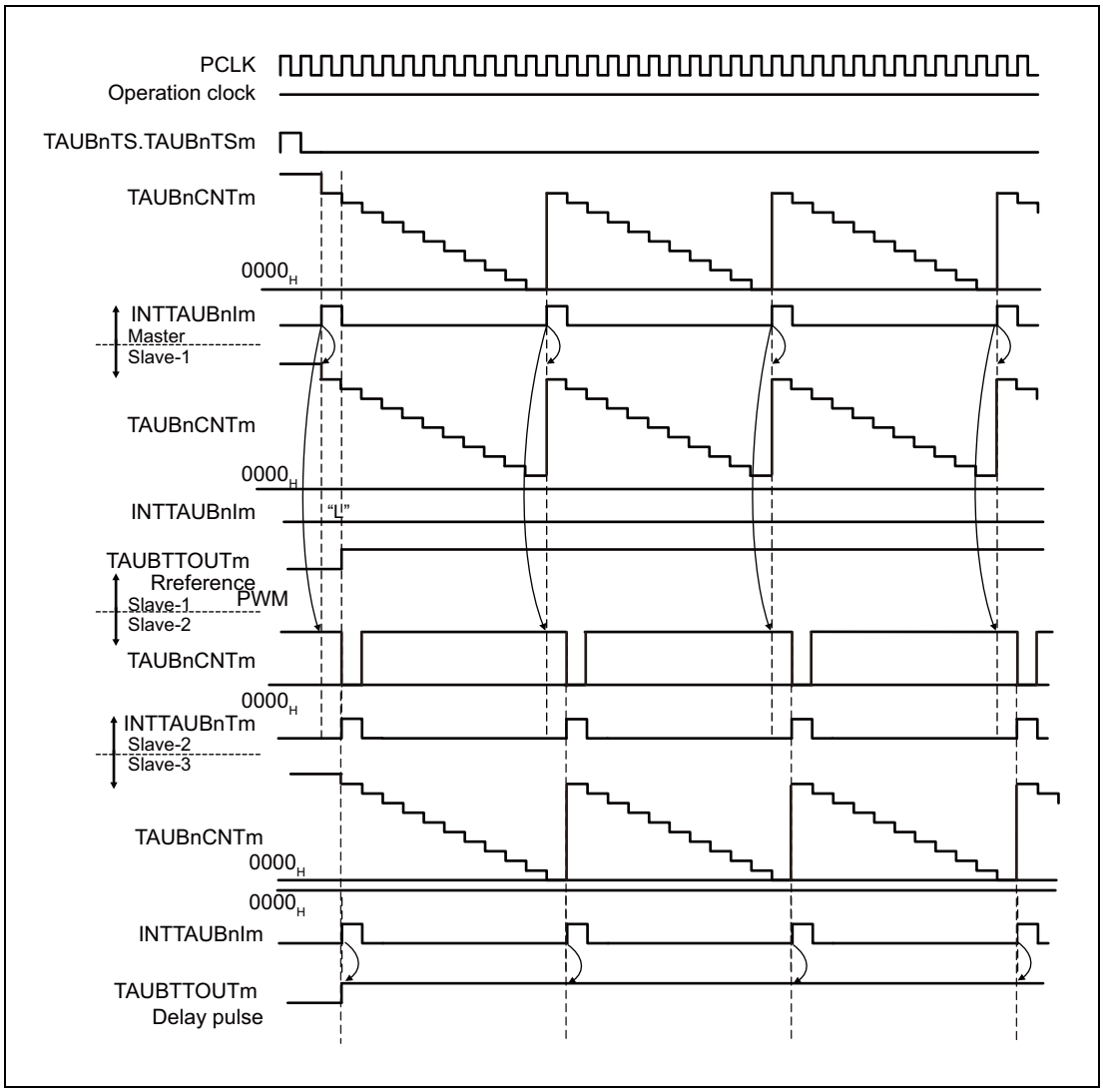


Figure 22.94 Duty Cycle (slave 3) = 100%

- If the value of TAUBnCDRm (slave 1 and 3) is higher than the value of TAUBnCDRm (master), the counter of the slave channels 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

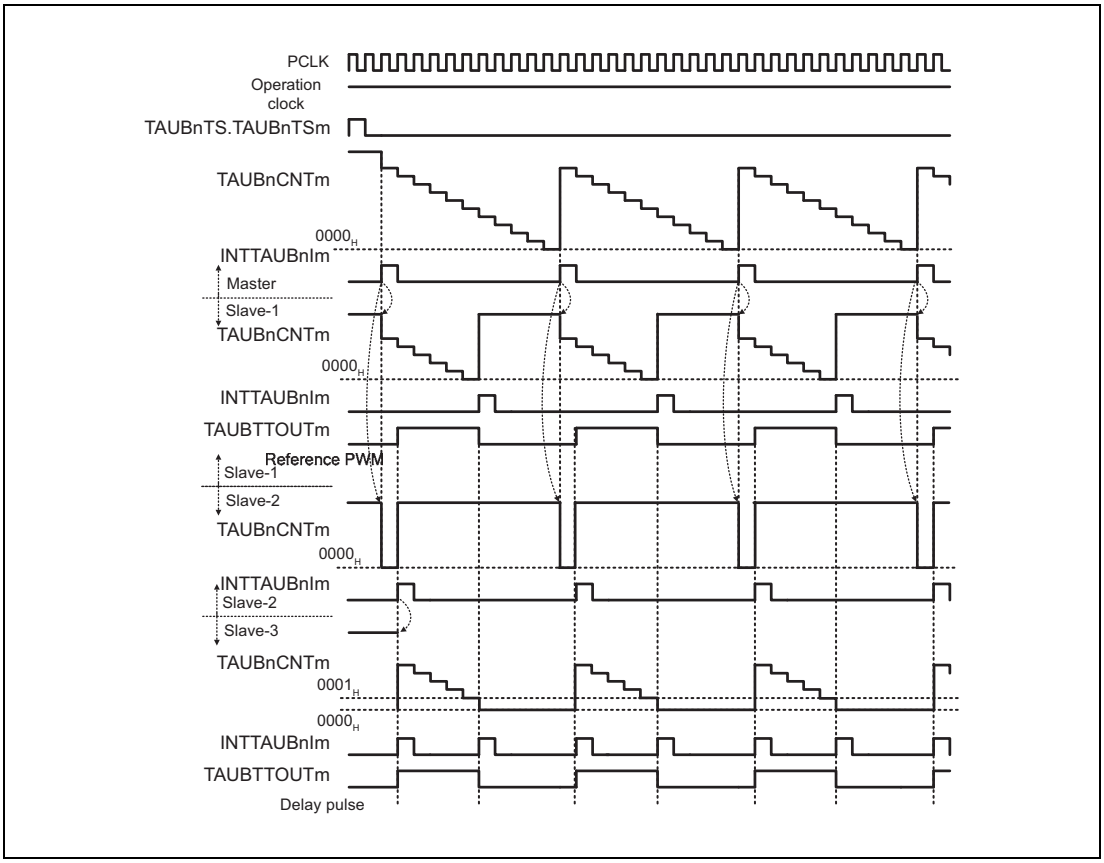


Figure 22.95 TAUBTTOUTm (slave 1) = TAUBTTOUTm (slave 3)

- 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.

22.14.4 AD Conversion Trigger Output Function Type 1

22.14.4.1 Overview

Summary

This function is identical to **Section 22.14.1, 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.

22.14.4.2 Block Diagram and General Timing Diagram

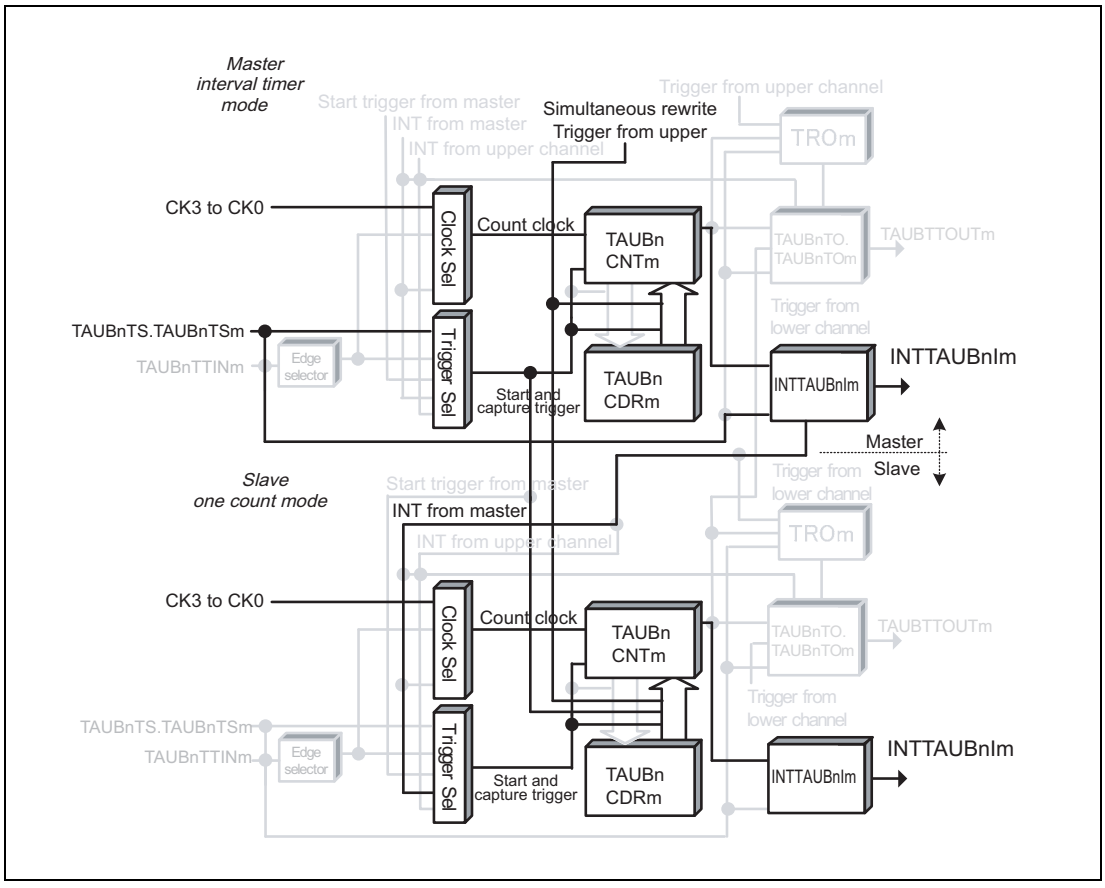


Figure 22.96 Block Diagram for AD Conversion Trigger Output Function Type 1

The following settings apply to the general timing diagram.

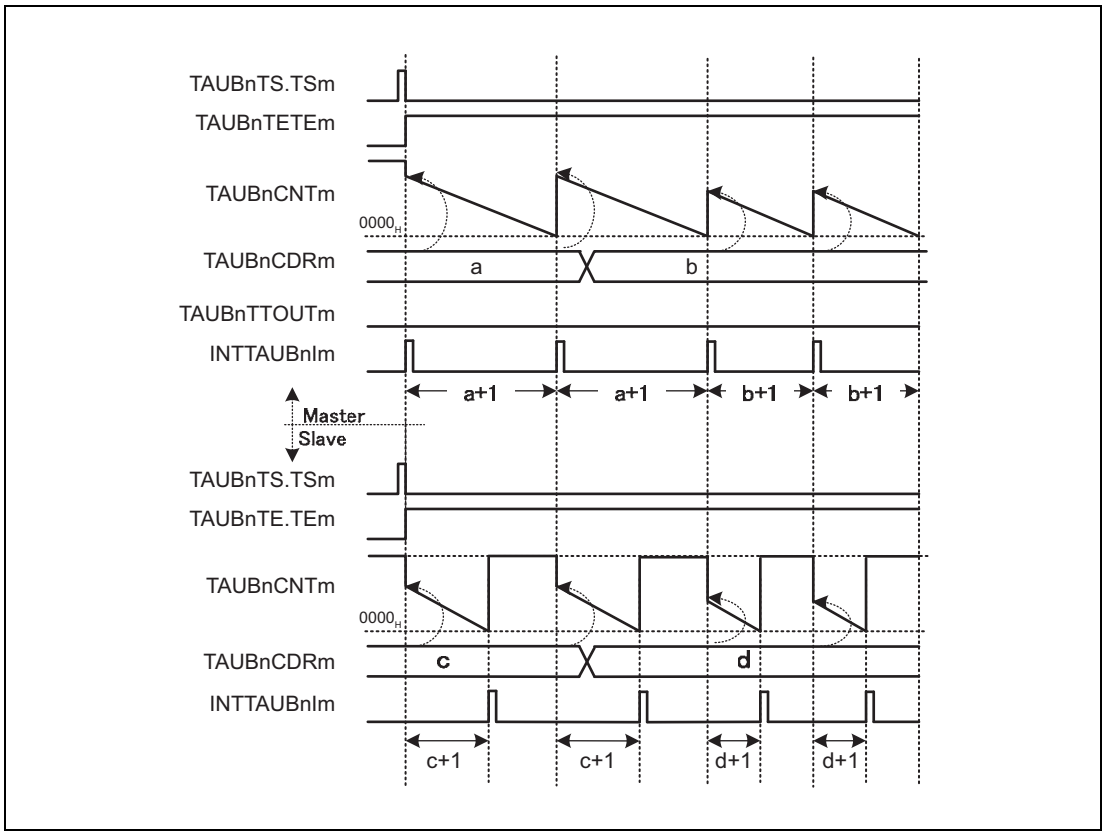


Figure 22.97 General Timing Diagram for AD Conversion Trigger Output Function Type 1

22.14.5 Triangle PWM Output Function

22.14.5.1 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 TAUBTTOUTm to be set using the master and slave channel(s) respectively.

The master channel generates a carrier cycle from two pulse cycles. 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 22.133, Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function**.
- The operation mode of the slave channel(s) must be set to up down count mode, see **Table 22.134, Contents of the TAUBnCMURm Register for the Master Channel of the Triangle PWM Output Function**.
- The channel output mode of the master channel must be set to independent channel output mode 1.
- The channel 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 started 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 written 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 22.6, Simultaneous Rewrite**.

22.14.5.2 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

22.14.5.3 Block Diagram and General Timing Diagram

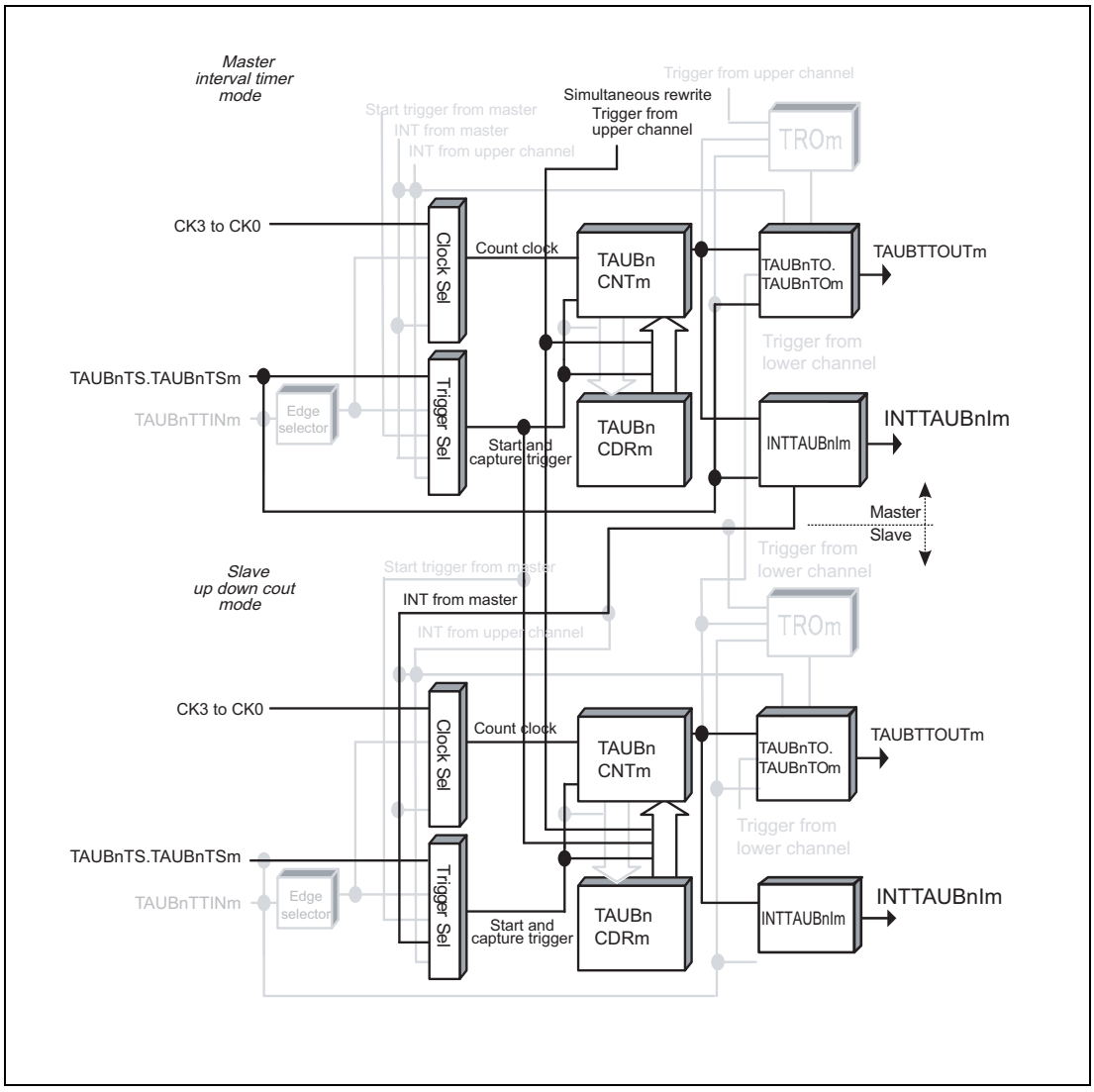


Figure 22.98 Block Diagram for Triangle PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

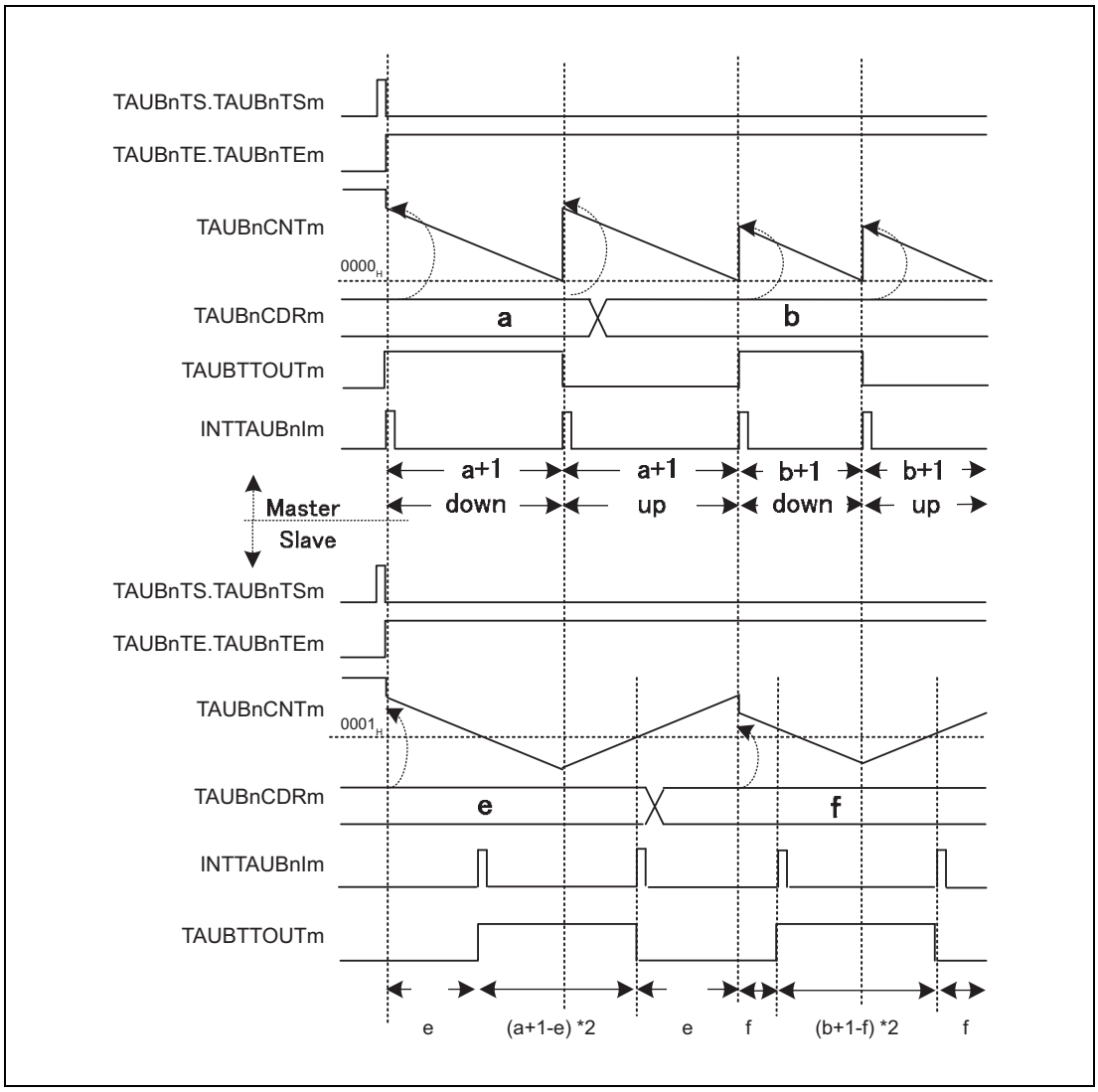


Figure 22.99 General Timing Diagram for Triangle PWM Output Function

22.14.5.4 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 22.133 Contents of the TAUBnCMORm Register for the Master Channel of the Triangle 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 to this bit, 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 to this bit, 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 22.134 Contents of the TAUBnCMURm 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 to these bits, 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 22.135 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 channel must be identical.

Table 22.136 Simultaneous Rewrite Settings for the Master Channel of the Triangle 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.

22.14.5.5 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 22.137 Contents of the TAUBnCMORm Register for the Slave Channel of the Triangle 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 to this bit, 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 to this bit, 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 22.138 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 to these bits, 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 22.139 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 channel must be identical.

Table 22.140 Simultaneous Rewrite Settings for the Slave Channel of the Triangle 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.

22.14.5.6 Operating Procedure for Triangle PWM Output Function

Table 22.141 Operating Procedure for Triangle 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 22.14.5.4, Register Settings for the Master Channel** Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 22.14.5.5, 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 channel loads TAUBnCDRm and counts down. When the counter of the master channel reaches 0000_H: • INTTAUBnIm (master) is generated • TAUBnTTOUTm (master) toggles • TAUBnCNTm (master) reloads the TAUBnCDRm value and continues count operation. • TAUBnCNTm (slave) reloads 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

22.14.5.7 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

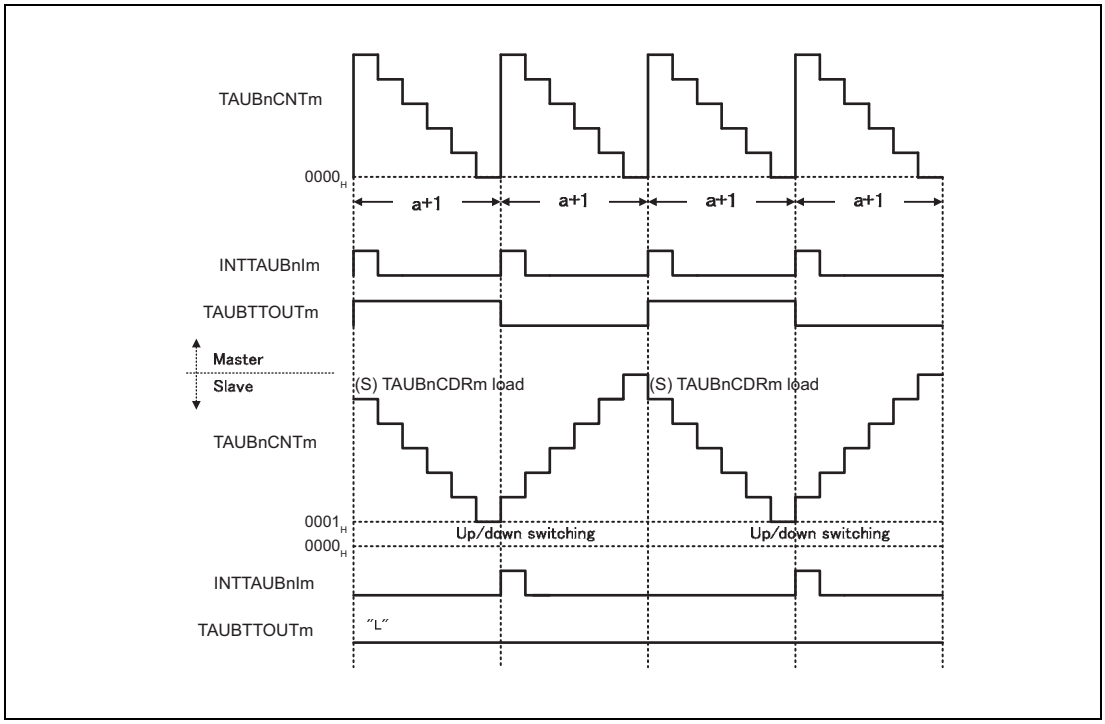


Figure 22.100 TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1

- If TAUBnCDRm (slave) ≥ TAUBnCDRm (master), the counter of slave channel cannot reach 0001_H 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

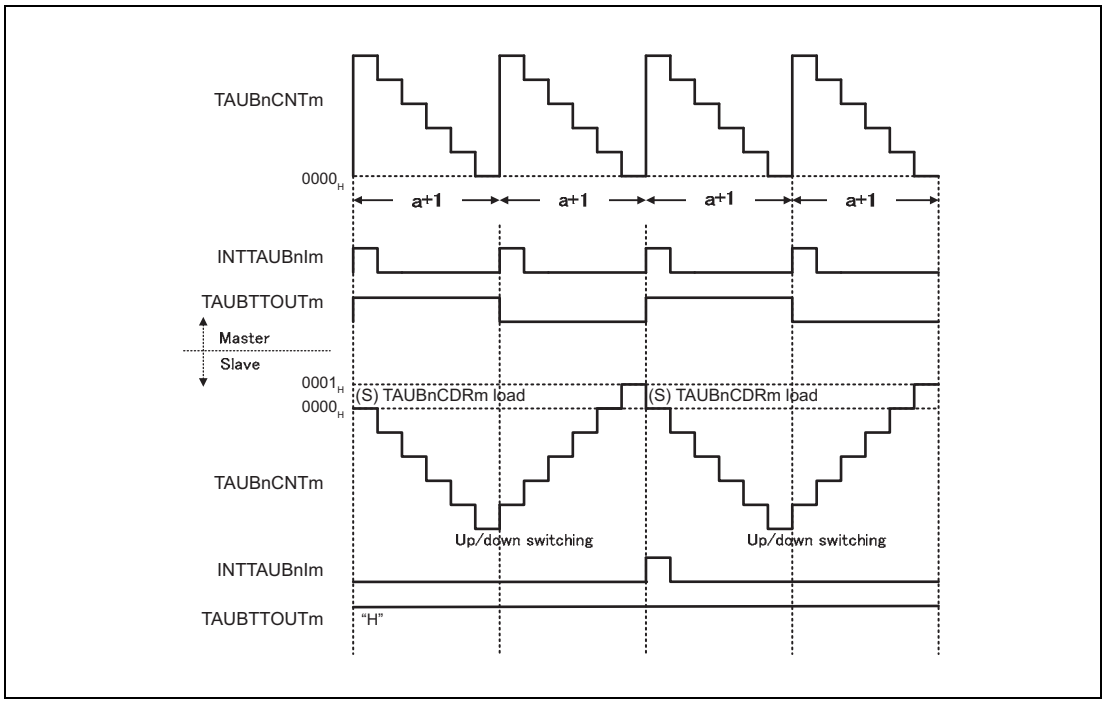


Figure 22.101 TAUBnCDRm (slave) = 0000_H

- If TAUBnCDRm (slave) = 0000_H, the counter of slave channel cannot reach 0001_H during counting up. The reset signal is never detected, so TAUBTTOUTm remains at high state.

22.14.6 Triangle PWM Output Function with Dead Time

22.14.6.1 Overview

Summary

This function generates multiple triangle 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 22.143, Contents of the TAUBnCMORm 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).
- The operation mode of slave channel 2 must be set to up down mode, see **Table 22.147, Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Triangle 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 22.151, Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Triangle 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
- 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 Triangle PWM Output Function with Dead Time.

Functional description

The counters are started by setting the channel trigger bits (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm, enabling count operation. The current values of TAUBnCDRm is written to TAUBnCNTm 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, 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.
- Slave channel 3:
  INTTAUBnIm of slave channel 2 triggers the counter of slave channel 3. The current value of TAUBnCDRm (slave 3) is written to TAUBnCNTm (slave 3) and the counter starts to count down from this value.
  When the counter reaches 0000_H, INTTAUBnIm is generated. The counter returns 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 22.142, 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 22.6, Simultaneous Rewrite**.

TAUBnTOL.TAUBnTOLm and TAUBnTDL.TAUBnTDLm bits should be set before the counter starts, and slave channels 2 and 3 should have opposite TAUBnTOL.TAUBnTOLm settings or opposite TAUBnTDL.TAUBnTDLm settings.

Table 22.142 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
1	Down	Set immediately
	Up	Reset after dead time has elapsed

22.14.6.2 Equations

$0000_H \leq$ TAUBnCDRm (master) $< FFFF_H$

Pulse 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

22.14.6.3 Block Diagram and General Timing Diagram

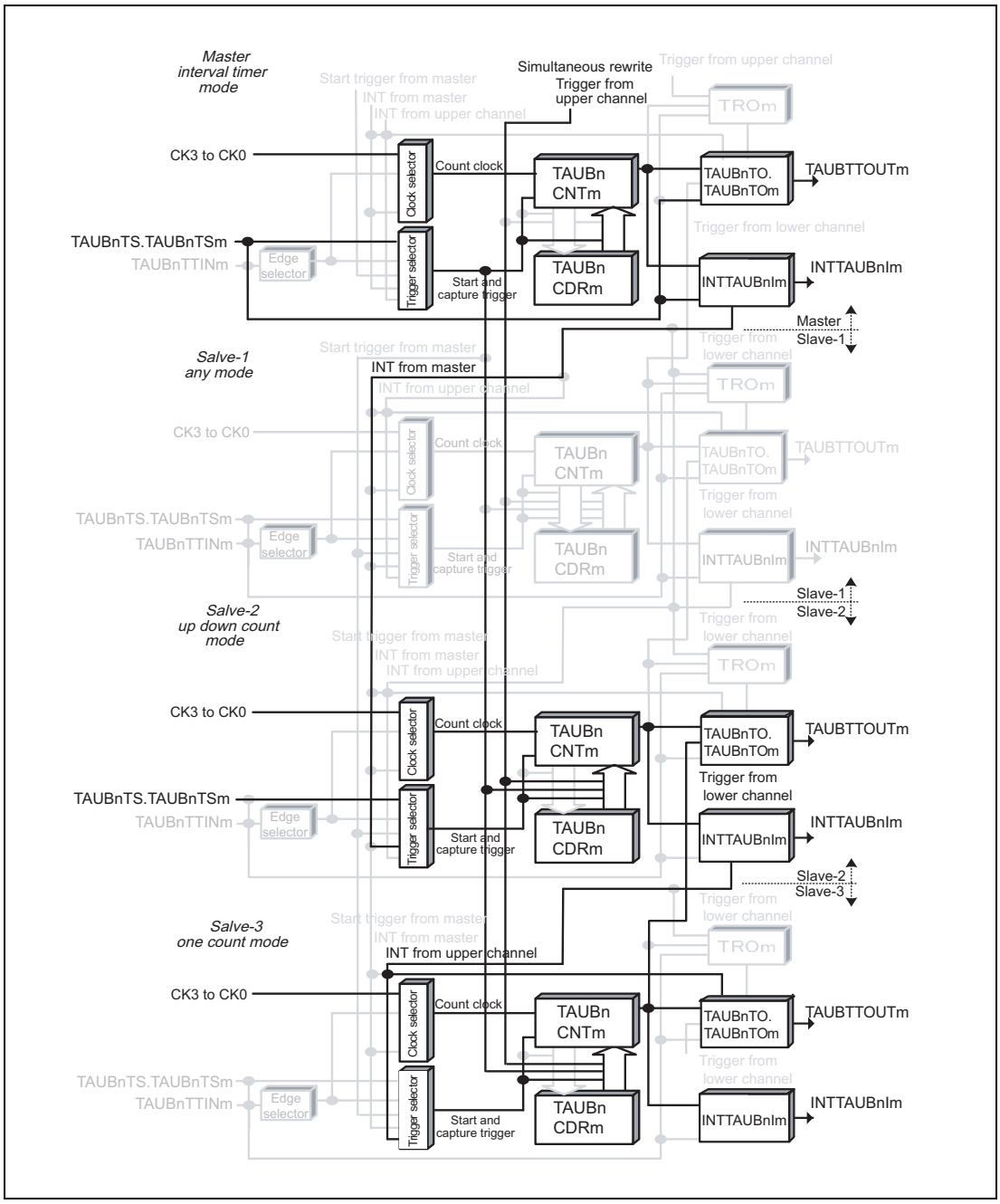


Figure 22.102 Block Diagram for Triangle PWM Output Function with Dead Time

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:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnTDL.TAUBnTDLm = 1
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)

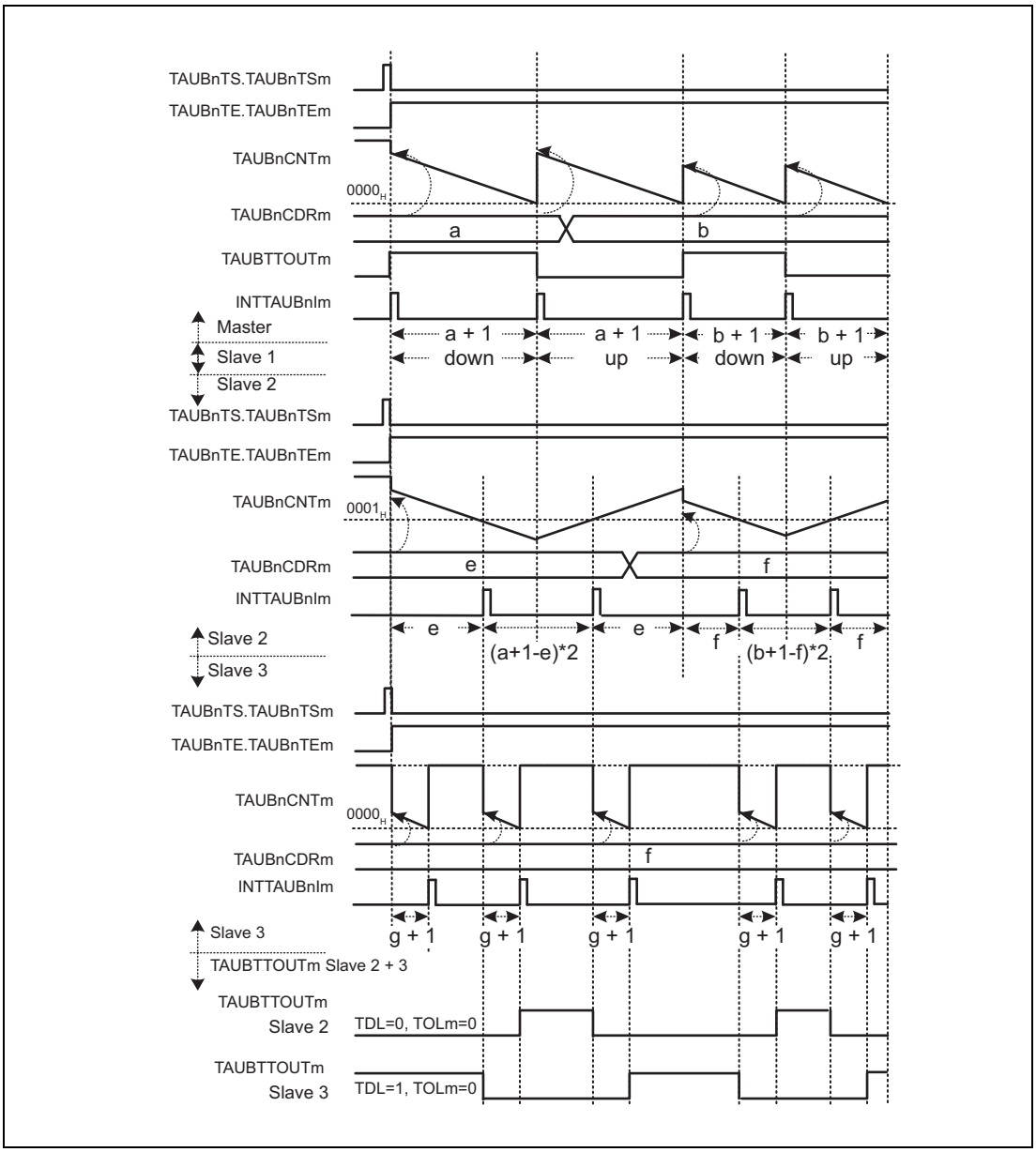


Figure 22.103 General Timing Diagram for Triangle PWM Output Function with Dead Time

22.14.6.4 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 22.143 Contents of the TAUBnCMORm Register for the Master Channel of the Triangle 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 to this bit, 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 to this bit, 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 22.144 Contents of the TAUBnCMURm 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 to these bits, 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 22.145 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 22.146 Simultaneous Rewrite Settings for the Master Channel of the Triangle 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.

22.14.6.5 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 22.147 Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Triangle 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 to this bit, 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 to this bit, 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 22.148 Contents of the TAUBnCMURm Register for the 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 to these bits, 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 22.149 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 TDLm 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 22.150 Simultaneous Rewrite Settings for Slave Channel 2 of the Triangle 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.

22.14.6.6 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 22.151 Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Triangle 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 to this bit, write the value after reset.
12	TAUBnCCS0	Write 00_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 to this bit, 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 22.152 Contents of the TAUBnCMURm Register for the 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 to these bits, 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 22.153 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 channel must be identical.

Table 22.154 Simultaneous Rewrite Settings for Slave Channel 3 of the Triangle 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.

22.14.6.7 Operating Procedure for Triangle PWM Output Function with Dead Time

Table 22.155 Operating Procedure for Delay Pulse Output

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 22.14.6.4, Register Settings for the Master Channel** Slave channel 2: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 22.14.6.5, Register Settings for Slave Channel 2** Slave channel 3: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 22.14.6.6, 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 on the master channel. TAUBnCMORm.TAUBnMD0 is set to 1.
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

22.14.6.8 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)

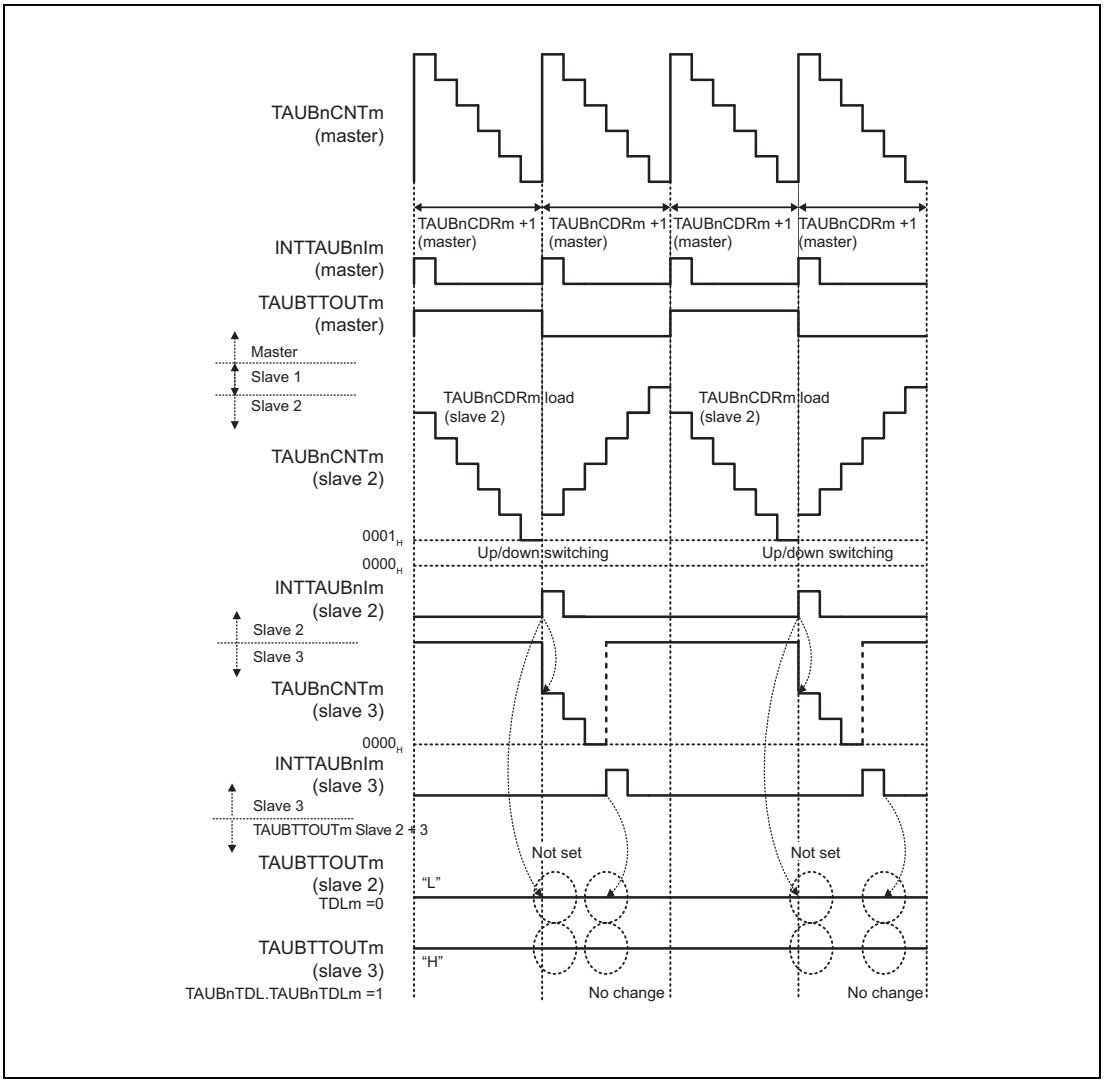


Figure 22.104 TAUBnCDRm (slave 2) ≥ TAUBnCDRm (master) + 1

- 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)

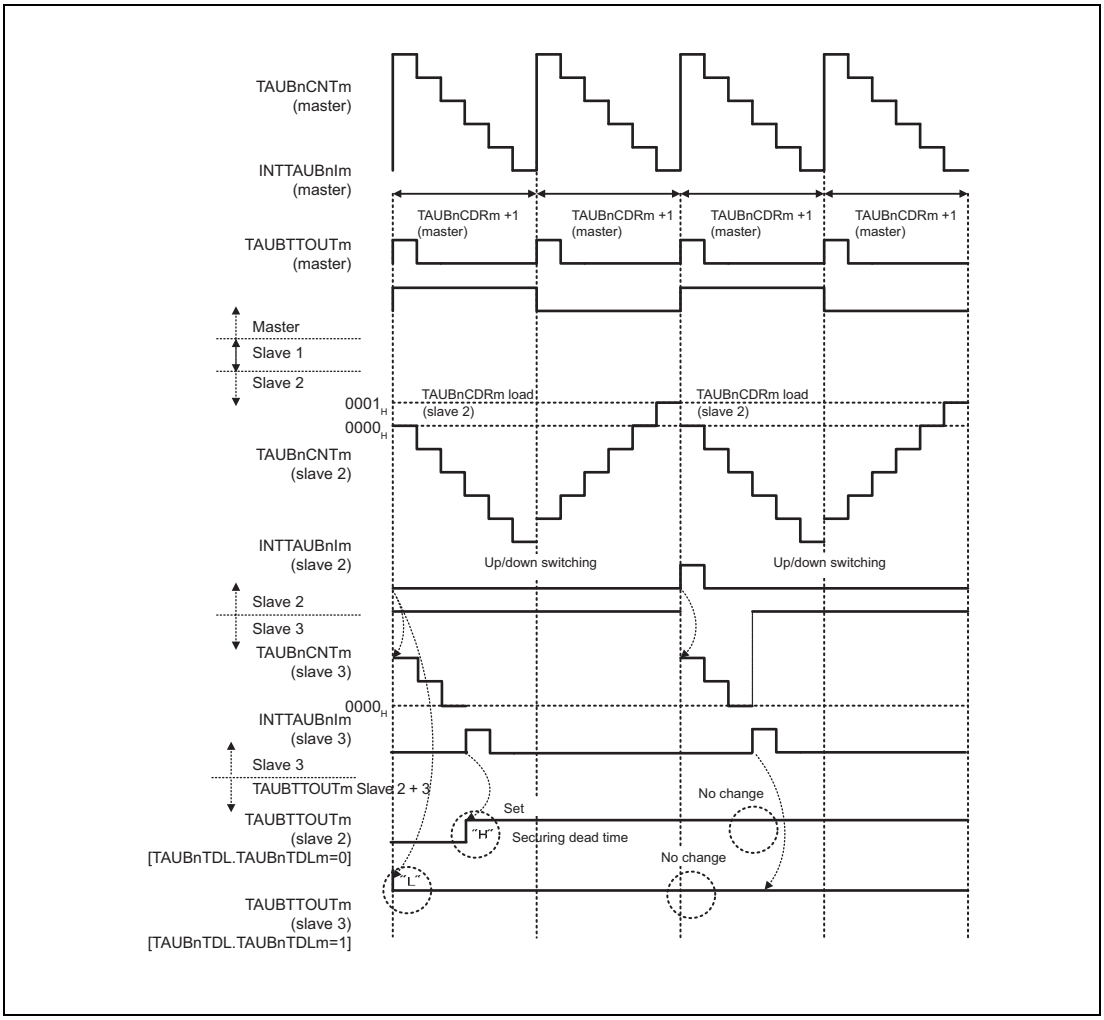


Figure 22.105 TAUBnCDRm (slave 2) = 0000_H

- 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)

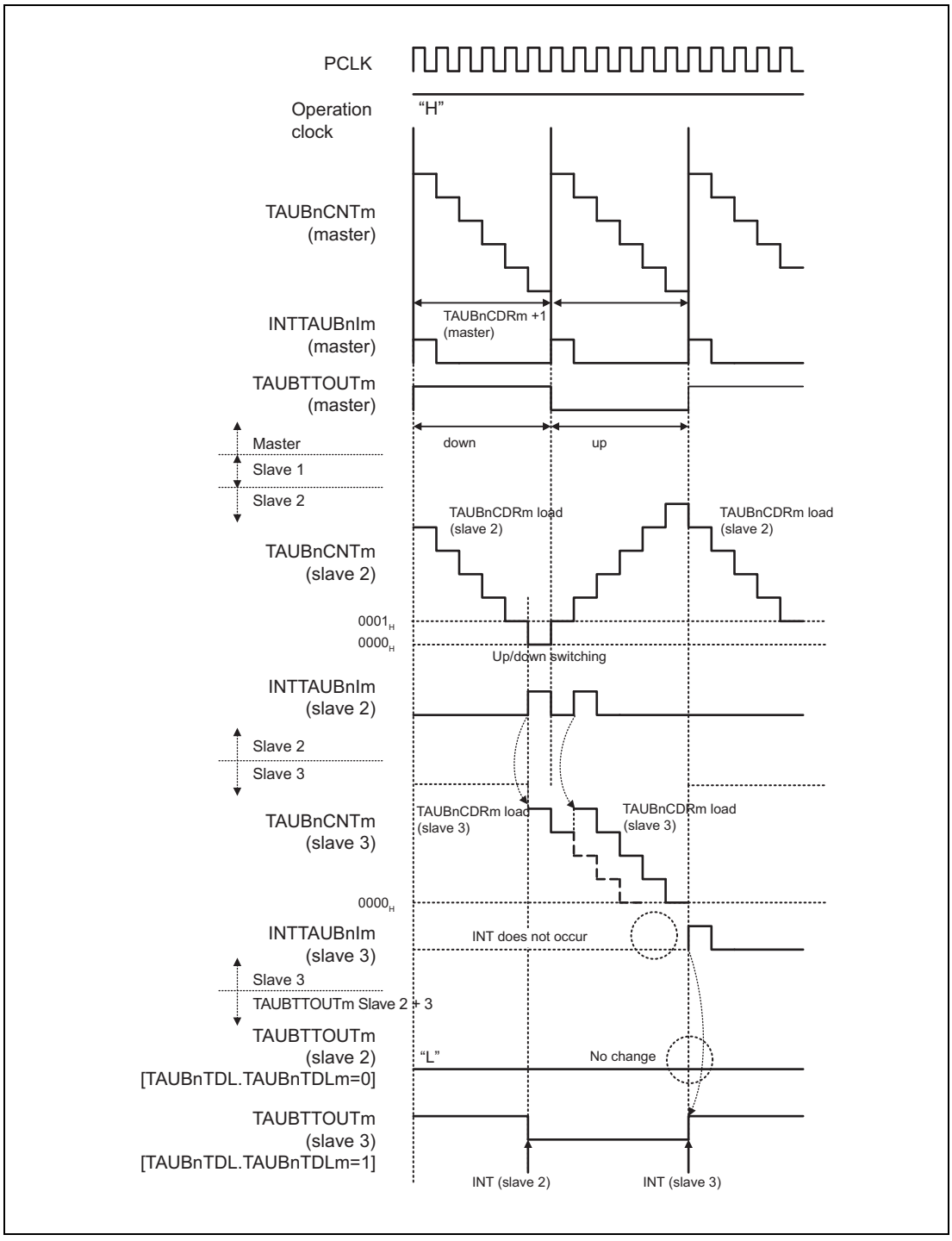


Figure 22.106 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0005_H TAUBnCDRm (slave 3) = 0004_H

- When the counter of slave channel 2 reaches 0000_H, 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 counter is counting up drying this period. This acts as a reset signal, 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)

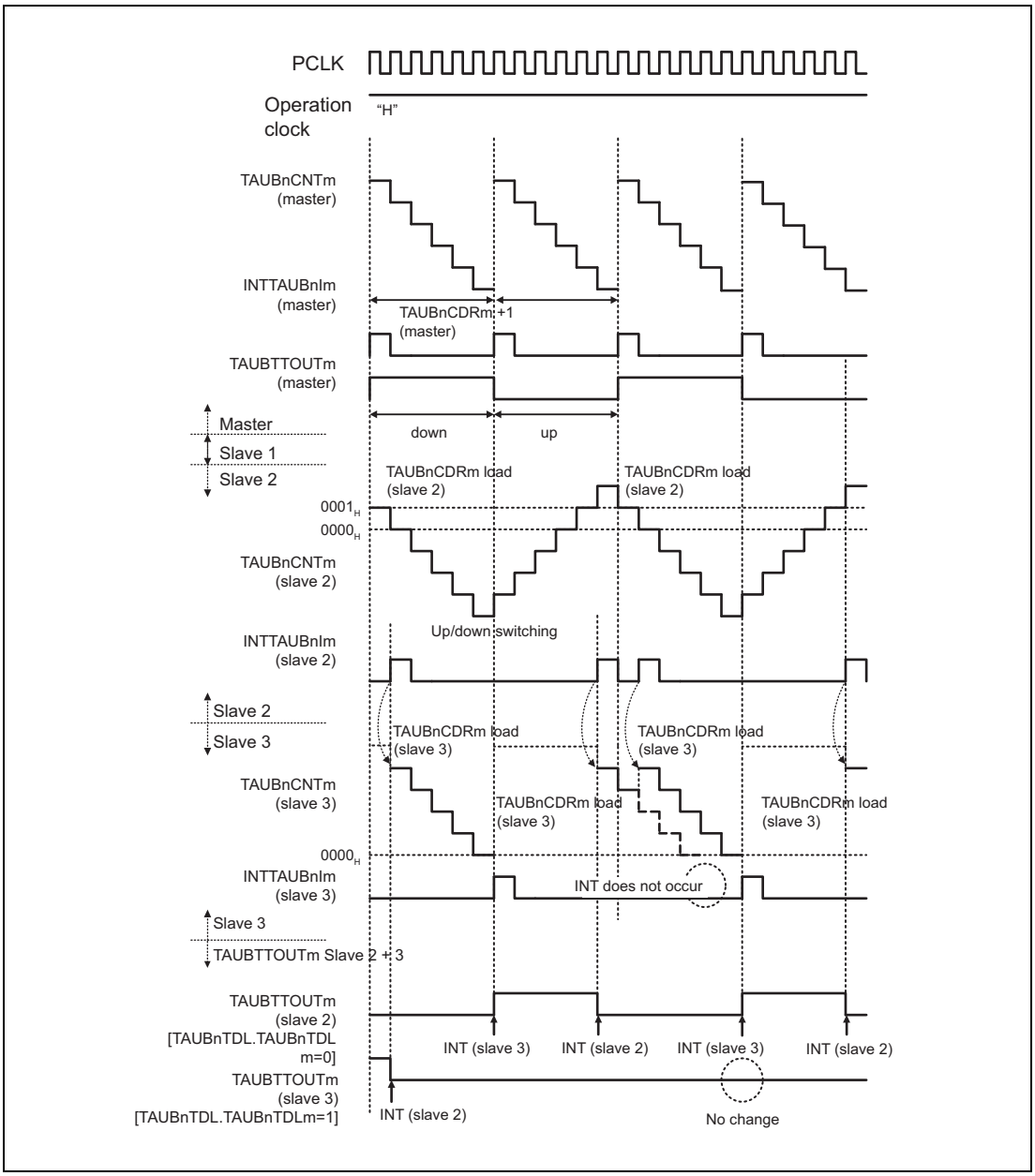


Figure 22.107 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0001_H TAUBnCDRm (slave 3) = 0004_H PWM Signal Width (negative phase) ≥ Carrier Cycle

- After the second interrupt, a slave for which TAUBnTDL.TAUBnTDLm = 1 is reset after the dead time elapsed. However if another interrupt occurs on slave 2, before the dead time has elapsed, the counter is counting up during this period. This acts as a set signal, 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 as normal 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)

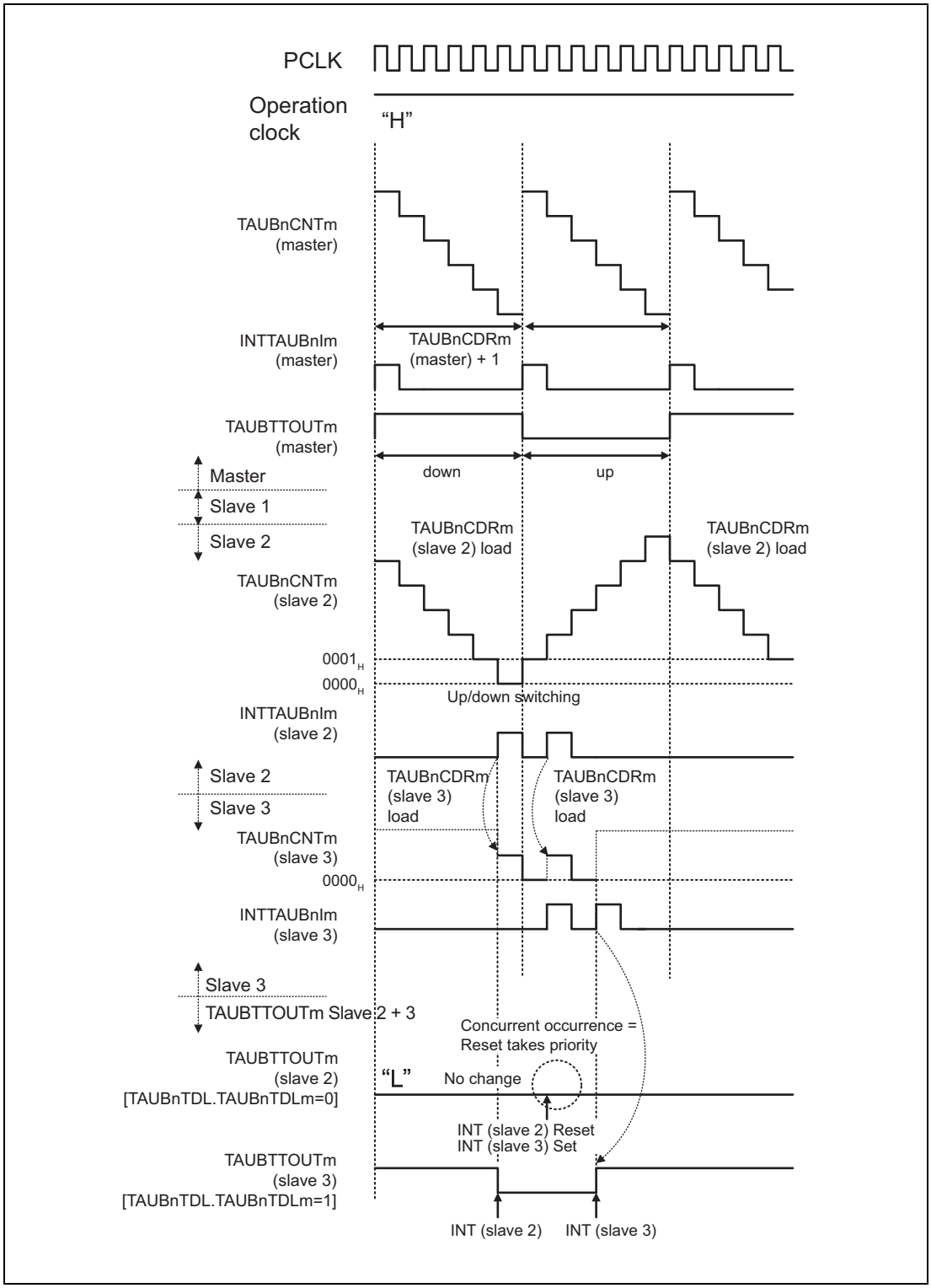


Figure 22.108 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0005_H, TAUBnCDRm (slave 3) = 0001_H PWM Signal Width (positive phase) = 0

- 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 channel 2 generates an INTTAUBnIm to reset TAUBTTOUTm simultaneously, this reset signal has priority (assuming TAUBnTOL.TAUBnTOLm = 0, otherwise the set signal has priority).
- Therefore, 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)

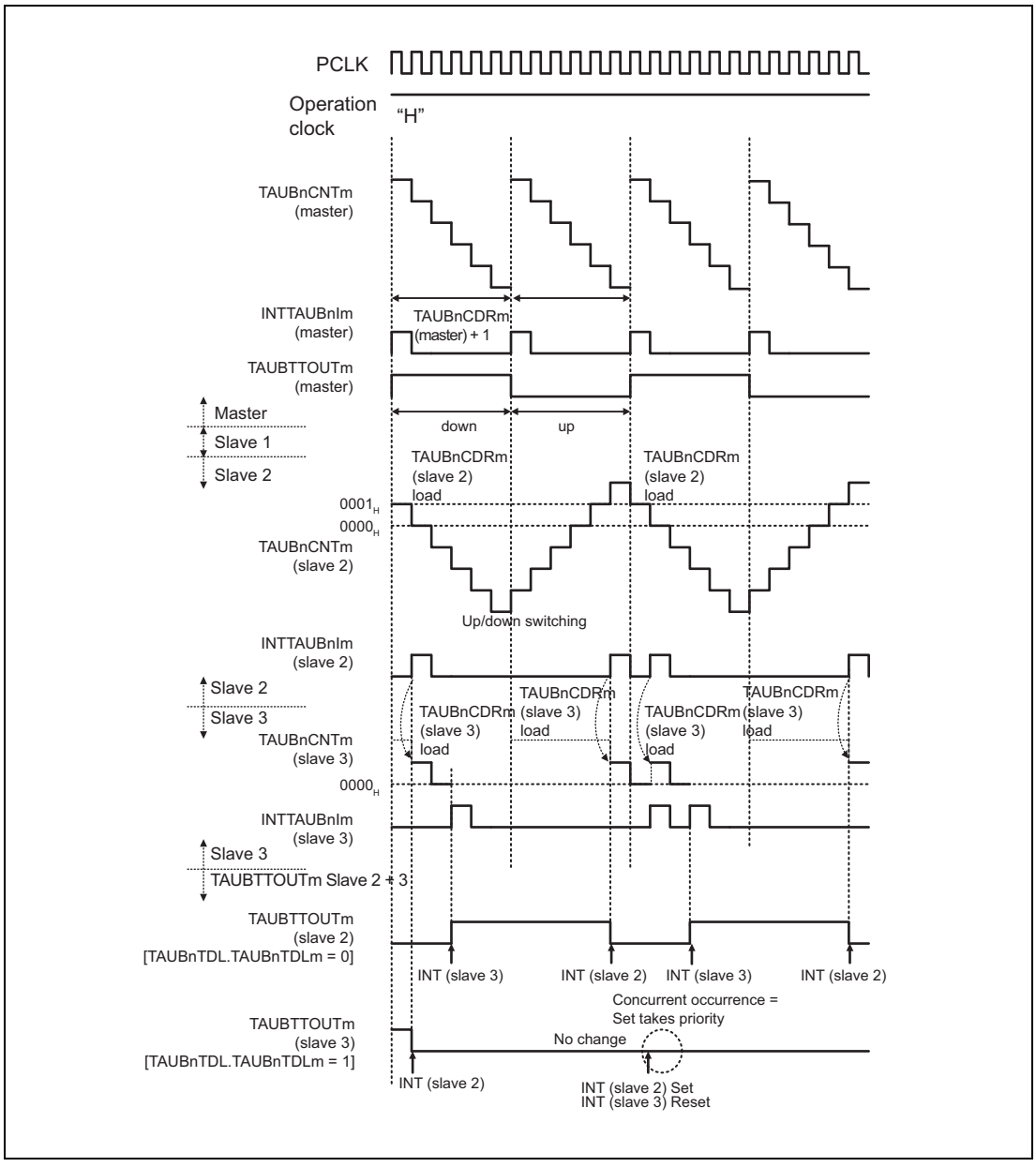


Figure 22.109 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0001_H, TAUBnCDRm (slave 3) = 0001_H PWM Signal Width (negative phase) = Carrier Cycle

- 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 channel 2 generates an INTTAUBnIm to reset TAUBTTOUTm simultaneously, the set signal has priority (assuming TAUBnTOL.TAUBnTOLm = 1, otherwise the reset signal has priority).
- Therefore, TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 1 remain in the value after reset.

(7) Rewriting to (slave channel 2) TAUBnCDRm = 0000_H (100% output)

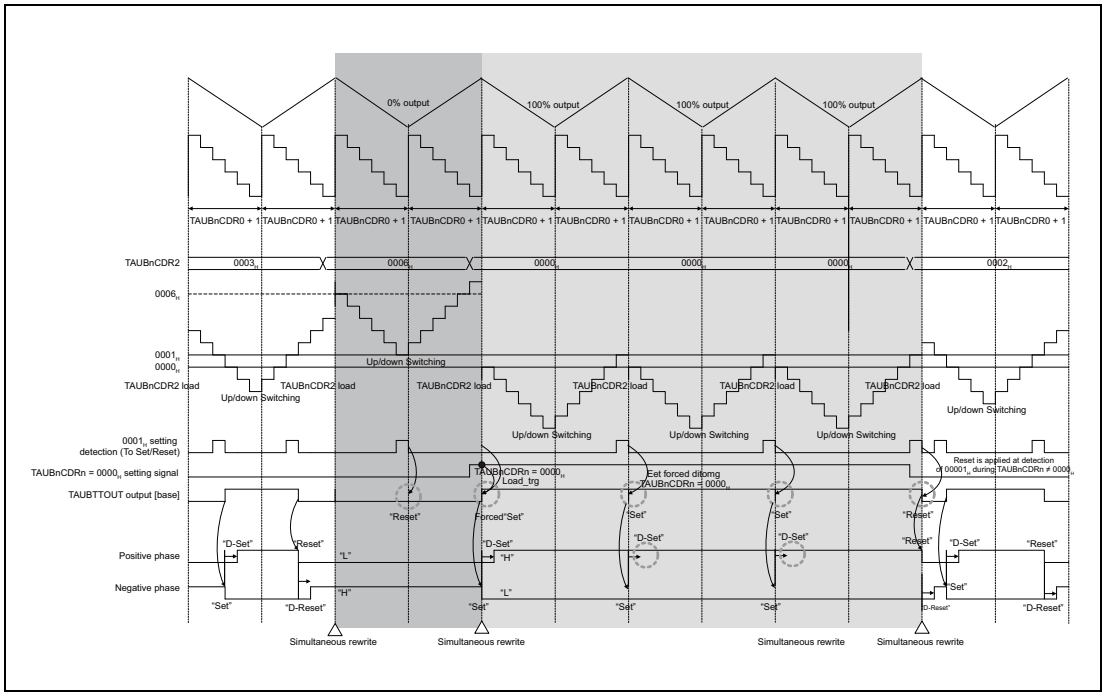


Figure 22.110 Rewriting to (slave channel 2) TAUBnCDRm = 0000_H (100% output)

When rewriting (slave channel 2) TAUBnCDRm ≠ 0000_H to (slave channel 2) TAUBnCDRm = 0000_H (100% output), set the negative phase side at the start of the carrier cycle, and set the positive phase side 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 side at the end of the carrier cycle, and reset the negative phase side after dead time is secured.

22.14.7 AD Conversion Trigger Output Function Type 2

22.14.7.1 Overview

Summary

This is achieved by setting the channel output mode of the slave to independent channel output mode controlled by software.

22.14.7.2 Block Diagram and General Timing Diagram

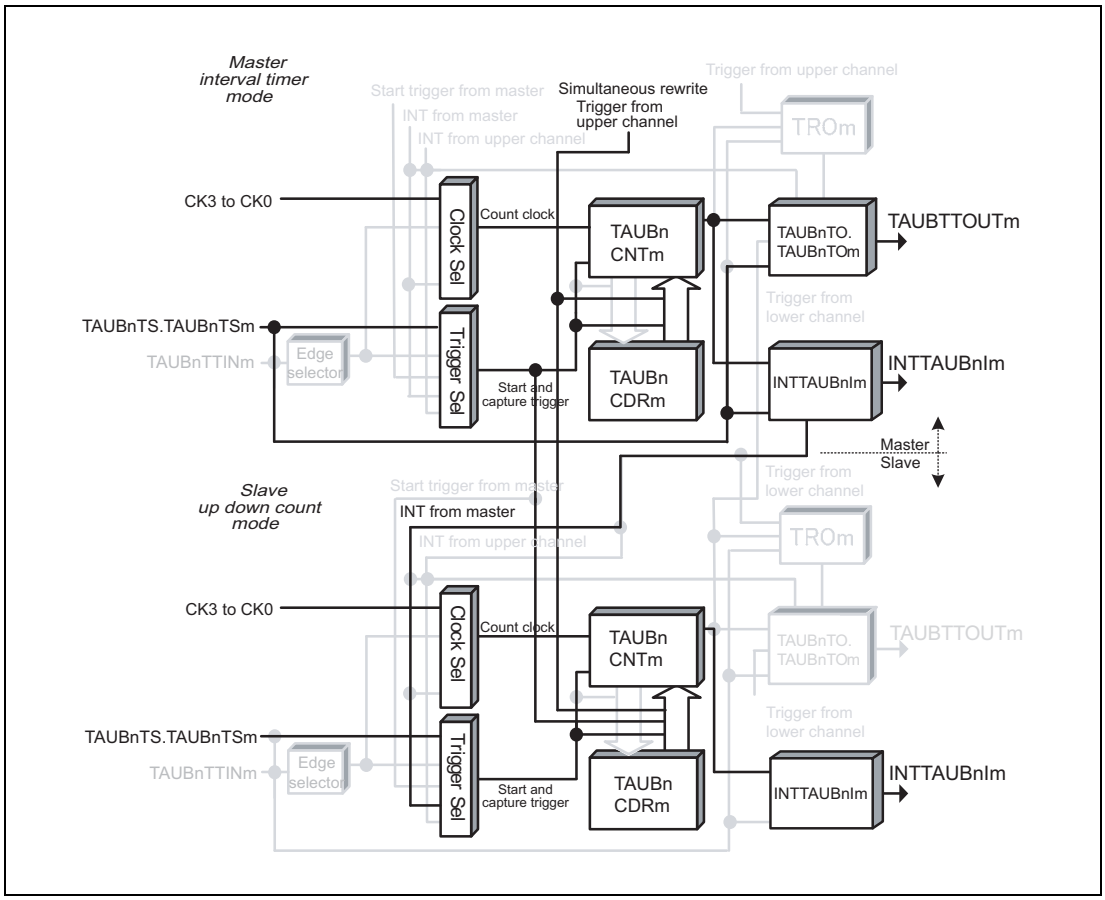


Figure 22.111 Block Diagram for AD Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

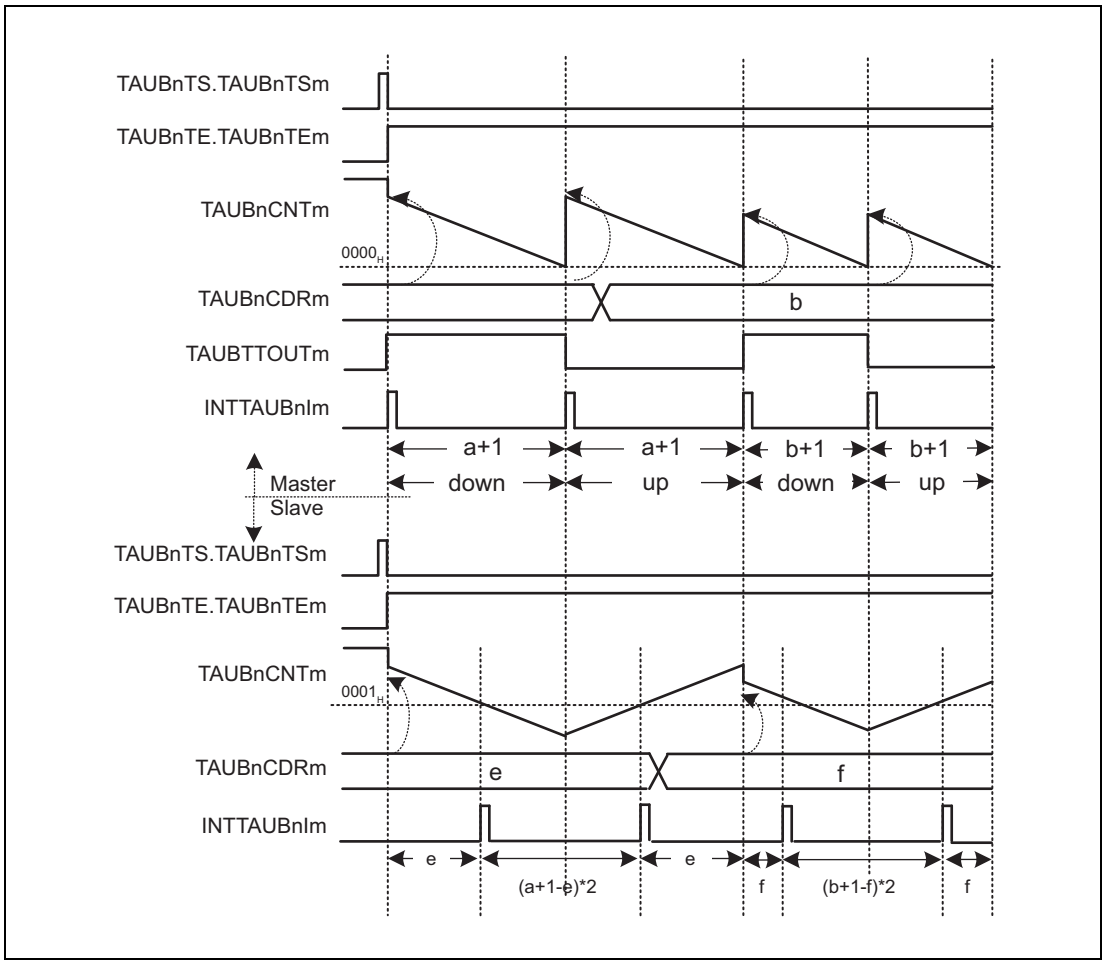


Figure 22.112 General Timing Diagram for AD Conversion Trigger Output Function Type 2

Section 23 Timer Array Unit D (TAUD)

This section contains a generic description of the timer array unit D (TAUD).

The first part in this section describes all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the TAUD.

23.1 Features of RH850/F1L TAUD

23.1.1 Number of Units and Channels

This microcontroller has the following number of TAUD units.

Table 23.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1					
Name	TAUDn (n = 0)					

TAUDn has the following number of channels of timers.

Table 23.2 Unit Configurations and Channels

Unit Name (Channel Name) TAUDn	Channels per Unit	RH850/F1L 48 pins (16 ch)	RH850/F1L 64 pins (16 ch)	RH850/F1L 80 pins (16 ch)	RH850/F1L 100 pins (16 ch)	RH850/F1L 144 pins (16 ch)	RH850/F1L 176 pins (16 ch)
TAUD0	16	√	√	√	√	√	√

Table 23.3 Index

Index	Meaning
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.

23.1.2 Register Base Address

TAUDn base addresses are listed in the following table.

TAUDn register addresses are given as offsets from the base addresses in general.

Table 23.4 Register Base Address

Base Address Name	Base Address
<TAUD0_base>	FFE2 0000_H

23.1.3 Clock Supply

The TAUDn clock supply is shown in the following table.

Table 23.5 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
TAUDn	PCLK	CKSCLK_IPERI1

23.1.4 Interrupt Requests

TAUDn interrupt requests are listed in the following table.

Table 23.6 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
TAUD0			
INTTAUD0I0	Channel 0 interrupt	0, 124	0 (Channels 0 to 7)
INTTAUD0I1	Channel 1 interrupt	39	15 (Channels 0 to 7)
INTTAUD0I2	Channel 2 interrupt	1, 150	0 (Channels 8 to 15)
INTTAUD0I3	Channel 3 interrupt	40	12 (Channels 8 to 15)
INTTAUD0I4	Channel 4 interrupt	2	1 (Channels 0 to 7)
INTTAUD0I5	Channel 5 interrupt	41	16 (Channels 0 to 7)
INTTAUD0I6	Channel 6 interrupt	3	1 (Channels 8 to 15)
INTTAUD0I7	Channel 7 interrupt	42	13 (Channels 8 to 15)
INTTAUD0I8	Channel 8 interrupt	4	2 (Channels 0 to 7)
INTTAUD0I9	Channel 9 interrupt	43	17 (Channels 0 to 7)
INTTAUD0I10	Channel 10 interrupt	5, 151	2 (Channels 8 to 15)
INTTAUD0I11	Channel 11 interrupt	44	14 (Channels 8 to 15)
INTTAUD0I12	Channel 12 interrupt	6, 152	3 (Channels 0 to 7)
INTTAUD0I13	Channel 13 interrupt	45	18 (Channels 0 to 7)
INTTAUD0I14	Channel 14 interrupt	7, 153	3 (Channels 8 to 15)
INTTAUD0I15	Channel 15 interrupt	46	15 (Channels 8 to 15)

23.1.5 Reset Sources

TAUDn reset sources are listed in the following table. TAUDn is initialized by these reset sources.

Table 23.7 Reset Sources

Unit Name	Reset Source
TAUDn	All reset sources (ISORES)

23.1.6 External Input/Output Signals

External input/output signals of TAUDn are listed below. .

Table 23.8 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
TAUD0		
TAUDTTIN0	Channel 0 input	TAUD0I0
TAUDTTIN1	Channel 1 input	TAUD0I1
TAUDTTIN2	Channel 2 input	TAUD0I2
TAUDTTIN3	Channel 3 input	TAUD0I3
TAUDTTIN4	Channel 4 input	TAUD0I4
TAUDTTIN5	Channel 5 input	TAUD0I5
TAUDTTIN6	Channel 6 input	TAUD0I6
TAUDTTIN7	Channel 7 input	TAUD0I7
TAUDTTIN8	Channel 8 input	TAUD0I8
TAUDTTIN9	Channel 9 input	TAUD0I9
TAUDTTIN10	Channel 10 input	TAUD0I10
TAUDTTIN11	Channel 11 input	TAUD0I11
TAUDTTIN12	Channel 12 input	TAUD0I12
TAUDTTIN13	Channel 13 input	TAUD0I13
TAUDTTIN14	Channel 14 input	TAUD0I14
TAUDTTIN15	Channel 15 input	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

23.1.7 Internal Input/Output Signals

The internal input/output signals of TAUDn are listed below.

Table 23.9 Internal Input/Output Signals

Unit Signal Name	Outline	Connected to
TAUDnTSSTm	Simultaneous channel start trigger input	PIC
TAUDnTUDCm (m = 0, 2, 8)	TAUD master up/down signal output	PIC

23.1.8 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.

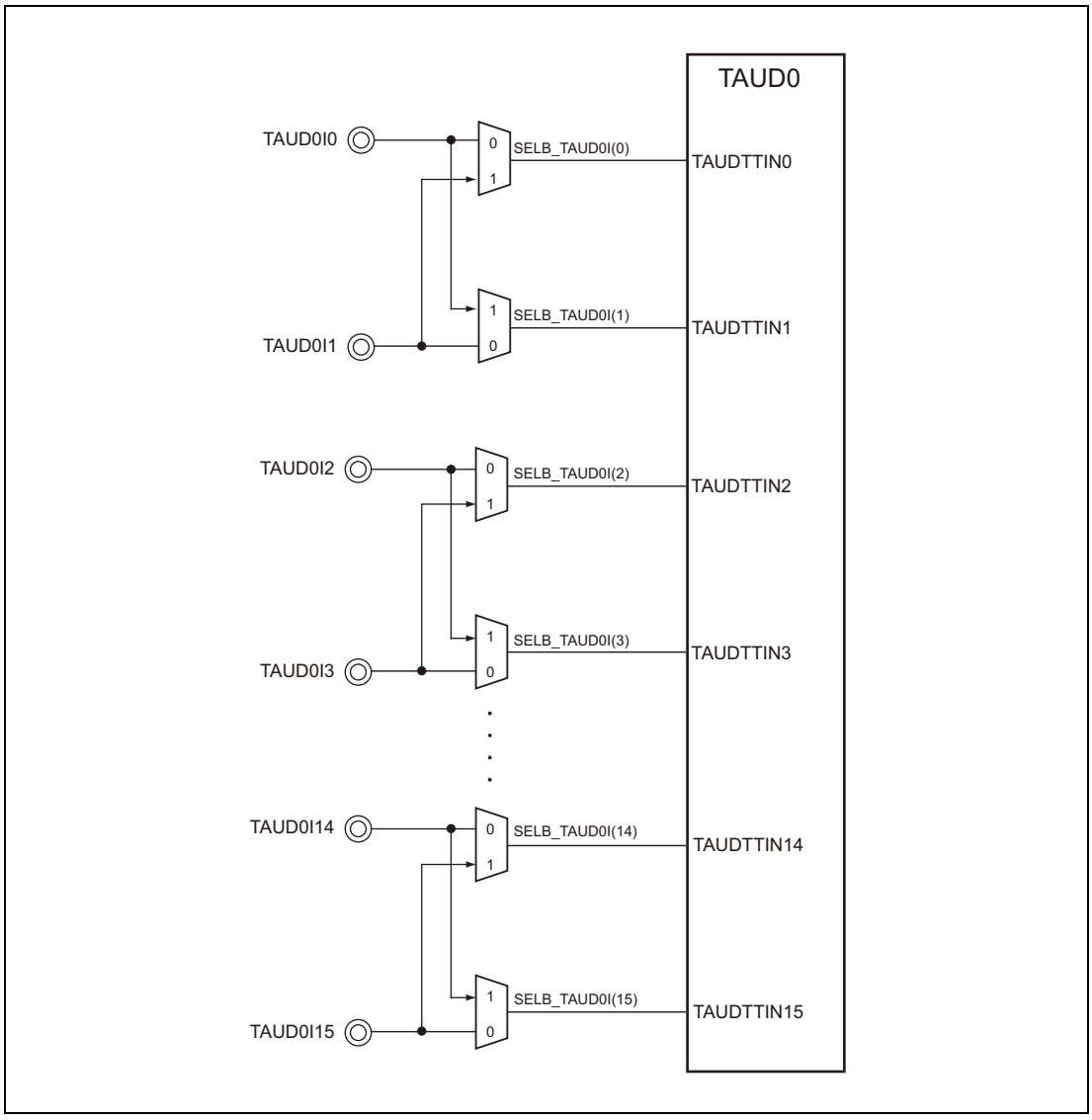


Figure 23.1 Selection of Signals Input to TAUD0

The following table shows the input signals to several TAUD0 inputs.

Table 23.10 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

23.1.8.1 SELB_TAUD0I — TAUDTTINm Input Signal Selection Register

This register selects the input signals to several TAUDTTINm inputs.

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

Address: FFBC 0200_H

Value after reset: 0000 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 23.11 SELB_TAUD0I Register Contents

Bit Position	Bit Name	Function
15 to 0	SELB_TAUD0Im	Selection of 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 the each channel during the timer counting.

23.2 Overview

23.2.1 Functional Overview

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), i.e., multiple channels depend on each other with one master and one or more slave channels.

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.

23.2.2 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.

23.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 23.12 Functional List of TAUD Operations

Operation Function	Example
Independent Channel Operation Functions	**Section 23.12**
Interval Timer Function	**Section 23.12.1**
TAUDTTINm Input Interval Timer Function	**Section 23.12.2**
Clock Divide Function	**Section 23.12.3**
External Event Count Function	**Section 23.12.4**
Delay Count Function	**Section 23.12.5**
One-Pulse Output Function	**Section 23.12.6**
TAUDTTINm Input Pulse Interval Measurement Function	**Section 23.12.7**
TAUDTTINm Input Signal Width Measurement Function	**Section 23.12.8**
TAUDTTINm Input Position Detection Function	**Section 23.12.9**
TAUDTTINm Input Period Count Detection Function	**Section 23.12.10**
TAUDTTINm Input Pulse Interval Judgment Function	**Section 23.12.11**
TAUDTTINm Input Signal Width Judgment Function	**Section 23.12.12**
Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)	**Section 23.12.13**
Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)	**Section 23.12.14**
Independent Channel Real-Time Functions	**Section 23.13**
Real-Time Output Function Type 1	**Section 23.13.1**
Real-Time Output Function Type 2	**Section 23.13.2**
Independent Channel Simultaneous Rewrite Functions	**Section 23.14**
Simultaneous Rewrite Trigger Generation Function Type 1	**Section 23.14.1**
Simultaneous Rewrite Trigger Generation Function Type 2	**Section 23.14.2**
Synchronous Channel Operation Functions	**Section 23.15**
PWM Output Function	**Section 23.15.1**
One-Shot Pulse Output Function	**Section 23.15.2**
Trigger Start PWM Output Function	**Section 23.15.3**
Delay Pulse Output Function	**Section 23.15.4**
Offset Trigger Output Function	**Section 23.15.5**
A/D Conversion Trigger Output Function Type 1	**Section 23.15.6**
Triangle PWM Output Function	**Section 23.15.7**
Triangle PWM Output Function with Dead Time	**Section 23.15.8**
A/D Conversion Trigger Output Function Type 2	**Section 23.15.9**
Interrupt Request Signals Culling Function	**Section 23.15.10**
PWM Output Function	**Section 23.15.1**
Synchronous Non-Complementary and Complementary Modulation Output Functions	**Section 23.16**
Non-Complementary Modulation Output Function Type 1	**Section 23.16.1**
Non-Complementary Modulation Output Function Type 2	**Section 23.16.2**
Complementary Modulation Output Function	**Section 23.16.3**

23.2.4 TAUD I/O and Interrupt Request Signals

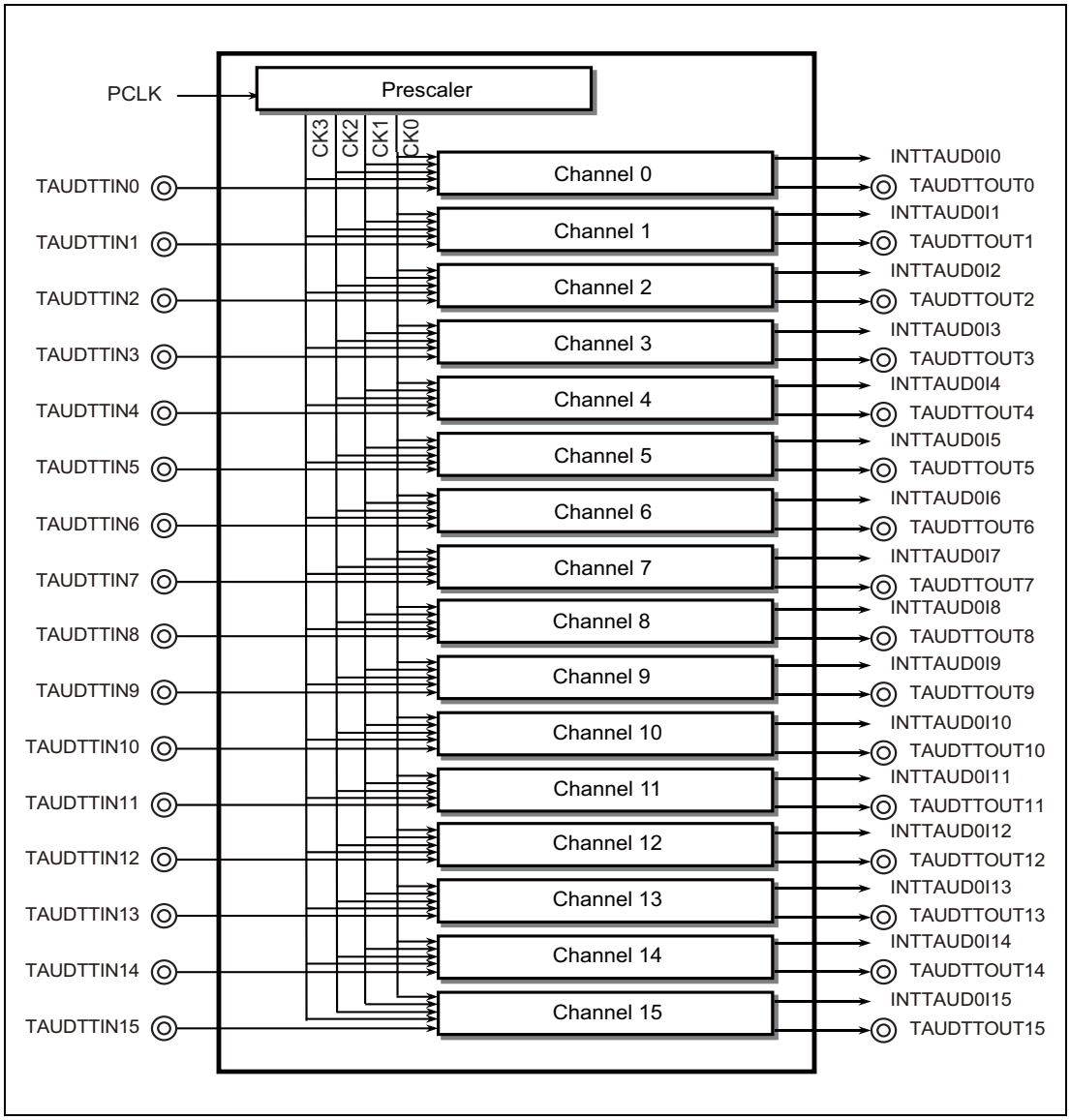


Figure 23.2 TAUD I/O and Interrupt Request Signals

23.2.5 Block Diagram

Figure 23.3 shows the main components of the TAUD.

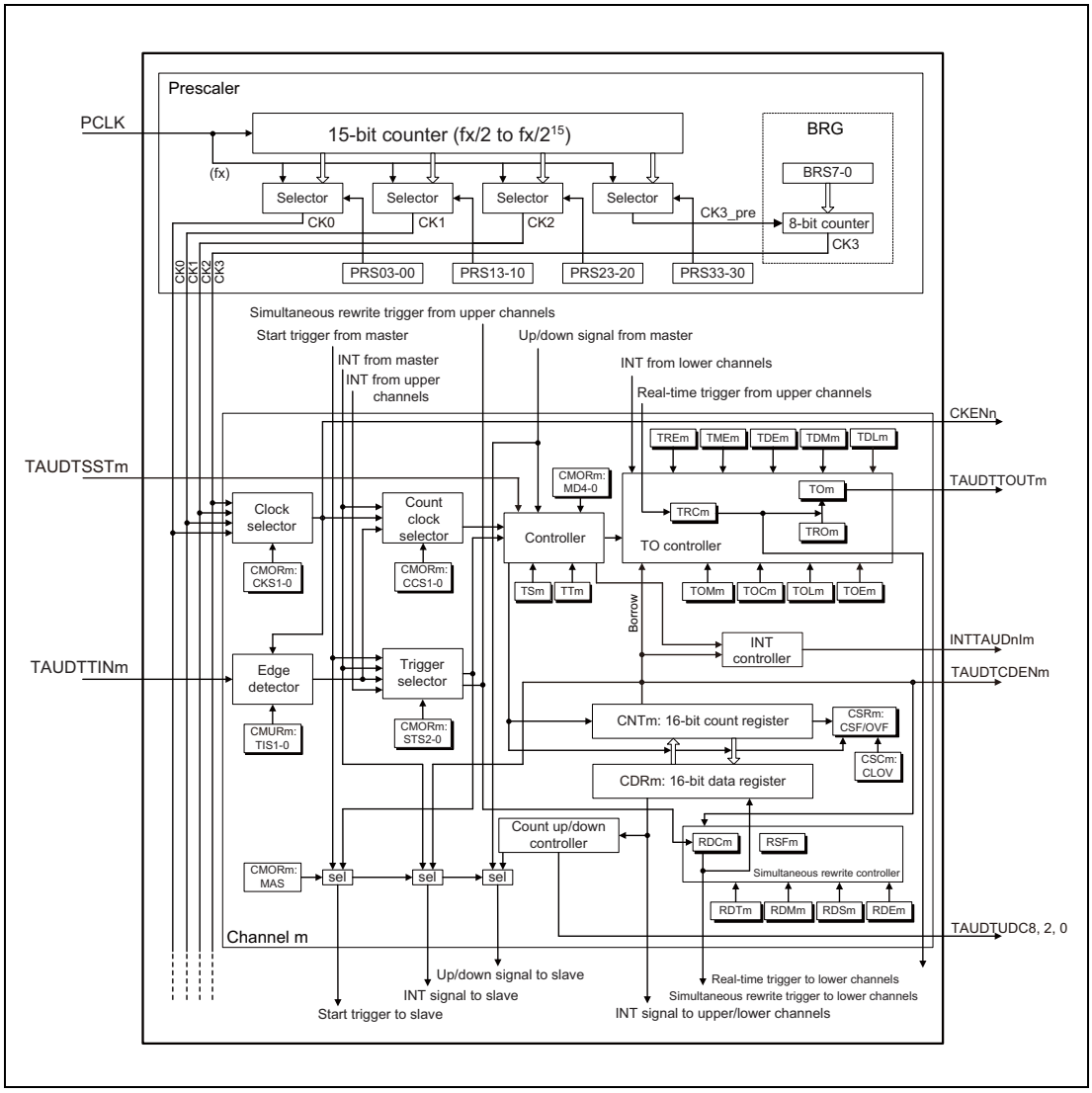


Figure 23.3 Block Diagram of the TAUD

The module name “TAUDn” has been omitted from the register names for the sake of clarity in the above figure.

23.2.6 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 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.

23.3 Registers

23.3.1 List of Registers

TAUD registers are listed in the following table.

For details about <TAUDn_base>, see **Section 23.1.2, Register Base Address**.

Table 23.13 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

23.3.2 Details of TAUDn Prescaler Registers

23.3.2.1 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: Readable/writable 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 23.14 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 23.14 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 23.14 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 23.1.3, Clock Supply**.

23.3.2.2 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: Readable/writable 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 23.15 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

23.3.3 Details of TAUDn Control Registers

23.3.3.1 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: Readable/writable 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	R/W

Table 23.16 TAUDnCDRm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnCDR [15:0]	Data register for capture/compare values

23.3.3.2 TAUDnCNTm — TAUDn Channel Counter Register

This is a channel m counter register.

Access: Readable in 16-bit units.

Address: <TAUDn_base> + 80_H + m × 4_H

Value after reset: $FFFF_H$ The initial value after reset depends on an operating mode. See **Table 23.18, TAUDnCNTm Read Values after Re-Enabling Counter**.

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 23.17 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 23.18 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 23.18 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 reset release

23.3.3.3 TAUDnCMORm — TAUDn Channel Mode OS Register

This register controls channel m operation.

Access: Readable/writable 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 23.19 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 allow 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 23.19 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>TAUDn STS2</th><th>TAUDn STS1</th><th>TAUDn STS0</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>TAUDn COS1</th><th>TAUDn COS0</th><th>TAUDnCDRm</th><th>TAUDnCSRm.TAUDnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of valid edge of TAUDTTINm input.</td><td>Updated (cleared or set) by detecting valid edge of TAUDTTINm input:
• If a counter overflow has occurred since the last detection of valid edge, set TAUDnCSRm.TAUDnOVF.
• If no counter overflow has occurred since the last detection of valid edge, clear TAUDnCSR.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 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 to this bit, write the value after reset.</td></tr>
</table>

Table 23.19 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 (Enables/disables start trigger detection during counting).
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 not 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

23.3.3.4 TAUDnCMURm — TAUDn Channel Mode User Register

This register specifies a type of valid edge detection used for TAUDTTINm input.

Access: Readable/writable 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 23.20 TAUDnCMURm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to this bit, 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].

23.3.3.5 TAUDnCSRm — TAUDn channel status register

This register indicates the count direction and overflow status of channel m counter.

Access: Only readable 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 23.21 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].

23.3.3.6 TAUDnCSCm — TAUDn Channel Status Clear Register

This is a trigger register for clearing the overflow flag TAUDnCSRm.TAUDnOVF of channel m.

Access: Only writable in 8-bit units.

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 23.22 TAUDnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	TAUDnCLOV	0: No function 1: Clears overflow flag TAUDnCSRm.TAUDnOVF.

23.3.3.7 TAUDnTS — TAUDn Channel Start Trigger Register

This register enables the counter operation of each channel.

Access: Only writable in 16-bit units.

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 23.23 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.

23.3.3.8 TAUDnTE — TAUDn Channel Enable Status Register

This register enables/disables a counter operation.

Access: Only readable 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 23.24 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.

23.3.3.9 TAUDnTT — TAUDn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access: Only writable in 16-bit units.

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 23.25 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.

23.3.4 Details of TAUDn Simultaneous Rewrite Registers

23.3.4.1 TAUDnRDE — TAUDn Channel Reload Data Enable Register

This register enables/disables simultaneous rewrite of TAUDnCDRm/TAUDnTOLm data register.

Access: Readable/writable 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 23.26 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

23.3.4.2 TAUDnRDS — TAUDn Channel Reload Data Control Channel Select Register

This register selects a channel that controls simultaneous rewrite.

Access: Readable/writable 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 23.27 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

23.3.4.3 TAUDnRDM — TAUDn Channel Reload Data Mode Register

This register selects the timing for generating a simultaneous rewrite control signal.

Access: Readable/writable 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 23.28 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.

23.3.4.4 TAUDnRDC — TAUDn Channel Reload Data Control Register

This register specifies a channel which generates an INTTAUDnlm signal to trigger simultaneous rewrite.

Access: Readable/writable 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 23.29 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 TAUDnRDS.TAUDnRDSm = 1.

23.3.4.5 TAUDnRDT — TAUDn Channel Reload Data Trigger Register

This register triggers a simultaneous rewrite enabling state.

Access: Only writable in 16-bit units.

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 23.30 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

23.3.4.6 TAUDnRSF — TAUDn Channel Reload Status Register

This flag register indicates simultaneous rewrite status.

Access: Only readable 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 23.31 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).

23.3.5 Details of TAUDn Output Registers

23.3.5.1 TAUDnTOE — TAUDn Channel Output Enable Register

This register enables/disables the independent channel output mode controlled by software.

Access: Readable/writable 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 23.32 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. Only TAUDnTOm bits for which timer output of a channel is disabled (TAUDnTOEm = 0) can be written.

23.3.5.2 TAUDnTO — TAUDn Channel Output Register

This register specifies and reads a TAUDTTOUTm level.

Access: Readable/writable 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 23.33 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.

23.3.5.3 TAUDnTOM — TAUDn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: Readable/writable 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 23.34 TAUDnTOM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUDnTOMm	Specifies an output mode. 0: Independent channel operation 1: Synchronous channel operation

23.3.5.4 TAUDnTOC — TAUDn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUDnTOMm.

Access: Readable/writable 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 23.35 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.

23.3.5.5 TAUDnTOL — TAUDn Channel Output Level Register

This register specifies the output logic of channel output bit (TAUDnTO.TAUDnTOm).

Access: Readable/writable 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
	TAUDnTOL15	TAUDnTOL14	TAUDnTOL13	TAUDnTOL12	TAUDnTOL11	TAUDnTOL10	TAUDnTOL09	TAUDnTOL08	TAUDnTOL07	TAUDnTOL06	TAUDnTOL05	TAUDnTOL04	TAUDnTOL03	TAUDnTOL02	TAUDnTOL01	TAUDnTOL00
Value after 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 23.36 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)

23.3.6 Details of TAUDn Dead Time Output Registers

23.3.6.1 TAUDnTDE — TAUDn Channel Dead Time Output Enable Register

This register enables/disables the dead time operation of every channel.

Access: Readable/writable 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 23.37 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

23.3.6.2 TAUDnTDM — TAUDn Channel Dead Time Output Mode Register

This register specifies the timing to add dead time during dead time output.

Access: Readable/writable 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 23.38 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

23.3.6.3 TAUDnTDL — TAUDn Channel Dead Time Output Level Register

This register selects a phase in which dead time is added.

Access: Readable/writable 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 23.39 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

23.3.7 Details of TAUDn Real-time/Modulation Output Registers

23.3.7.1 TAUDnTRE — TAUDn Channel Real-time Output Enable Register

This register enables/disables real-time output.

Access: Readable/writable 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 23.40 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 TAUTnTOE.TAUDnTOEm = 1.These bit settings are applied only when TAUTnTOE.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.

23.3.7.2 TAUDnTRC — TAUDn Channel Real-time Output Control Register

This register controls the real-time output trigger of each channel.

Access: Readable/writable 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 23.41 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.

23.3.7.3 TAUDnTRO — TAUDn Channel Real-time Output Register

This register sets a value which is output to TAUDTTOUTm.

Access: Readable/writable 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 23.42 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.

23.3.7.4 TAUDnTME — TAUDn Channel Modulation Output Enable Register

This register enables/disables modulation output for timer output and real-time output.

Access: Readable/writable 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 23.43 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.

23.3.8 TAUDn Emulation Register

23.3.8.1 TAUDnEMU — TAUDn Emulation Register

This register controls SVSTOP operations.

Access: Readable/writable in 8-bit units.
Perform write operations when the counter is being 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 23.44 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 to these bits, write the value after reset.

23.4 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 23.12, Independent Channel Operation Functions** and **Section 23.15, Synchronous Channel Operation Functions**.
2. The function can be changed while the counter is stopped (TAUDnTE.TAUDnTEm = 0).

23.5 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 23.5.1, Rules of Synchronous Channel Operation**.

Two special features for synchronous channel operation are detailed in the following:

- **Section 23.5.2, Simultaneous Start and Stop of Synchronous Channel Counters**
- **Section 23.6, Simultaneous Rewrite**

23.5.1 Rules of Synchronous Channel Operation

Number of masters and slaves

- 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 using 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 23.4**.

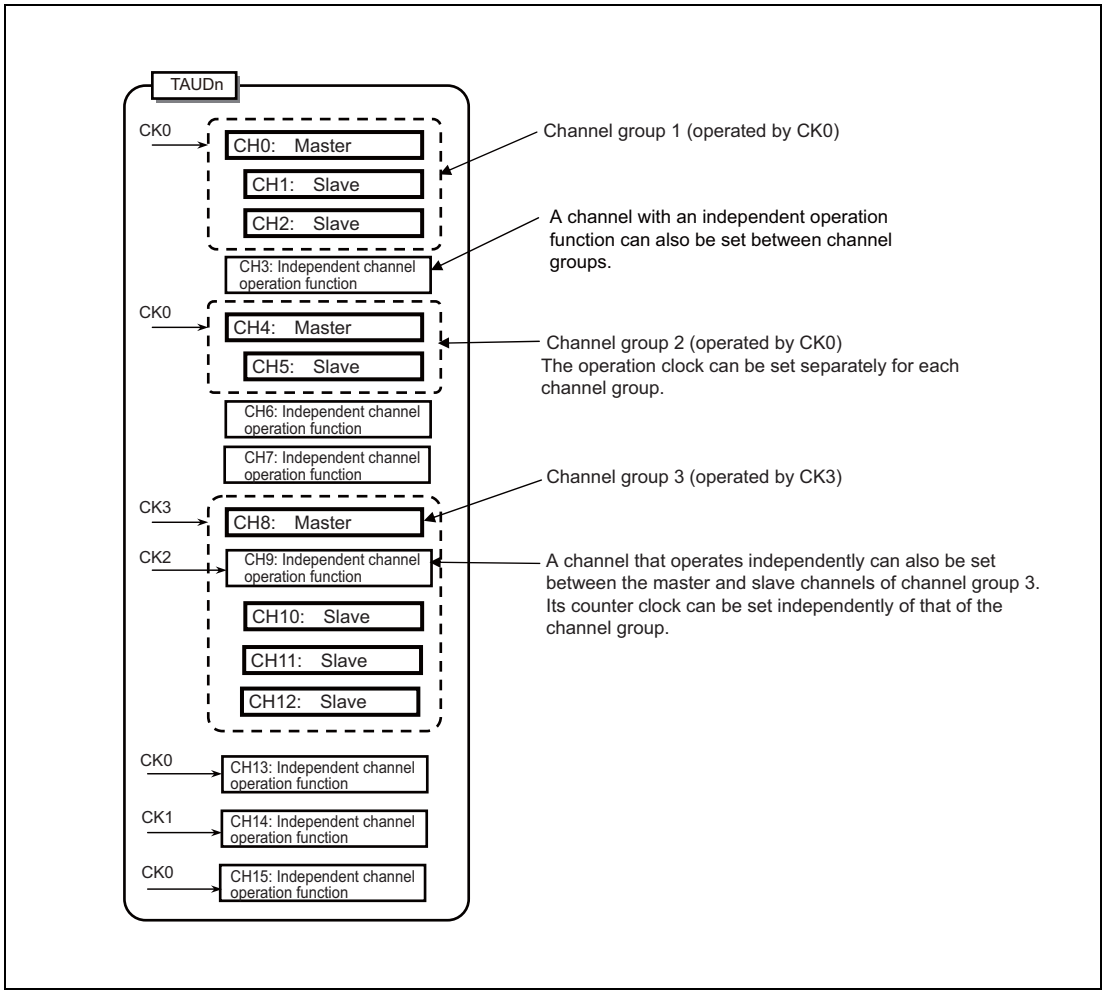


Figure 23.4 Grouping of Channels and Assignment of Count Clocks

Control trigger signal for master/salve 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 for their own to lower channels.
- Master channels cannot use control trigger signals from upper master channels.

23.5.2 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.

23.5.2.1 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.

23.5.2.2 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 27.8, Simultaneous Start Trigger Function**.

23.6 Simultaneous Rewrite

23.6.1 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 23.45**, along with how to specify them and when they cause simultaneous rewrite to be triggered.

Table 23.45 Simultaneous Rewrite Methods and when They are Triggered

Method	Simultaneous Rewrite Triggered when	TAUDnRDE. TAUDnRDEm	TAUDnRDS. TAUDnRDSm	TAUDnRDM. TAUDnRDMm
—	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
C2	INTTAUDnIm is generated on an upper channel specified by TAUDnRDC.TAUDnRDCm that in turn is triggered by an external signal	1	1	0

Table 23.46 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 23.12, Independent Channel Operation Functions** and **Section 23.15, Synchronous Channel Operation Functions**.

Table 23.46 Channel Operation Functions and Methods They Use

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

23.6.2 How to Control Simultaneous Rewrite

Figure 23.5 shows the general procedure for simultaneous rewrite. The three main blocks (initial settings, start and counter count operation, and simultaneous rewrite) are explained afterwards.

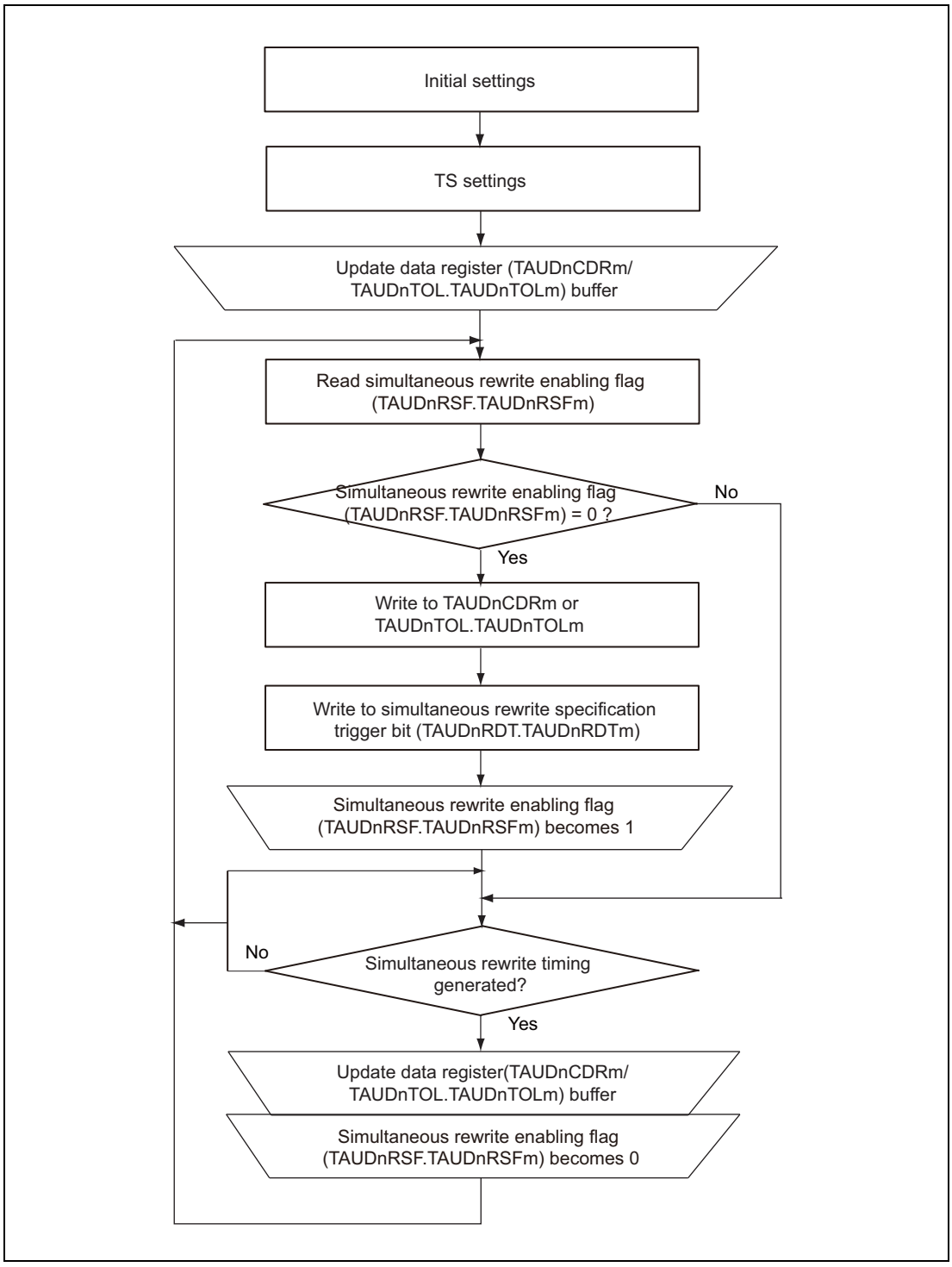


Figure 23.5 General Procedure for Simultaneous Rewrite

23.6.2.1 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 23.45, Simultaneous Rewrite Methods and when They are Triggered**.
- To select which upper channel is monitored for simultaneous rewrite triggers, use TAUDnRDC.TAUDnRDCm (prerequisite: TAUDnRDS.TAUDnRDSm is set in upper channel).

23.6.2.2 Start Counter and Count Operation

- To start all the TAUDnCNTm counters of the channel group, set the corresponding TAUDnTS.TAUDnTSm bits to1. 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.

23.6.2.3 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.
- The TAUDnRSF.TAUDnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

23.6.3 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 for 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.

23.6.4 Types of Simultaneous Rewrite

In the following section, the four simultaneous rewrite methods are explained using timing diagrams.

23.6.4.1 Simultaneous Rewrite when the Master Channel (Re)starts Counting (Method A)

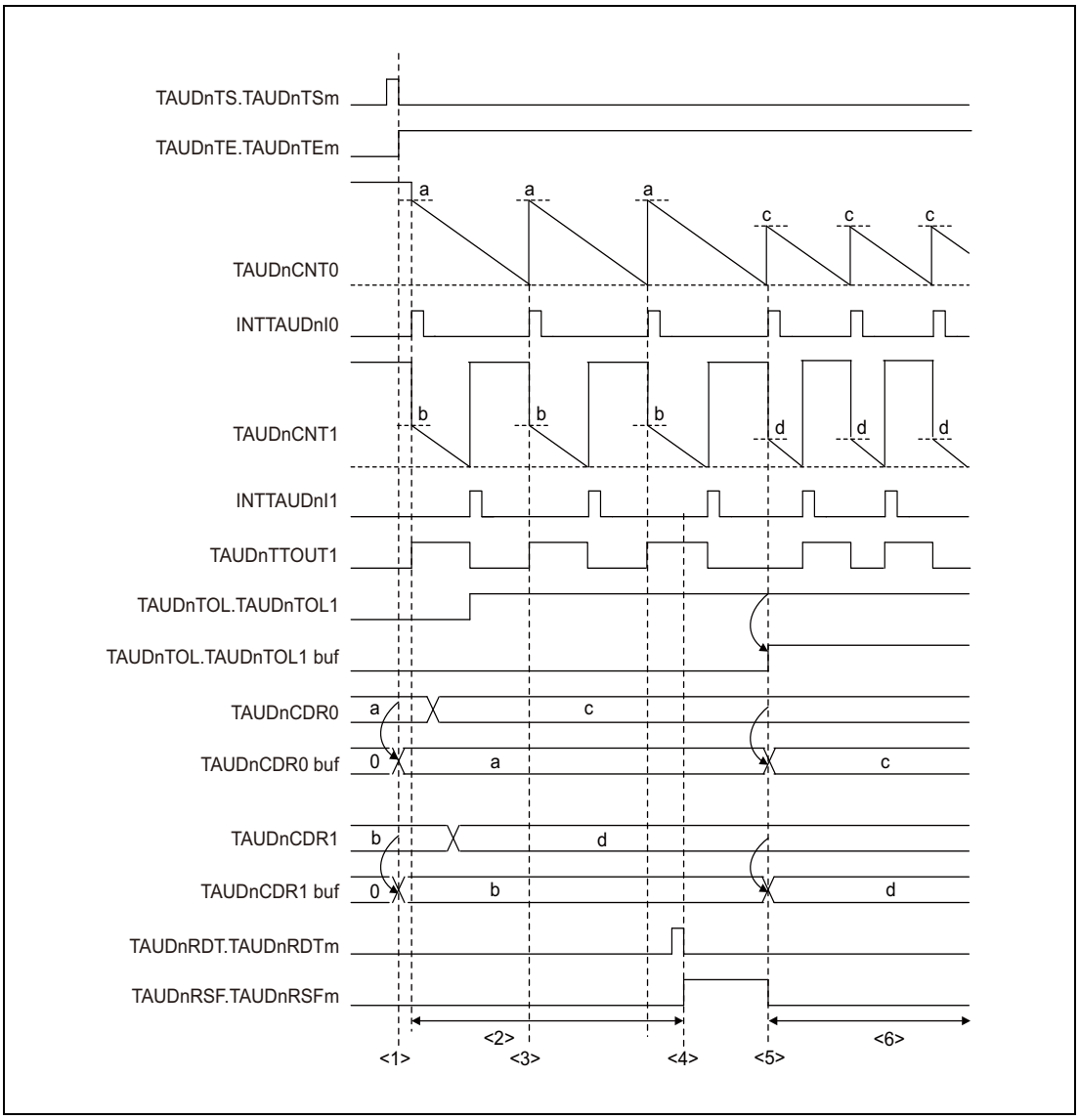


Figure 23.6 Simultaneous Rewrite when the Master Channel (Re)starts Counting

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.

23.6.4.2 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel (Method B)

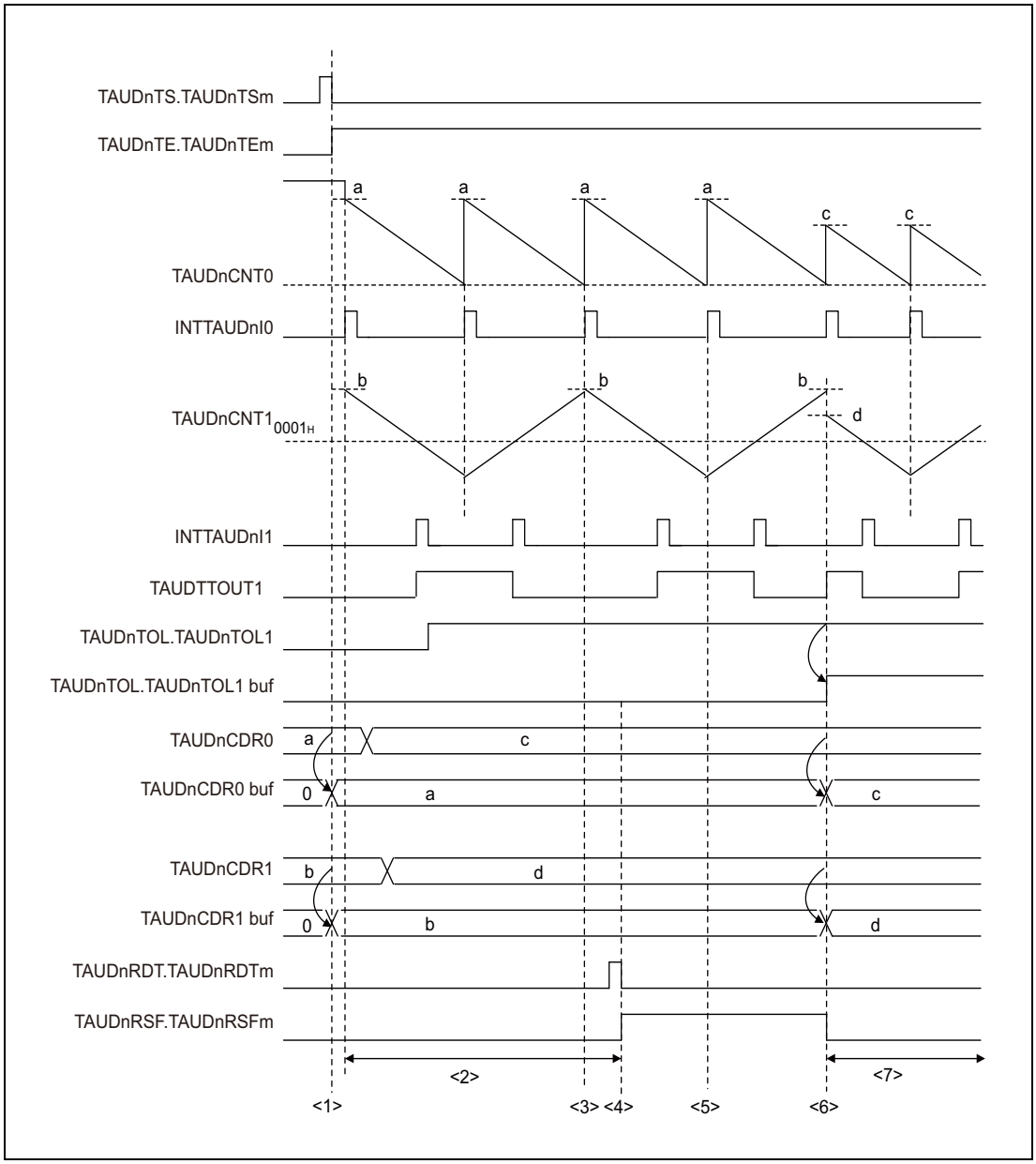


Figure 23.7 Simultaneous Rewrite at the Peak of a Triangular Wave of Slave Channel

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.

23.6.4.3 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm (Method C1)

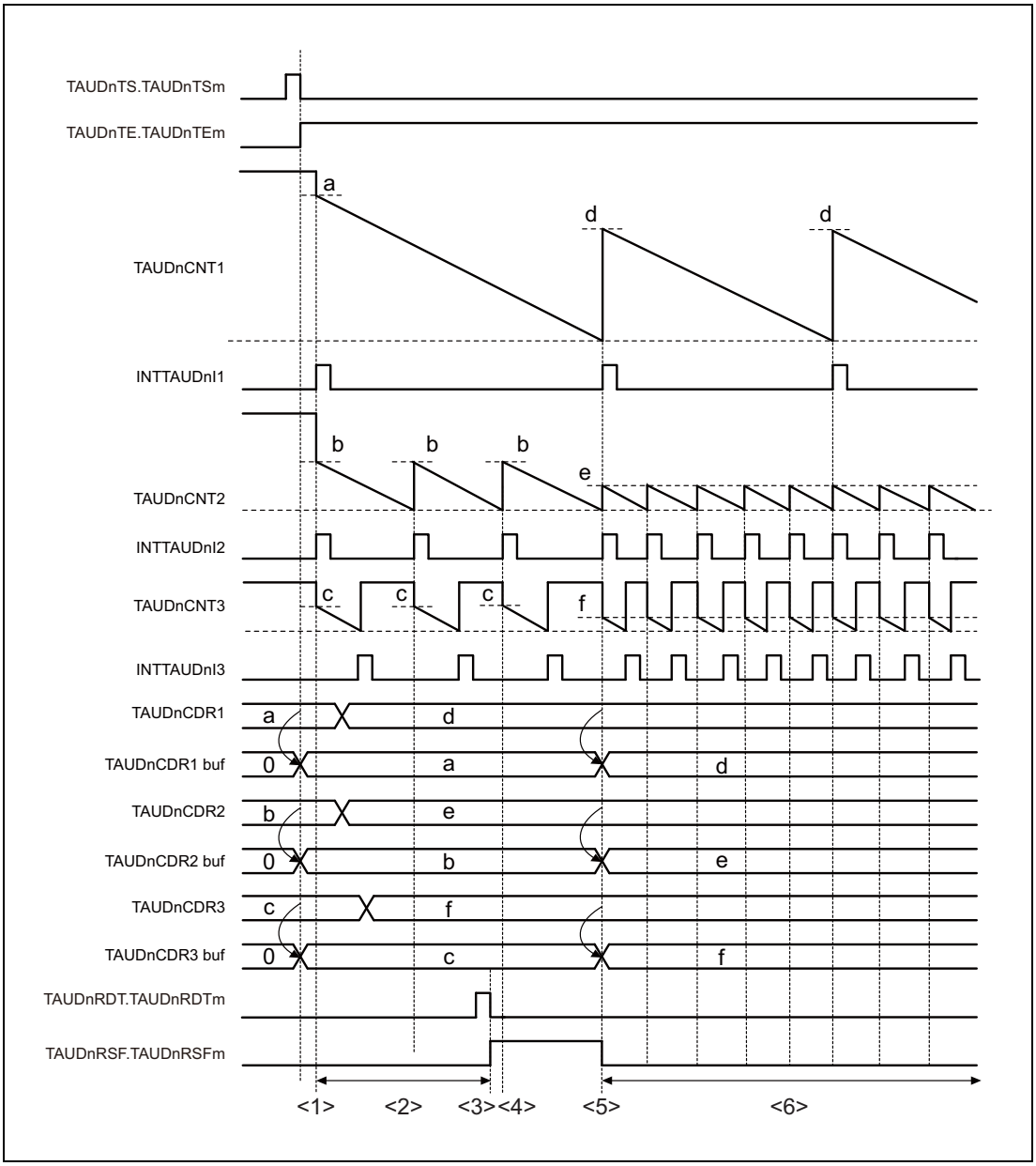


Figure 23.8 Simultaneous Rewrite When INTTAUDnIm Is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm

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 counter1 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 rechanged.

23.6.4.4 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)

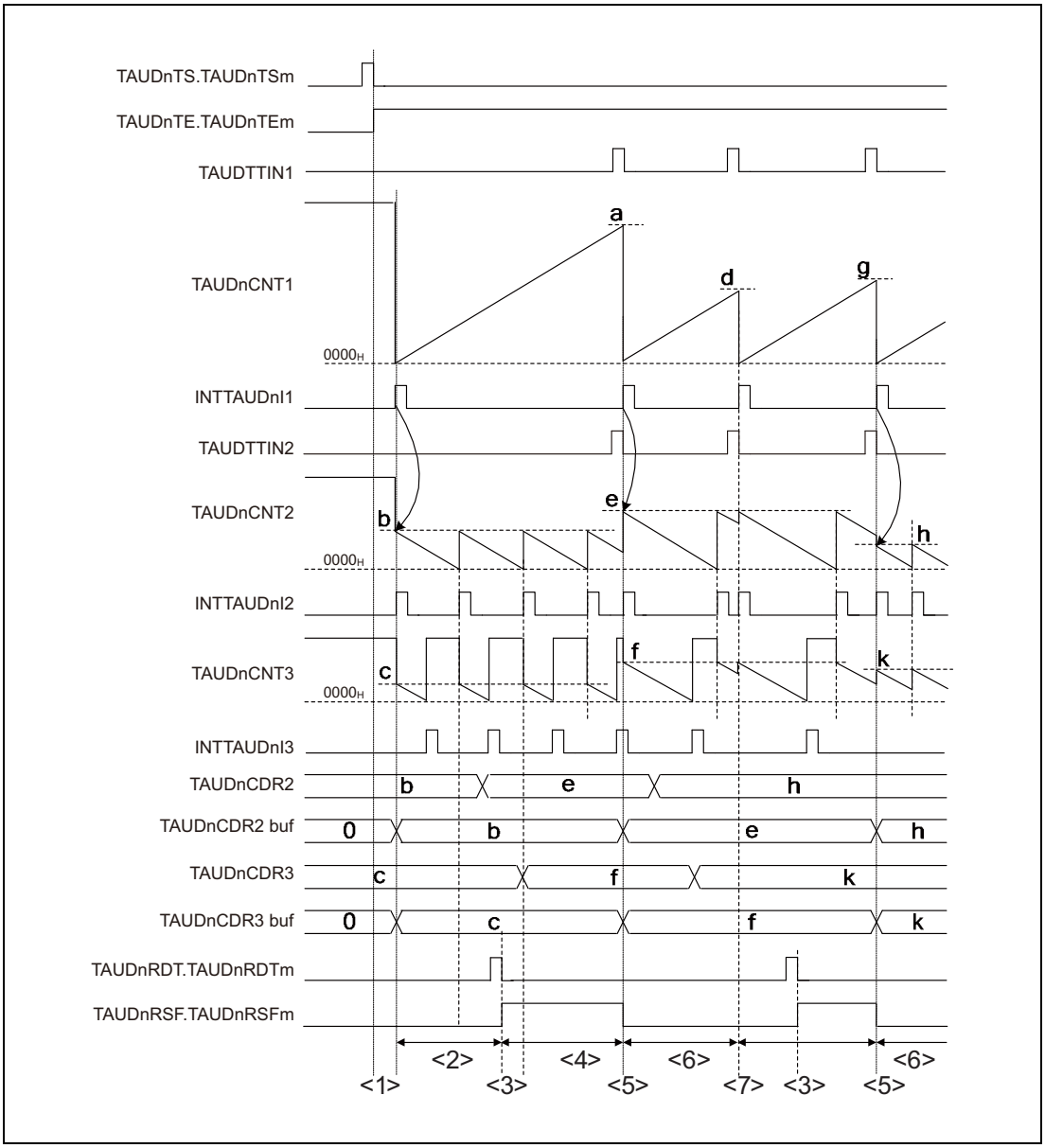


Figure 23.9 Simultaneous Rewrite when INTTAUDnIm is Generated on an Upper Channel Specified by TAUDnRDC.TAUDnRDCm that in Turn is Triggered by an External Signal

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).

23.7 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 23.47, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 23.7.2, Channel Output Modes Controlled Independently by TAUDn Signals**
- **Section 23.7.3, 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 23.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 23.47**.

Table 23.47 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	1	
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.TAUDnTE = 1):
   - TAUDnTOE.TAUDnTOEm
   - 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

23.7.1 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 23.47, Channel Output Modes**, and the output logic using the TAUDnTOL.TAUDnTOLm bit.

(3) Start the counter (TAUDnTS.TAUDnTSm = 1).

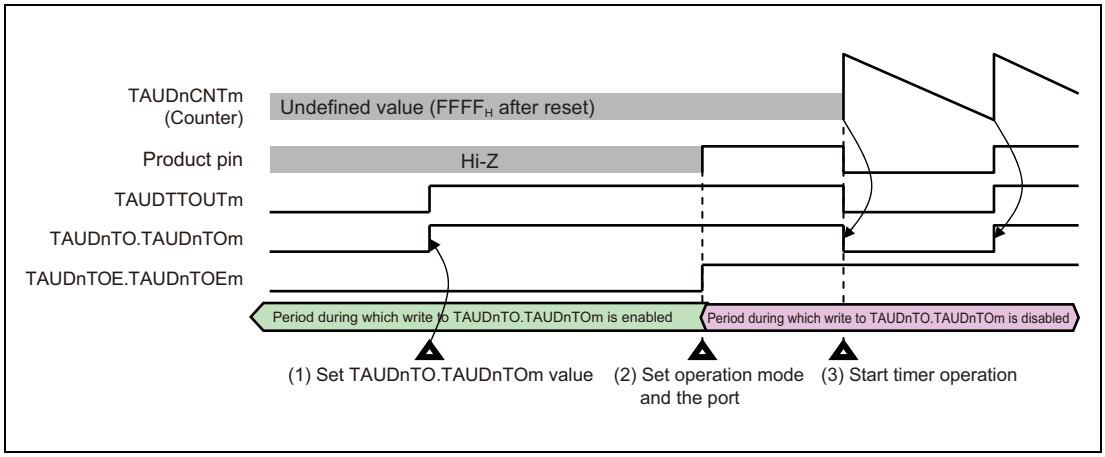


Figure 23.10 General Procedure for Specifying a TAUDTTOUTm Channel Output Mode

23.7.2 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 23.47, Channel Output Modes**.

23.7.2.1 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 23.47, Channel Output Modes**.

23.7.2.2 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 even if TAUDnTRE.TAUDnTREm is set to 0.

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 23.11**.

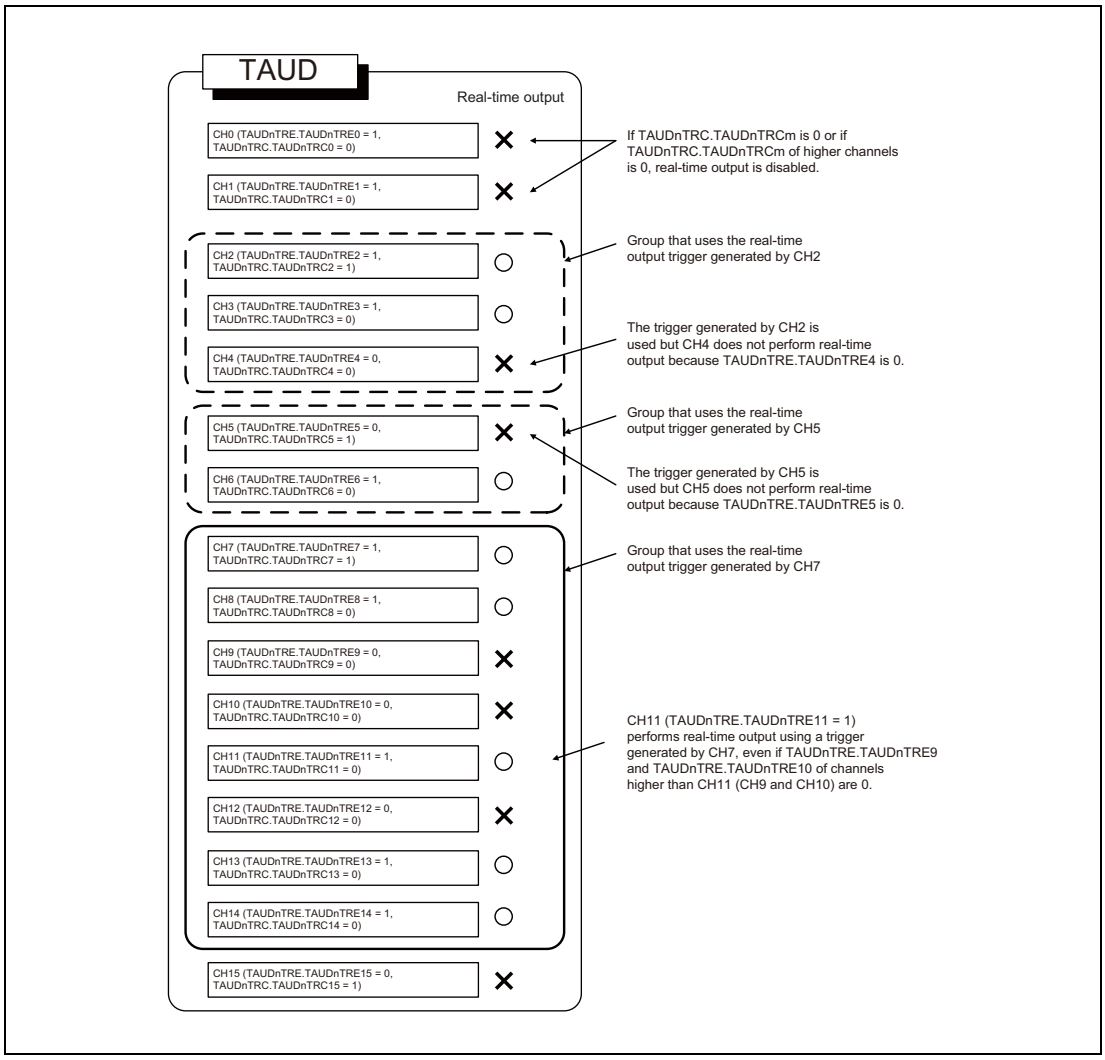


Figure 23.11 Real-Time Output

23.7.2.3 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 23.47, Channel Output Modes**.

23.7.3 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 23.47, Channel Output Modes**.

23.7.3.1 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 23.47, Channel Output Modes**.

23.7.3.2 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.TDLm 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.

23.7.3.3 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 wave at TAUDTTOUTm. For details, see **Section 23.15.7, 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.

23.7.3.4 Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUDnTTOUTm. The set/reset conditions are shown in **Figure 23.12**.

Set/reset conditions

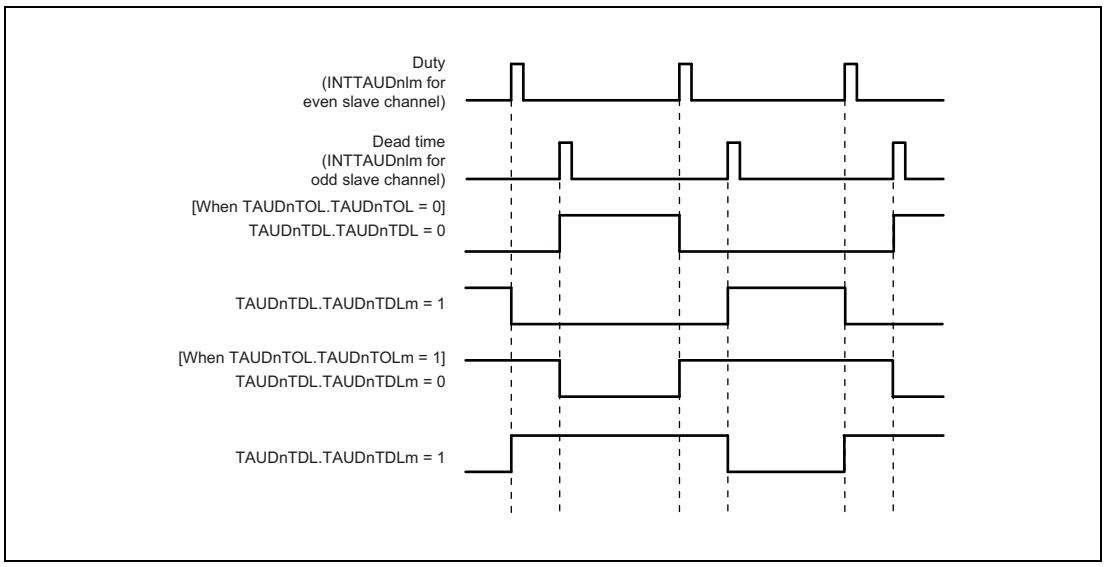


Figure 23.12 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output

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

23.7.3.5 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 23.13**.

Set/reset conditions

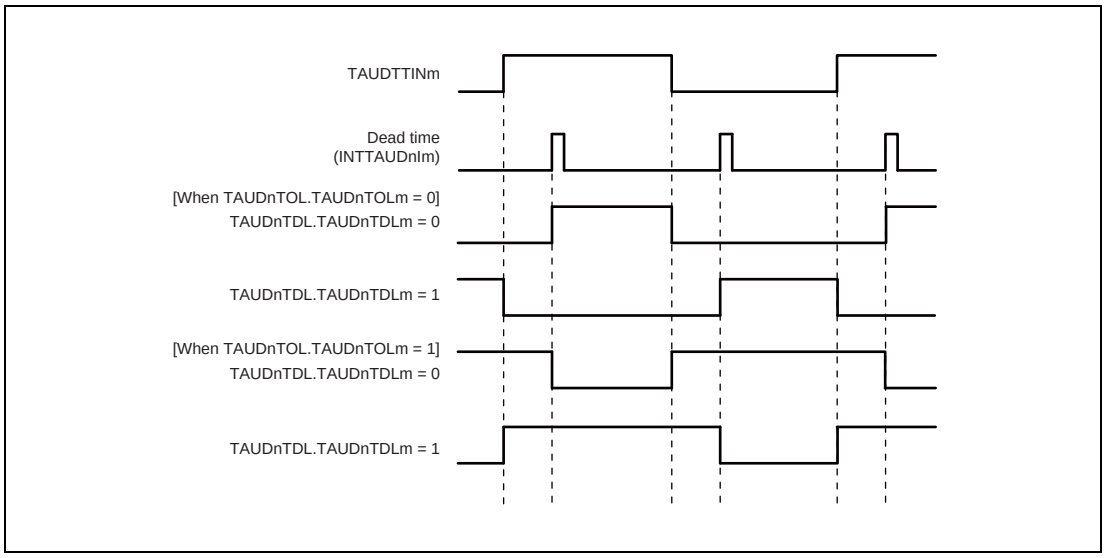


Figure 23.13 Set/Reset Conditions for Synchronous Channel Output Mode 2 with One-Phase PWM Output

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

23.7.3.6 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 23.16.3, 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.

23.7.3.7 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 rectangular wave while mode 2 has a triangular wave.

23.8 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.

23.8.1 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.

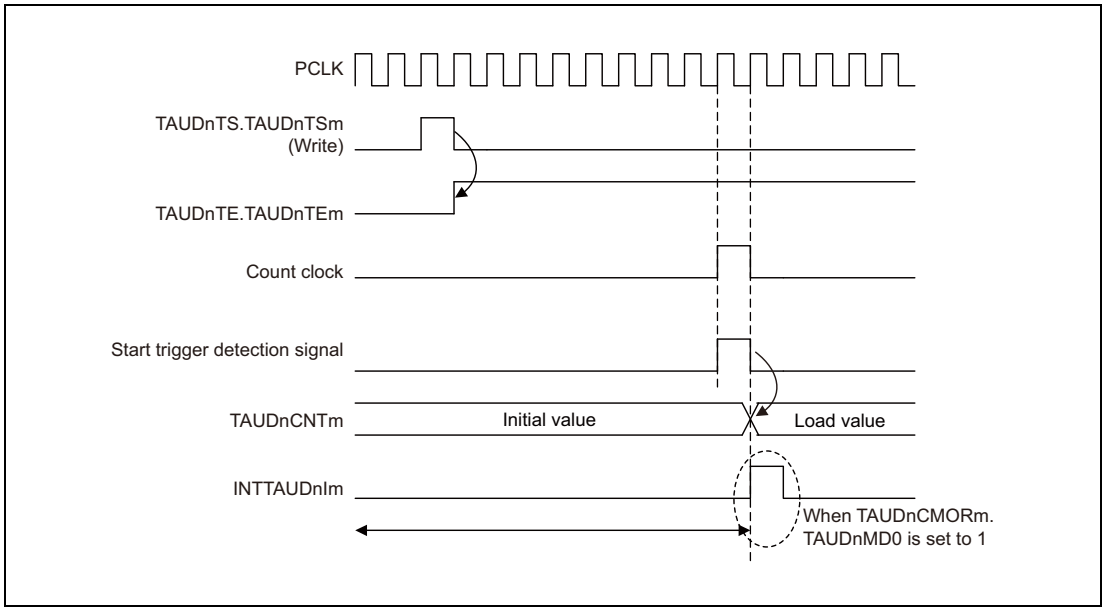


Figure 23.14 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, Count-up/-down Mode, and Count Capture Mode

NOTE

Make sure to set TAUDnCMORm.TAUDnMD0 to 0 when using the count-up/-down mode.

23.8.2 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 increments with subsequent count clocks.

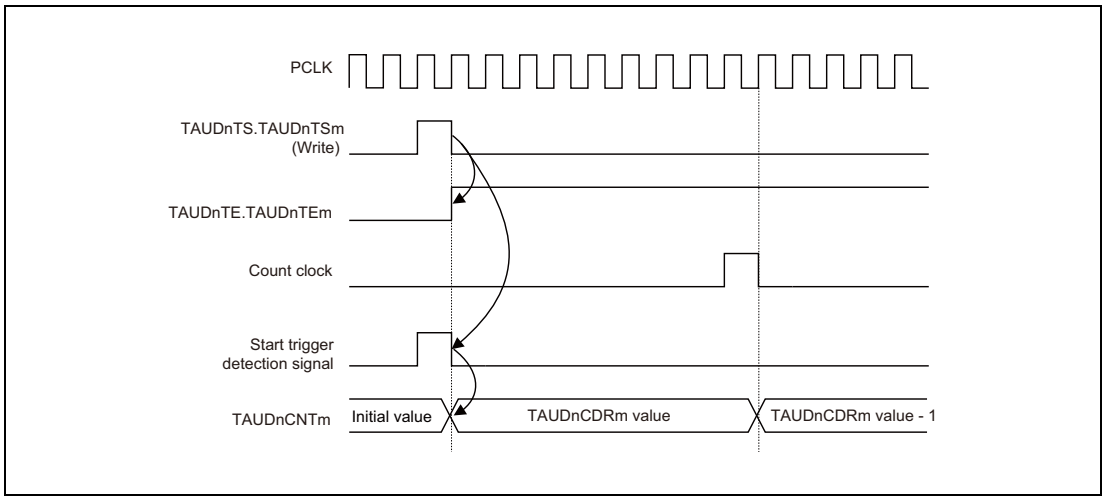


Figure 23.15 Start Timing in Event Count Mode

23.8.3 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 is irrelevant to start of counter operation, determine the frequency with which all operations take place.

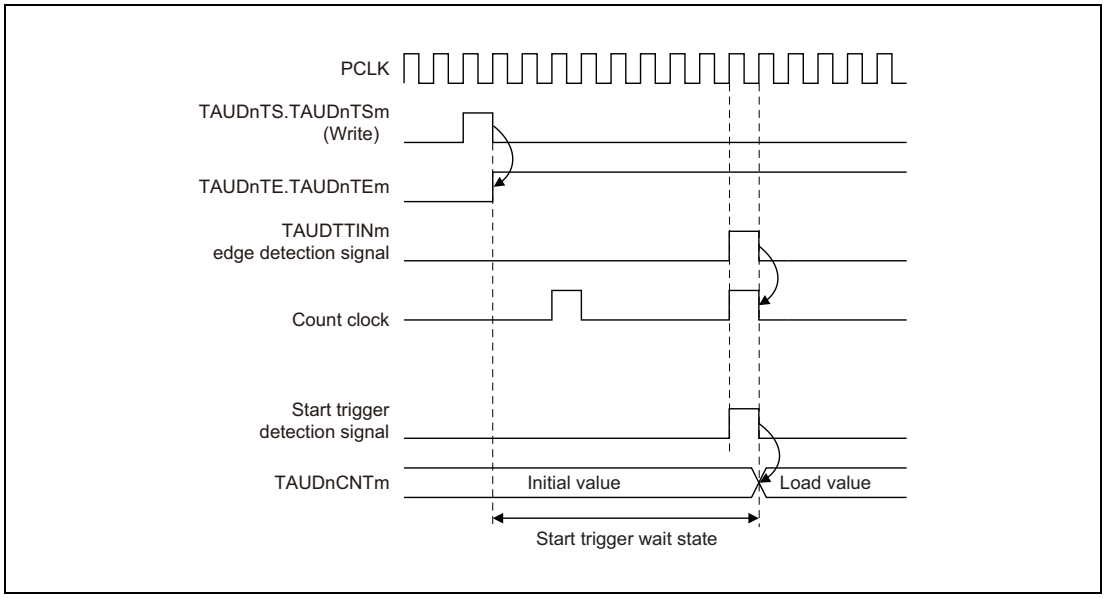


Figure 23.16 Start Timing in Other Operating Modes

23.9 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.

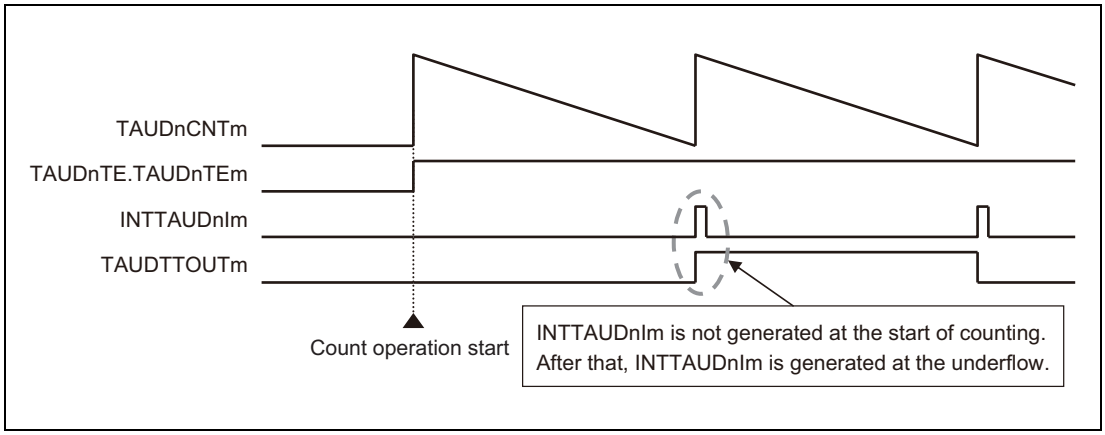


Figure 23.17 INTTAUDnIm Generation Timing (when TAUDnCMORm.TAUDnMD0 = 0)

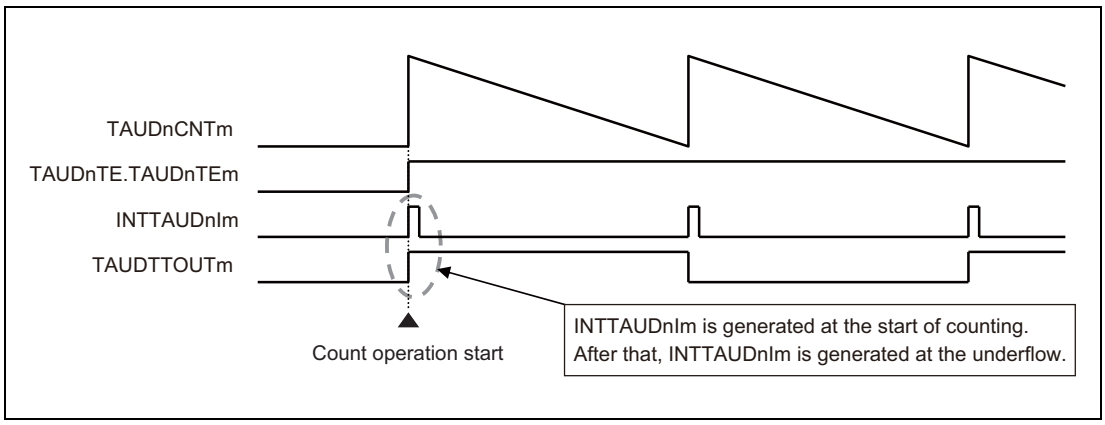


Figure 23.18 INTTAUDnIm Generation Timing (when TAUDnCMORm.TAUDnMD0 = 1)

23.10 Interrupt Generation upon Overflow

Certain independent functions that count up, overflow without generating an interrupt when they reach $FFFF_H$. This section describes how it is possible to generate an interrupt, by combining a channel operating in one of these modes with a channel in a different operation mode which 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.

23.10.1 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 $FFFF_H$.

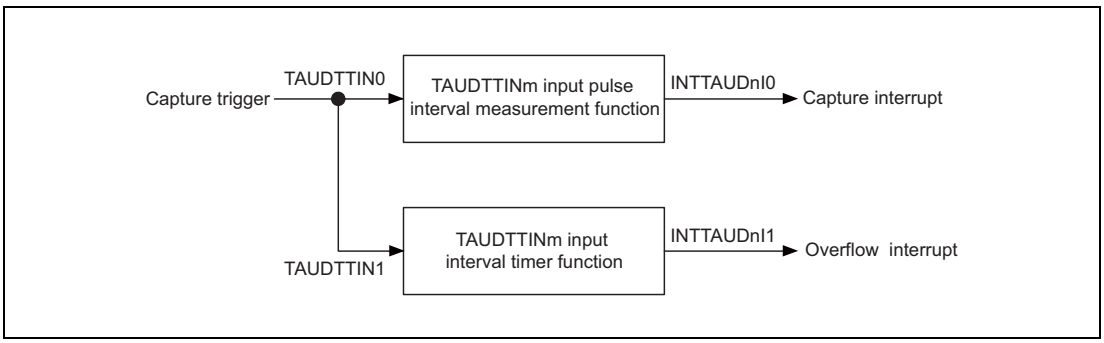


Figure 23.19 Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function

Timing diagram

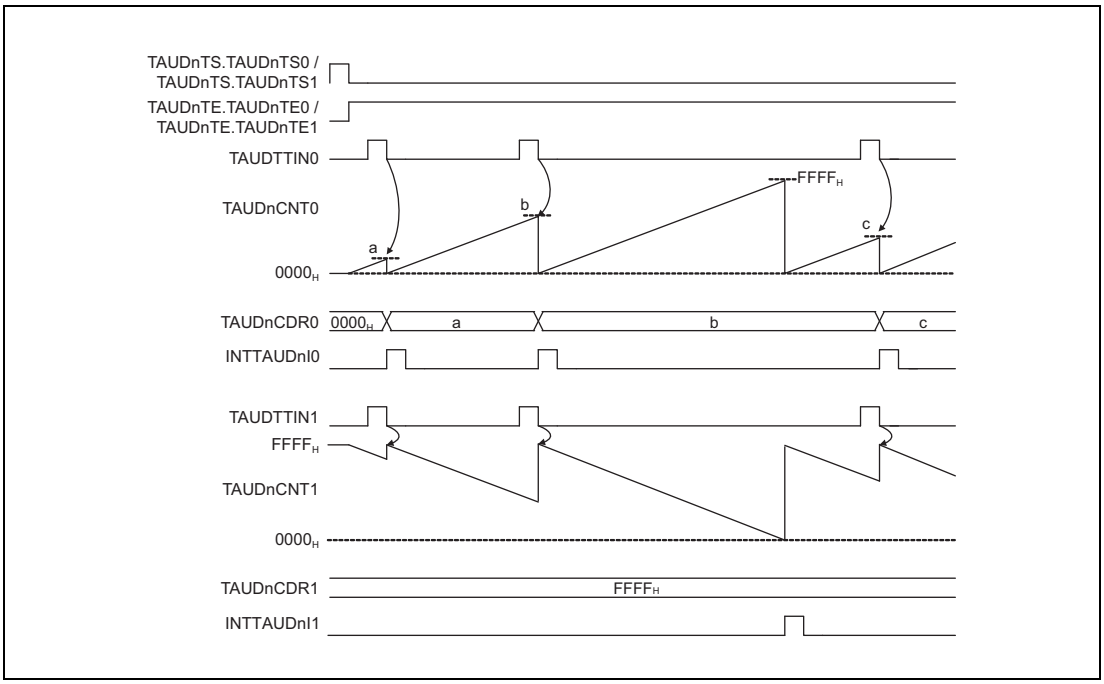


Figure 23.20 Interrupt Generation via Combination of the TAUDTTINm Input Pulse Interval Measurement Function and the TAUDTTINm Input Interval Timer Function

23.10.2 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 $FFFF_H$.

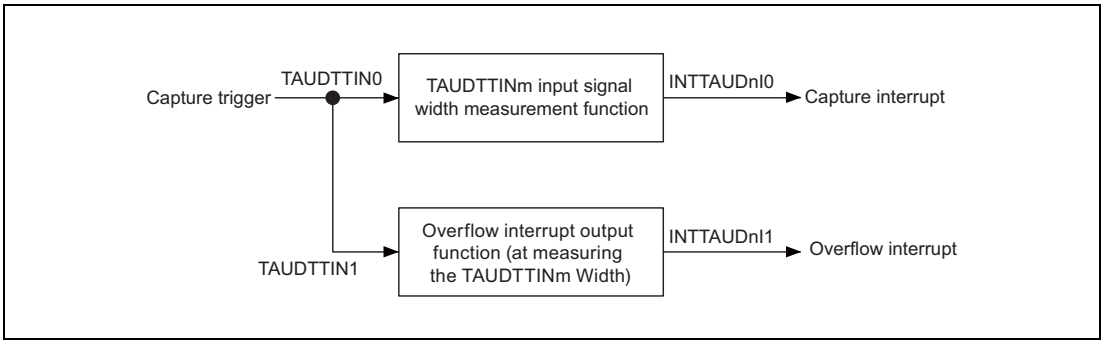


Figure 23.21 Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width)

Timing diagram

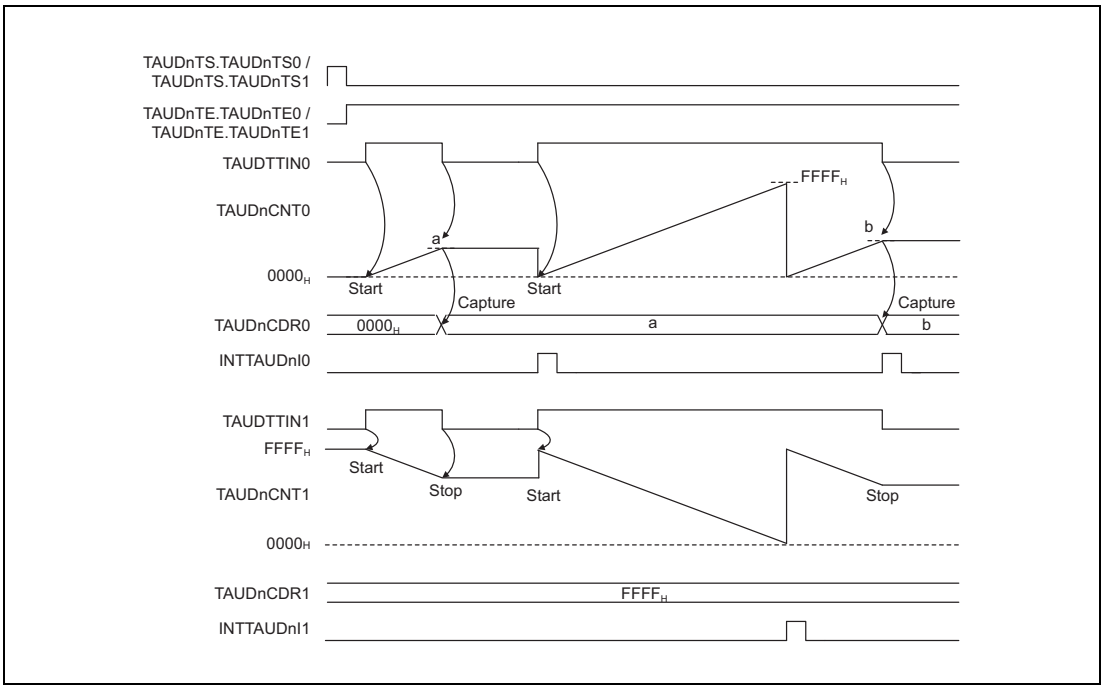


Figure 23.22 Interrupt Generation via Combination of the TAUDTTINm Input Signal Width Measurement Function and the Overflow Interrupt Output Function (at Measuring the TAUDTTINm Width)

23.10.3 Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function

When the counters of both channels are started simultaneously, INTTAUDnIm of the interval timer function can detect the overflow when TAUDnCNTm of the TAUDTTINm input position detection function exceeds $FFFF_H$.

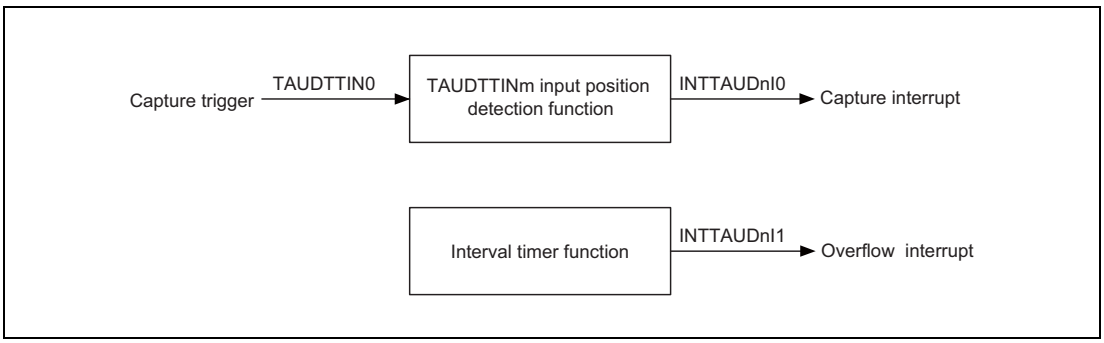


Figure 23.23 Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function

Timing diagram

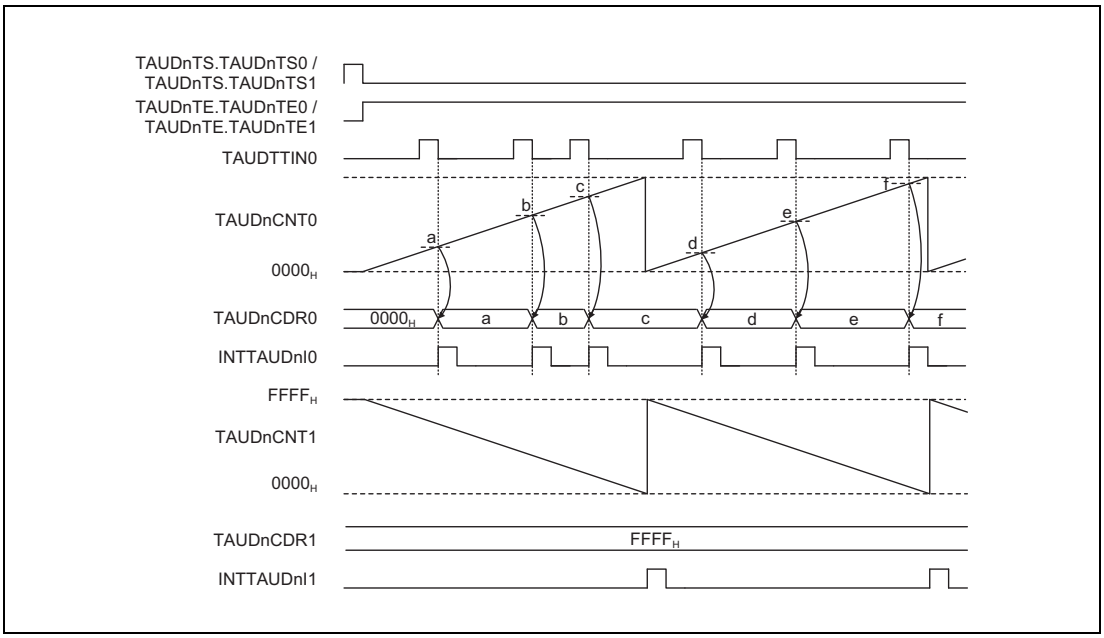


Figure 23.24 Interrupt Generation via Combination of the TAUDTTINm Input Position Detection Function and the Interval Timer Function

23.10.4 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 $FFFF_H$.

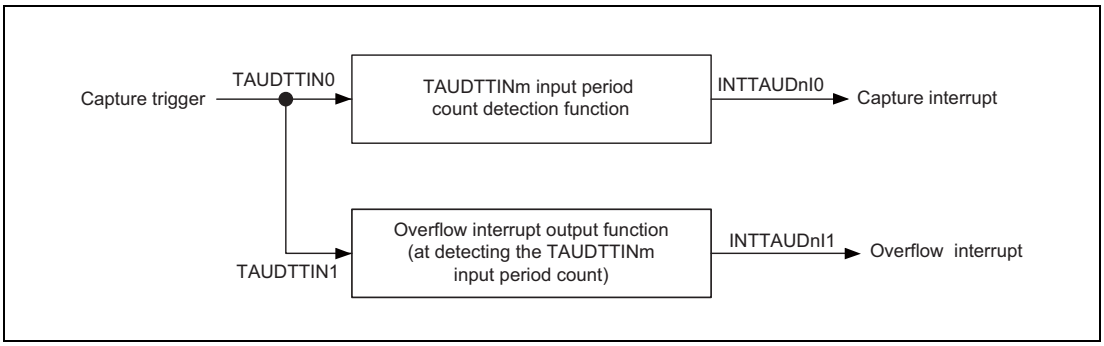


Figure 23.25 Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count)

Timing diagram

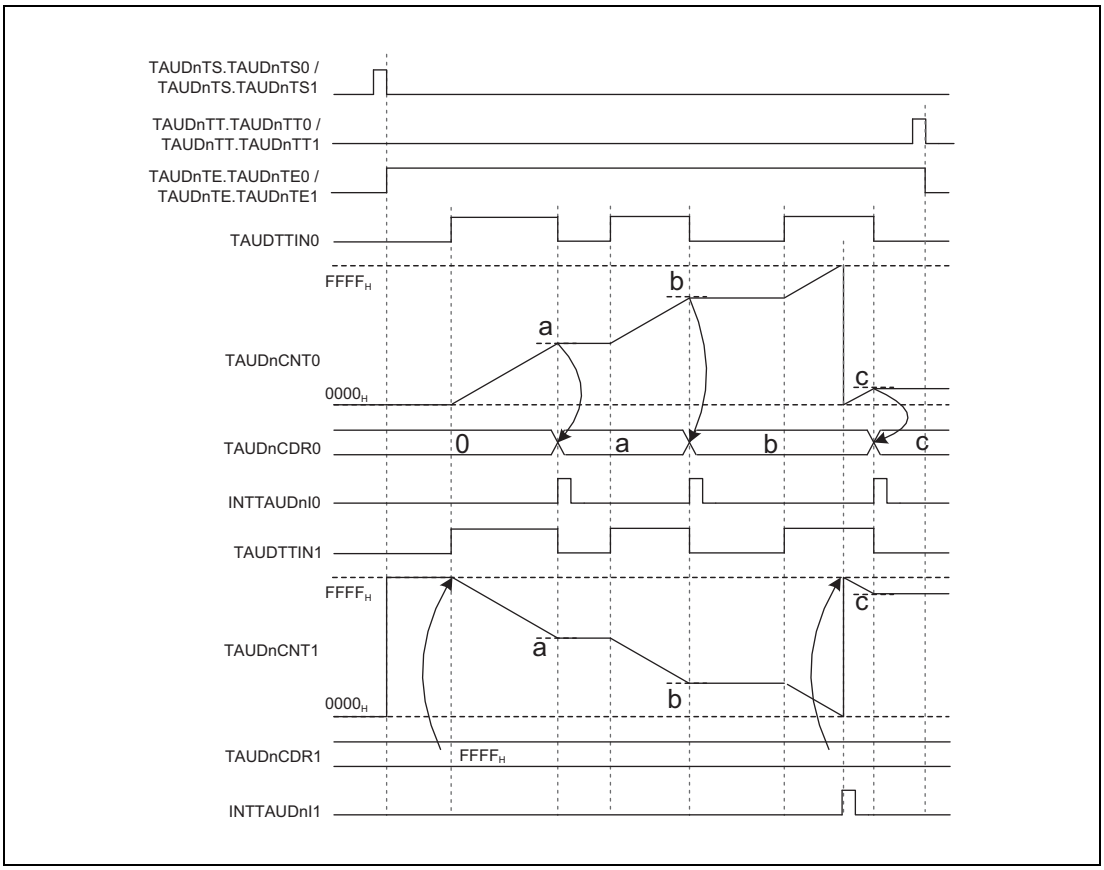


Figure 23.26 Interrupt Generation via Combination of the TAUDTTINm Input Period Count Detection Function and the Overflow Interrupt Output Function (at Detecting the TAUDTTINm Input Period Count)

23.11 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 23.27 shows when edge detection takes place.

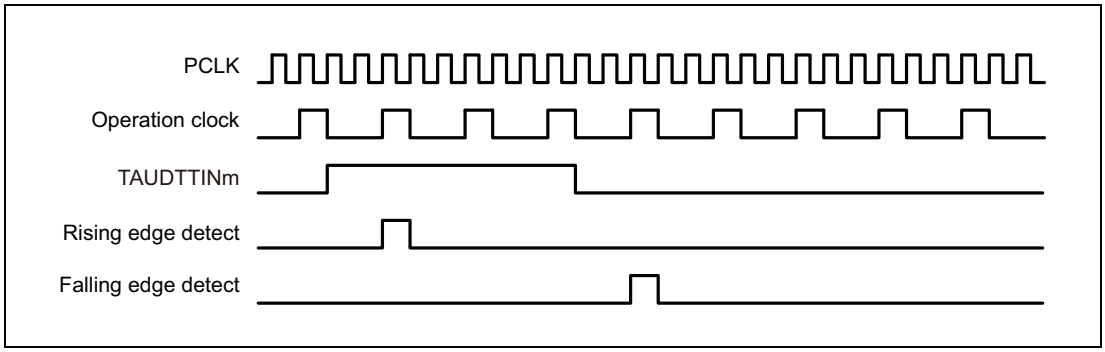


Figure 23.27 Basic Edge Detection Timing

Figure 23.27 shows an operation timing image. Actually, a noise filter or synchronization circuit which is located between the TAUDnlm pin and TAUDn causes a delay time.

23.12 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 23.2, Overview**.

This section describes functions that generate interrupts at regular intervals or with a specified delay.

23.12.1 Interval Timer Function

23.12.1.1 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 23.48, Contents of the TAUDnCMORm Register for Interval Timer Function**.
- The channel output mode must be set to Independent Channel Output Mode 1, see **Section 23.7, 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 TAUDnTTOUTm 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.MD0 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.MD0 is set to 1. For details see **Section 23.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

23.12.1.2 Equations

INTTAUDnIm cycle = count clock cycle × (TAUDnCDRm + 1)

TAUDTTOUTm square wave cycle = count clock cycle × (TAUDnCDRm + 1) × 2

23.12.1.3 Block Diagram and General Timing Diagram

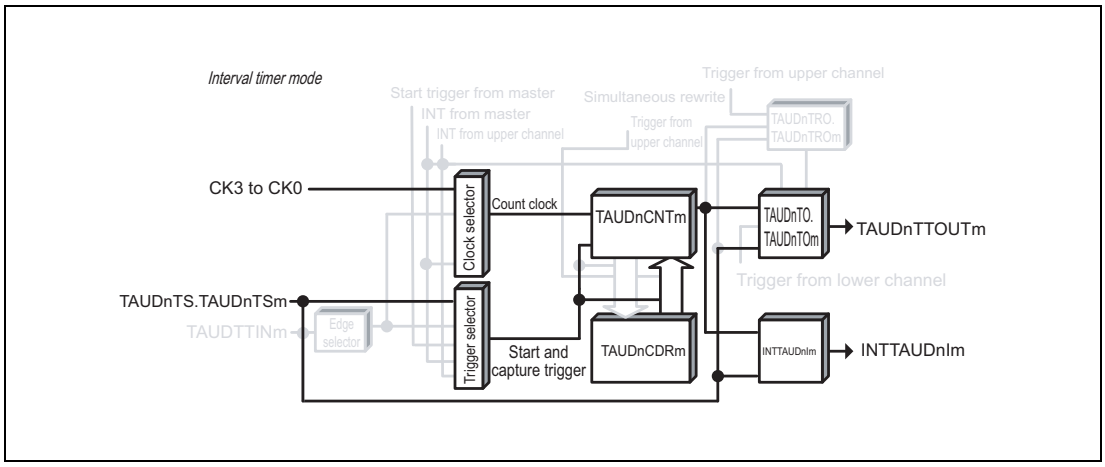


Figure 23.28 Block Diagram of Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation (TAUDnCMORm.TAUDnMD0 = 1).

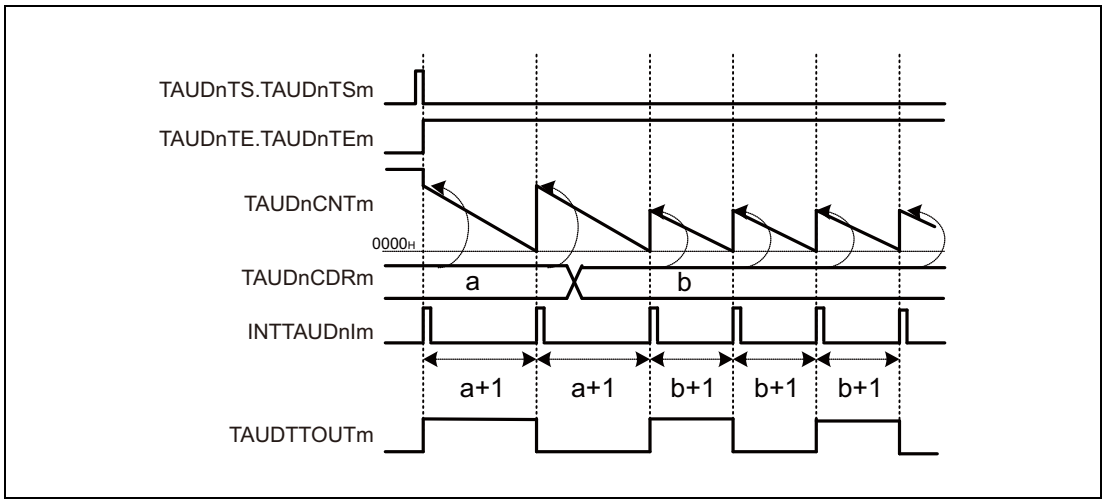


Figure 23.29 General Timing Diagram of Interval Timer Function

23.12.1.4 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 23.48 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 to this bit, 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 23.49 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode

Table 23.50 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 23.7, 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 23.51 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	

23.12.1.5 Operating Procedure for Interval Timer Function

Table 23.52 Operating Procedure for Interval Timer Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.48, Contents of the TAUDnCMORm Register for Interval Timer Function**, and **Table 23.49, 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 23.50, 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.MD0 = 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

23.12.1.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H, count clock = PCLK/2

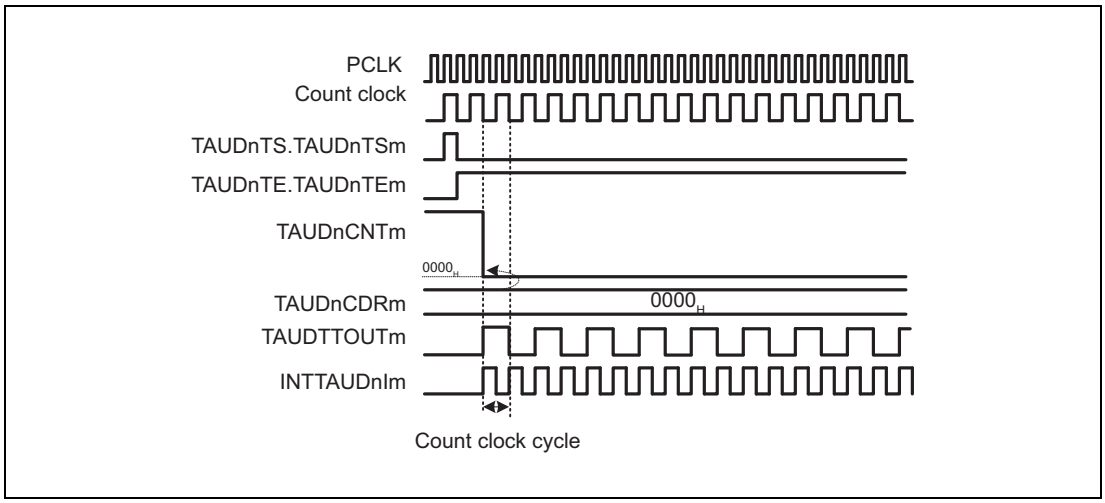


Figure 23.30 TAUDnCDRm = 0000_H, Count Clock = PCLK/2

- 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

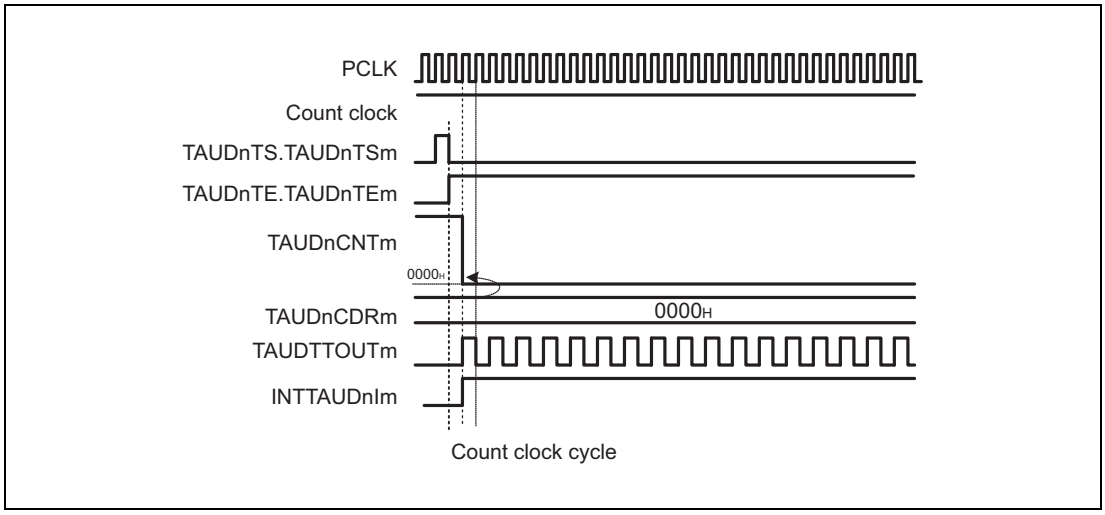


Figure 23.31 TAUDnCDRm = 0000_H, Count Clock = PCLK

- 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 generated continuously, resulting in TAUDTTOUTm toggling every PCLK clock.

NOTE

INTTAUDnlm is fixed at high level.

(3) Operation stop and restart

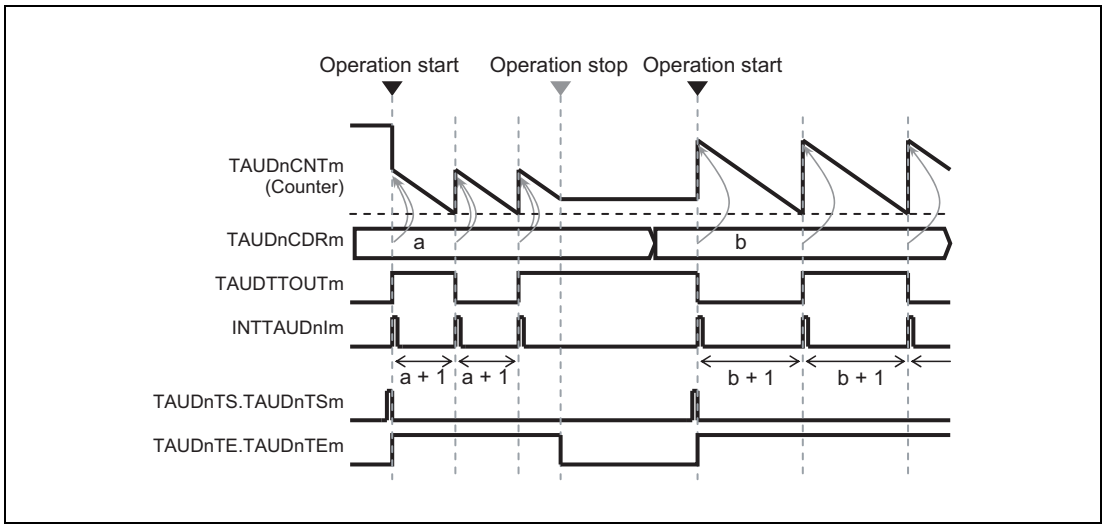


Figure 23.32 Operation Stop and Restart (TAUDnCMORm.TAUDnMD0 = 1)

- 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)

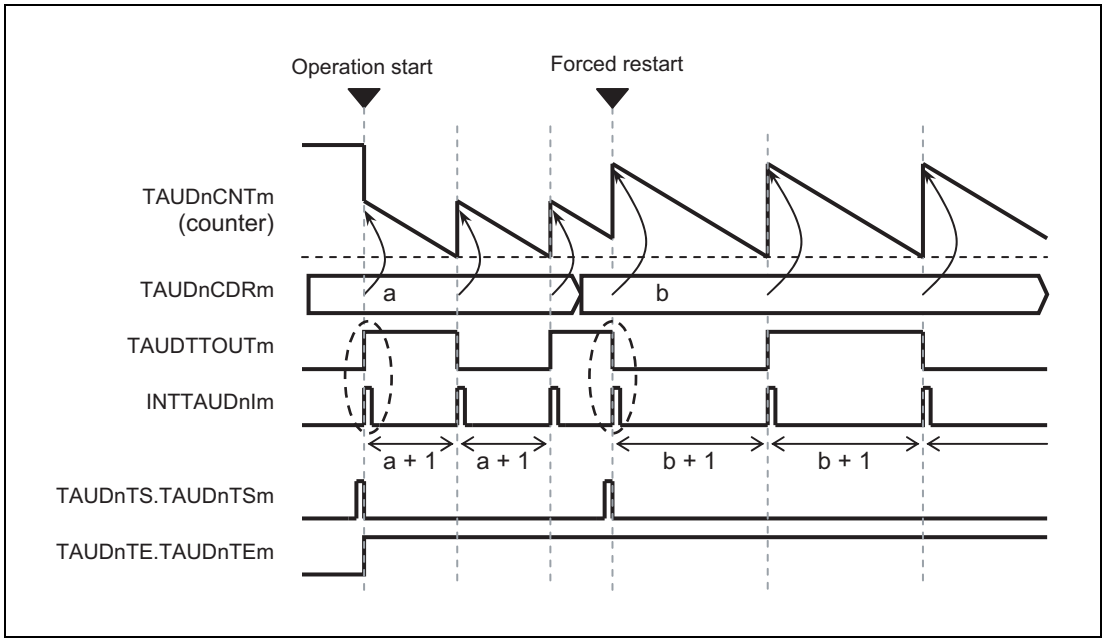


Figure 23.33 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 1)

- 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 (INTTAUDnlm) is generated and TAUDTTOUTm is inverted.

(5) Forced restart (TAUDnCMORm.TAUDnMD0 = 0)

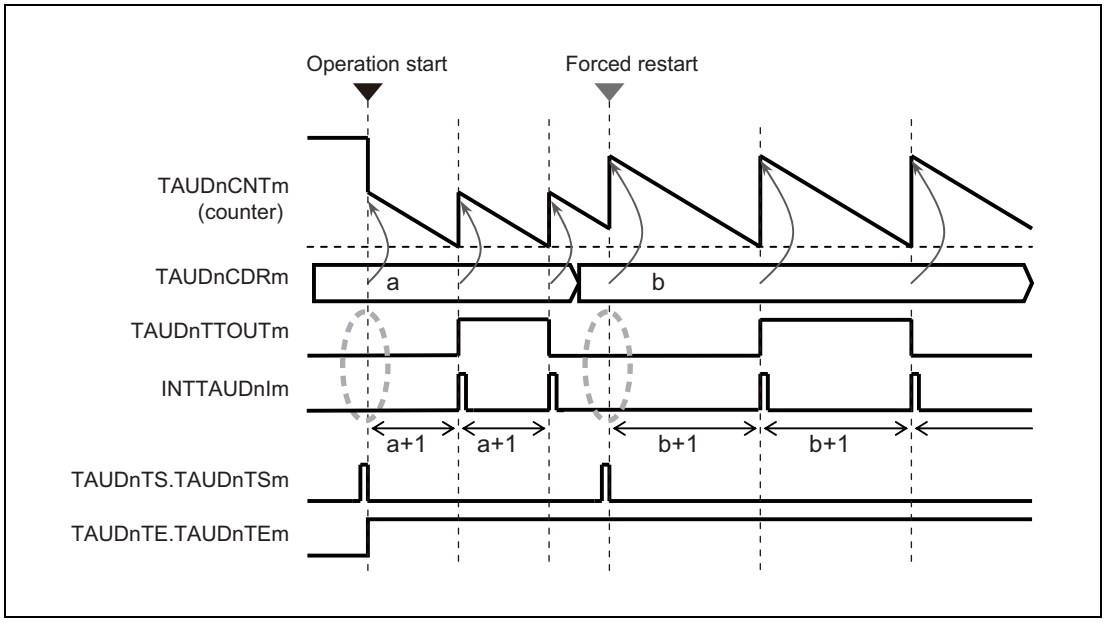


Figure 23.34 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 0)

- When a forced restart is made, an interrupt (INTTAUDnlm) is not generated and TAUDTTOUTm is not inverted.

23.12.2 TAUDTTINm Input Interval Timer Function

23.12.2.1 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 23.53, 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 23.7, Channel Output Modes**.

Functional description

This function operates in an identical manner to the interval timer function (see **Section 23.12.1, 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.

23.12.2.2 Equations

INTTAUDnIm cycle = count clock cycle × (TAUDnCDRm + 1)

TAUDTTOUTm square wave cycle = count clock cycle × (TAUDnCDRm + 1) × 2

23.12.2.3 Block Diagram and General Timing Diagram

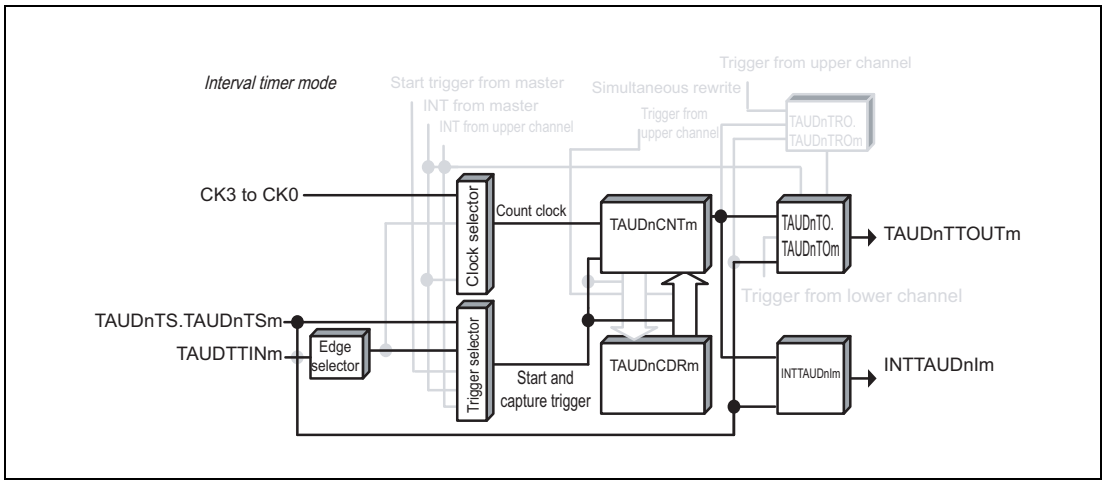


Figure 23.35 Block Diagram of TAUDTTINm Input Interval Timer Function

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)

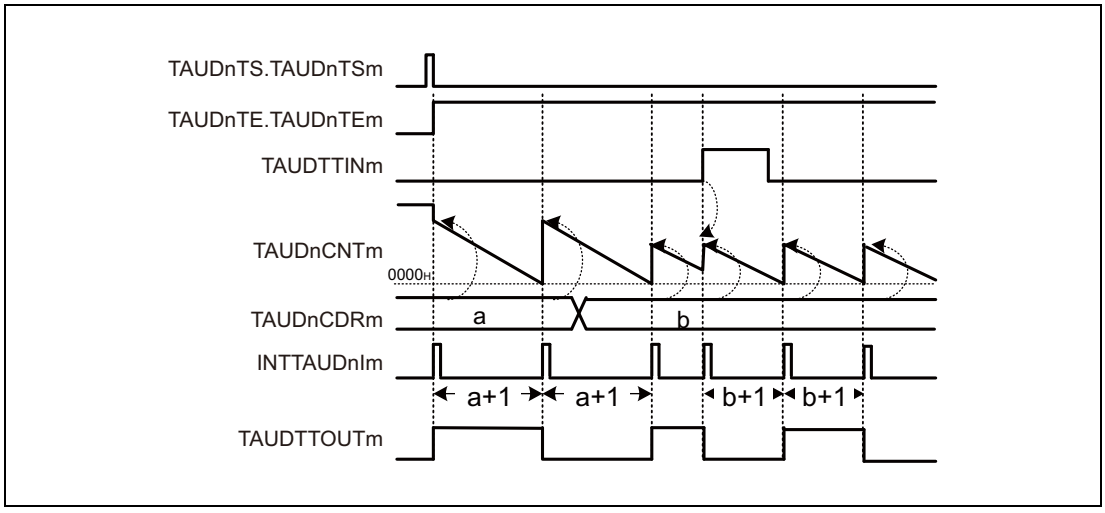


Figure 23.36 General Timing Diagram of TAUDTTINm Input Interval Timer Function

23.12.2.4 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 23.53 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 to this bit, 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 23.54 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 to these bits, 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

(3) Channel output mode

Table 23.55 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. TAUDnTTOUTm can then be controlled independently of the interrupts. For details, see **Section 23.7, 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 23.56 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	

23.12.2.5 Operating Procedure for TAUDTTINm Input Interval Timer Function

Table 23.57 Operating Procedure for TAUDTTINm Input Interval Timer Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.53, Contents of the TAUDnCMORm Register for TAUDTTINm Input Interval Timer Function**, and **Table 23.54, 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 23.55, 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

23.12.2.6 Specific Timing Diagrams

The timing diagrams in **Section 23.12.1, Interval Timer Function**. The counter can also be restarted by a valid TAUDTTINm input edge without using this function.

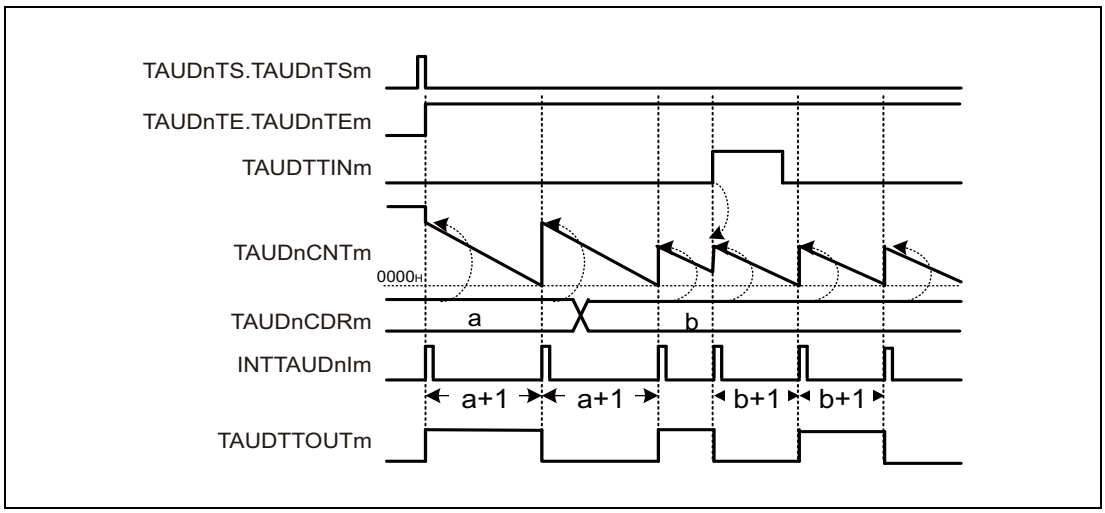


Figure 23.37 Counter Triggered by Rising TAUDTTINm Input Edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B), TAUDnCMORm.TAUDnMD0 = 1

- If a valid TAUDTTINm input edge is detected, an interrupt is generated which causes TAUDTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

23.12.3 Clock Divide Function

23.12.3.1 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 23.58, Contents of the TAUDnCMORm Register for Clock Divide Function**.)
- The channel output mode should be set to independent channel output mode 1. (See **Section 23.7, 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 23.9, 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.

23.12.3.2 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)

23.12.3.3 Block Diagram and General Timing Diagram

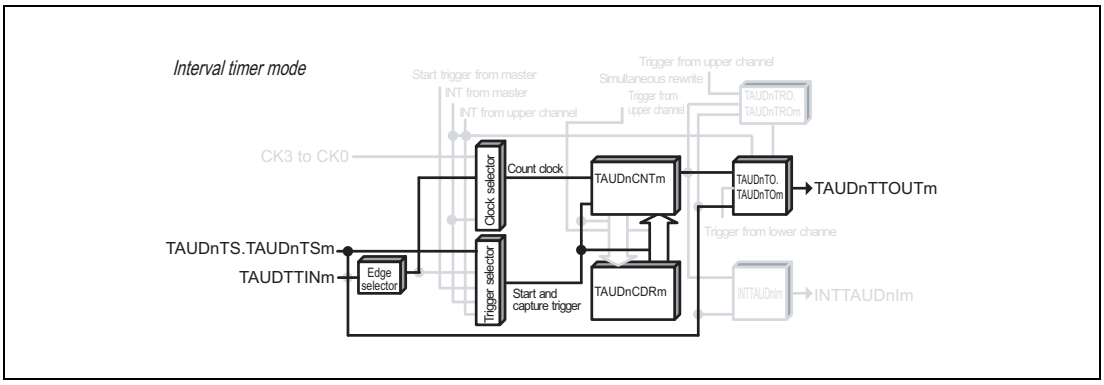


Figure 23.38 Block Diagram of Clock Divide Function

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)

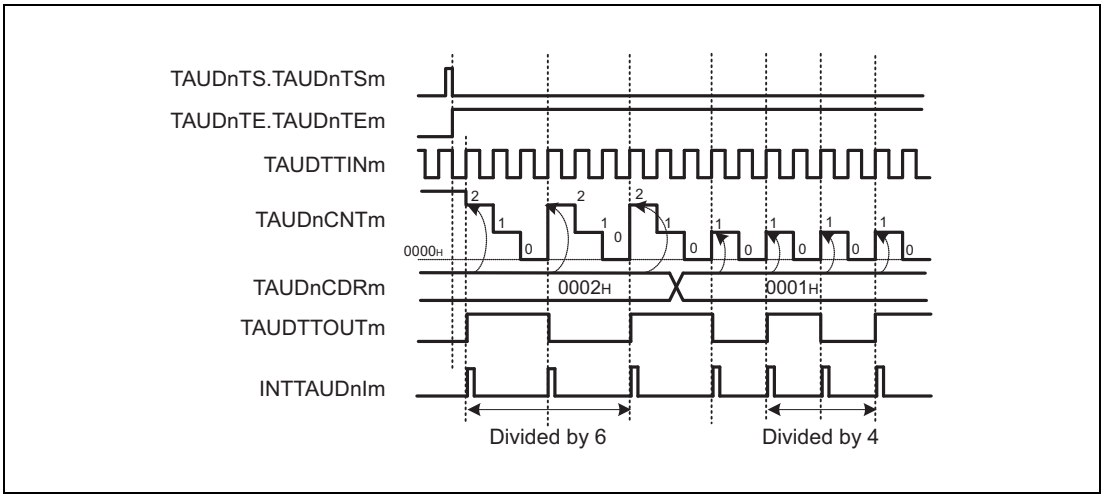


Figure 23.39 General Timing Diagram of Clock Divide Function

23.12.3.4 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 23.58 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 to this bit, 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 23.59 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 to these bits, 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

(3) Channel output mode

Table 23.60 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 this function. Therefore, these registers should be set to 0.

Table 23.61 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	

23.12.3.5 Operating Procedure for Clock Divide Function

Table 23.62 Operating Procedure for Clock Divide Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.58, Contents of the TAUDnCMORm Register for Clock Divide Function**, and **Table 23.59, 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 23.60, 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

23.12.3.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H

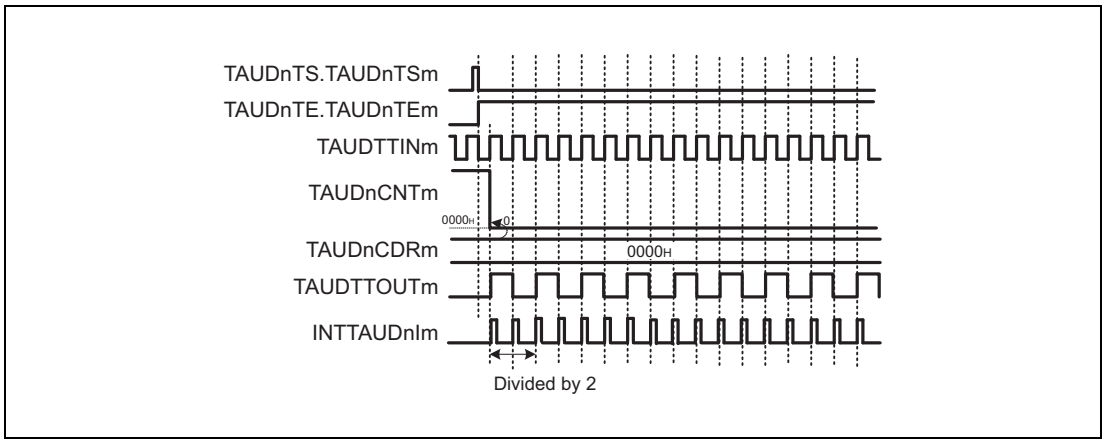


Figure 23.40 TAUDnCDRm = 0000_H, TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B

- If TAUDnCDRm is 0000_H, TAUDnCNTm is always 0000_H.
- INTTAUDnIm is generated every count clock, resulting in TAUDTTOUTm toggling every count clock.

Figure 23.40 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 TAUDnlm pin and TAUDn.

(2) Restart

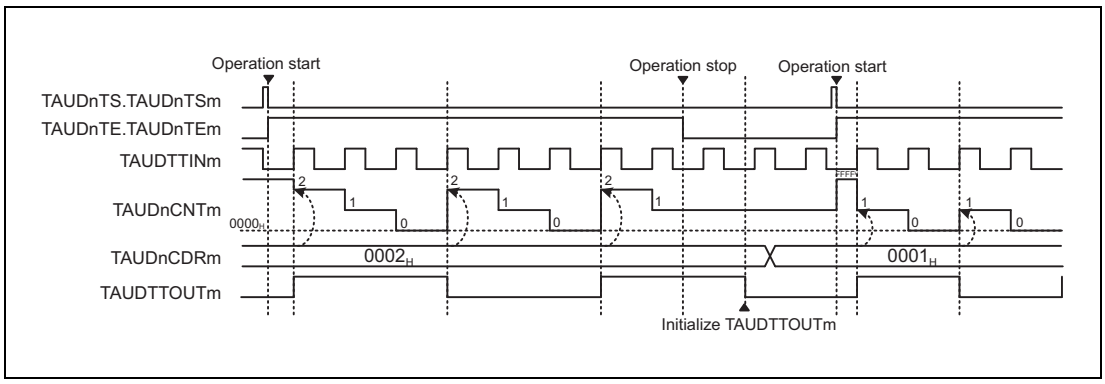


Figure 23.41 Restart (TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

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

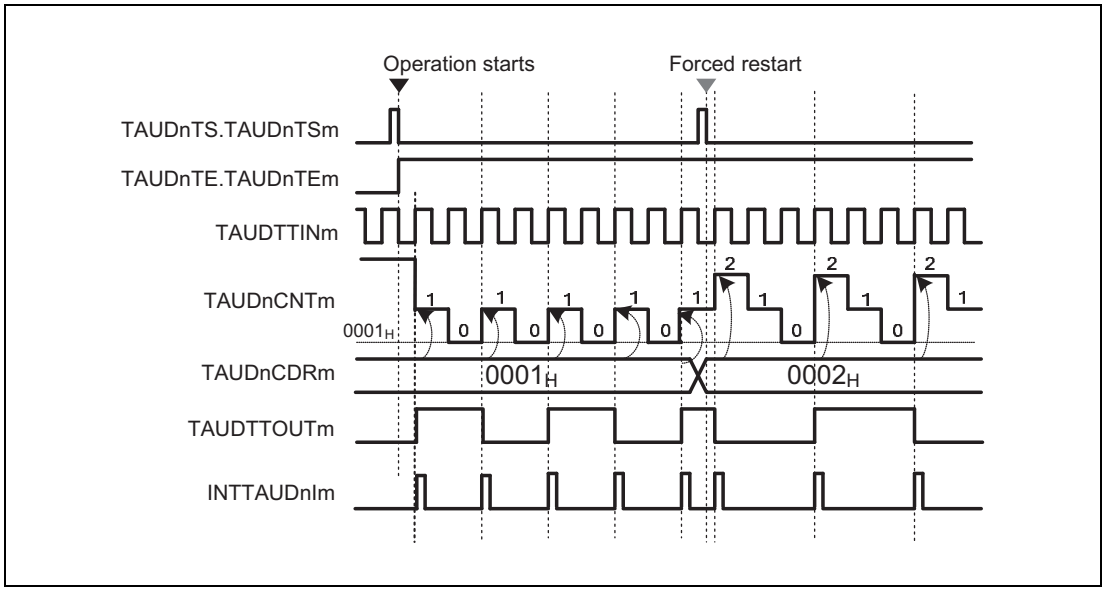


Figure 23.42 Forced Restart Operation (TAUDnCMORm.TAUDnMD0 = 1, TAUDnCMURm.TAUDnTIS[1:0] = 01)

- 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.

23.12.4 External Event Count Function

23.12.4.1 Overview

Summary

This function is used as an event timer, which generates an interrupt (INTTAUDnIm) when a specific number of TAUDTTINm input pulses has occurred.

Prerequisites

- The operating mode should be set to the event count mode. (See **Table 23.63, 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.

23.12.4.2 Equations

Number of valid edges detected before INTTAUDnIm generation = TAUDnCDRm + 1

23.12.4.3 Block Diagram and General Timing Diagram

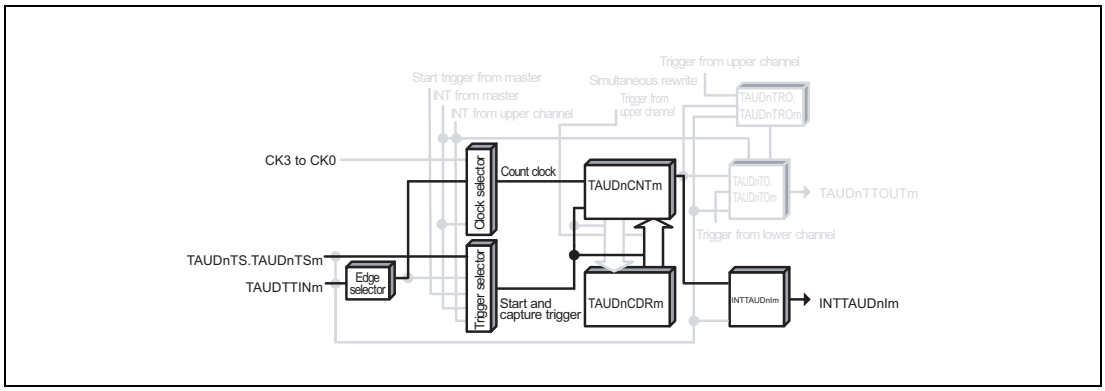


Figure 23.43 Block Diagram of External Event Count Function

The following settings apply to the general timing diagram.

- Detection of rising edge (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

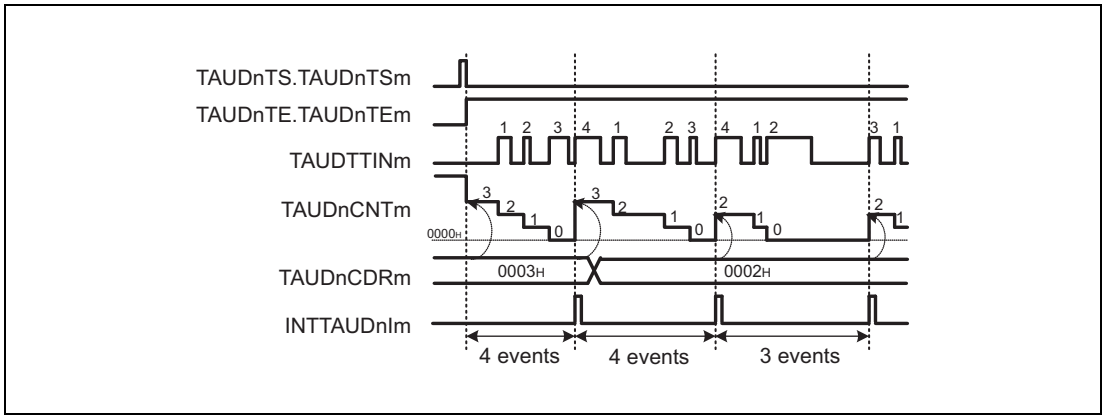


Figure 23.44 General Timing Diagram of External Event Count Function

23.12.4.4 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 23.63 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 to this bit, 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 23.64 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Falling edge 01: Rising edge 10: Falling and rising edges

(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 this function. Therefore, these registers should be set to 0.

Table 23.65 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	

23.12.4.5 Operating Procedure for External Event Count Function

Table 23.66 Operating Procedure for External Event Count Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.63, Contents of the TAUDnCMORm Register for External Event Count Function**, and **Table 23.64, 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 the counter reaches 0000_H: • 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

23.12.4.6 Specific Timing Diagrams

(1) TAUDnCDRm = 0000_H

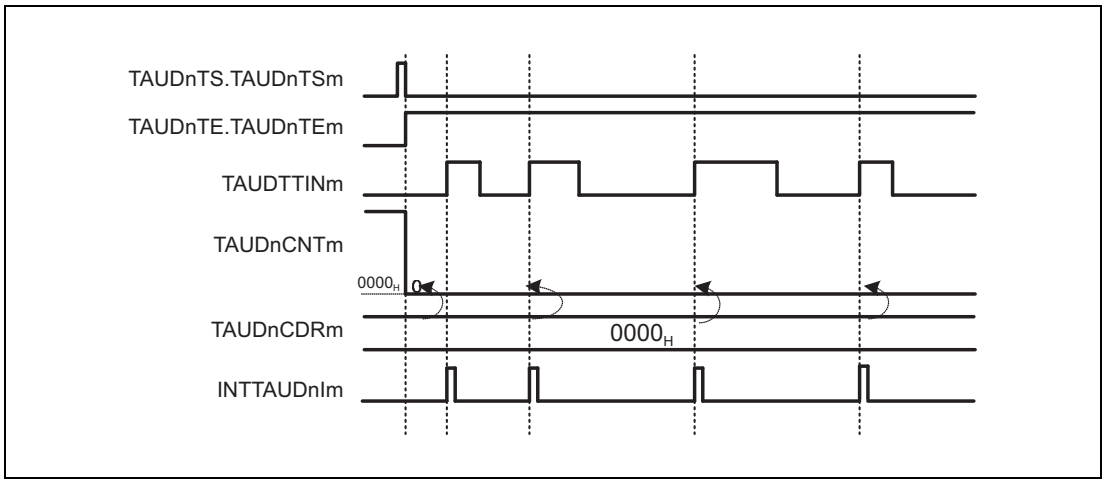


Figure 23.45 TAUDnCDRm = 0000_H, TAUDnCMURm.TAUDnTIS[1:0] = 01_B

- If 0000_H = TAUDnCDRm, 0000_H is loaded into TAUDnCNTm each time a valid TAUDTTINm input edge is detected.
  In other words, INTTAUDnlm is generated each time a valid TAUDTTINm input edge is detected.

(2) Operation stop and restart

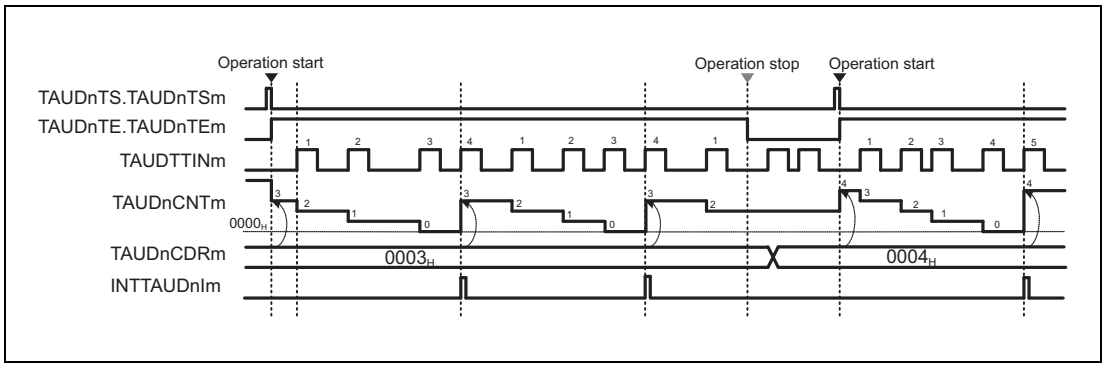


Figure 23.46 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

- 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

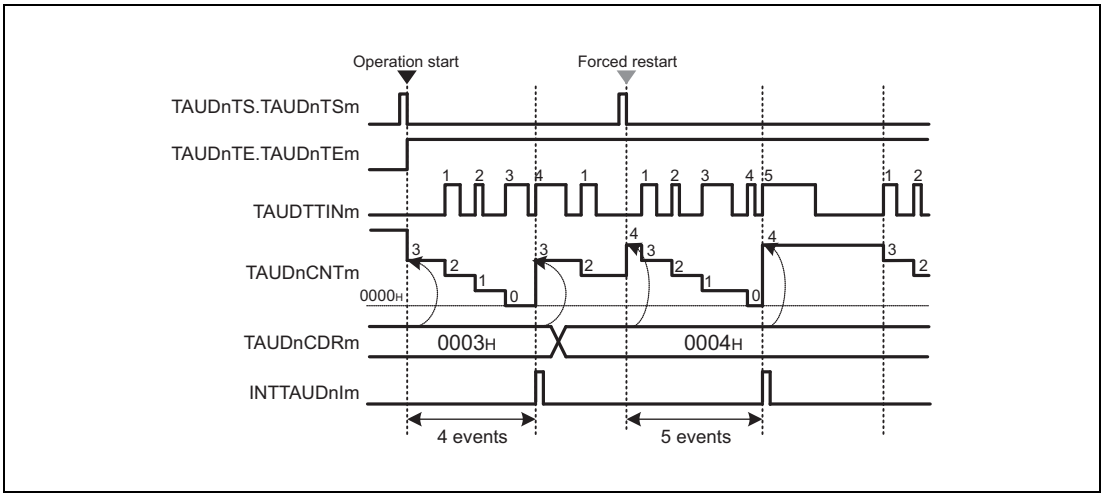


Figure 23.47 Forced Restart Operation (TAUDnCMURm.TAUDnTIS[1:0] = 01_B)

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.

23.12.5 Delay Count Function

23.12.5.1 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 23.67, 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.TIS[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.

23.12.5.2 Equations

Delay between TAUDTTINm and INTTAUDnIm = count clock cycle × (TAUDnCDRm + 1)

23.12.5.3 Block Diagram and General Timing Diagram

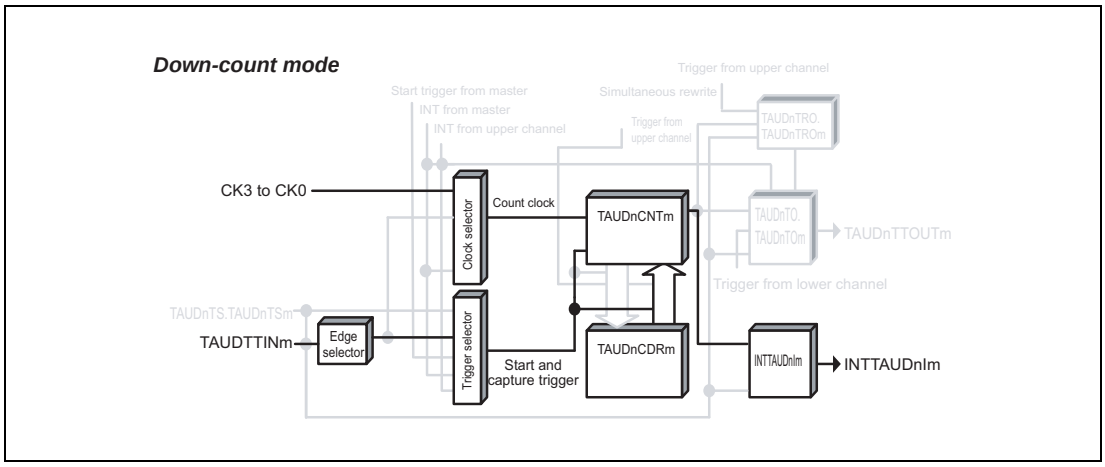


Figure 23.48 Block Diagram of Delay Count Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

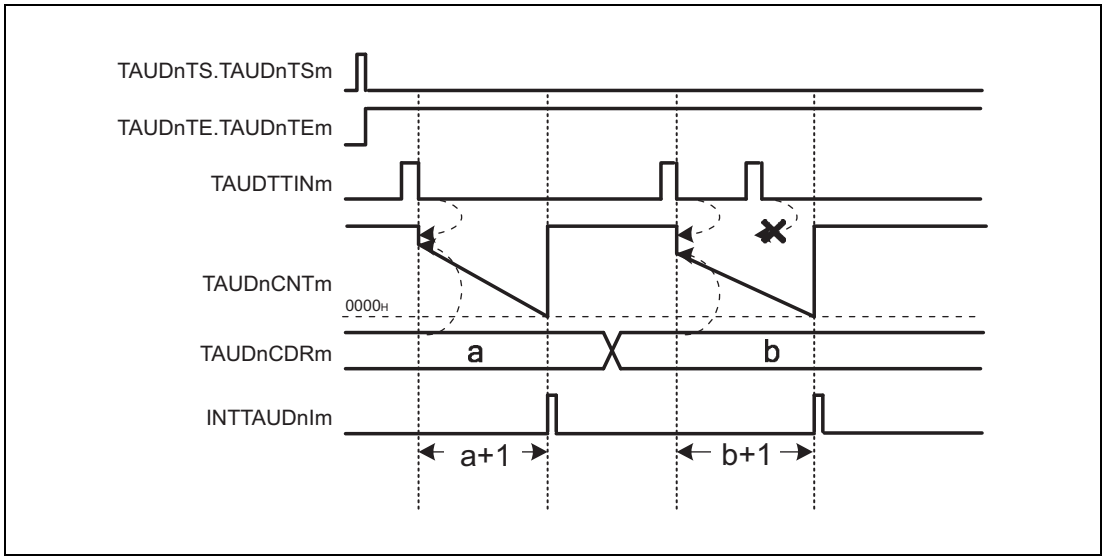


Figure 23.49 General Timing Diagram of Delay Count Function

23.12.5.4 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 23.67 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 to this bit, 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 23.68 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 to these bits, 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 this function. Therefore, these registers should be set to 0.

Table 23.69 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	

23.12.5.5 Operating Procedure for Delay Count Function

Table 23.70 Operating Procedure for Delay Count Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.67, Contents of the TAUDnCMORm Register for Delay Count Function**, and **Table 23.68, Contents of the TAUDnCMURm Register for Delay 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. 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.

Restart Operation

23.12.6 One-Pulse Output Function

23.12.6.1 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 channel output mode should be set to independent channel output mode 2. (See **Table 23.71, Contents of the TAUDnCMORm Register for One-Pulse Output Function**.)
- The channel output mode should be set to independent channel output mode 1. (See **Section 23.7, 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 toggles.

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.

23.12.6.2 Equations

Interval between TAUDTTINm and INTTAUDnIm = TAUDTTOUTm (timer output) width = count clock cycle × TAUDnCDRm

23.12.6.3 Block Diagram and General Timing Diagram

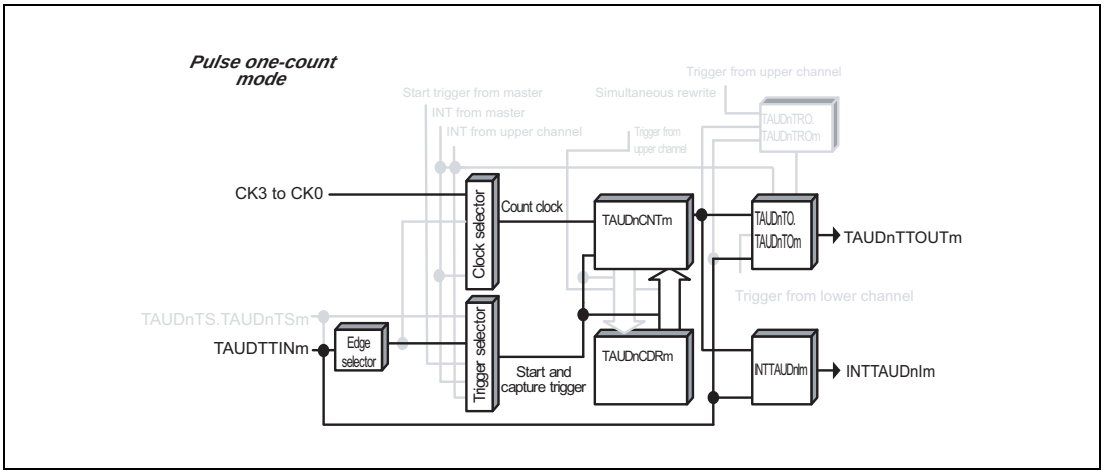


Figure 23.50 Block Diagram of One-Pulse Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

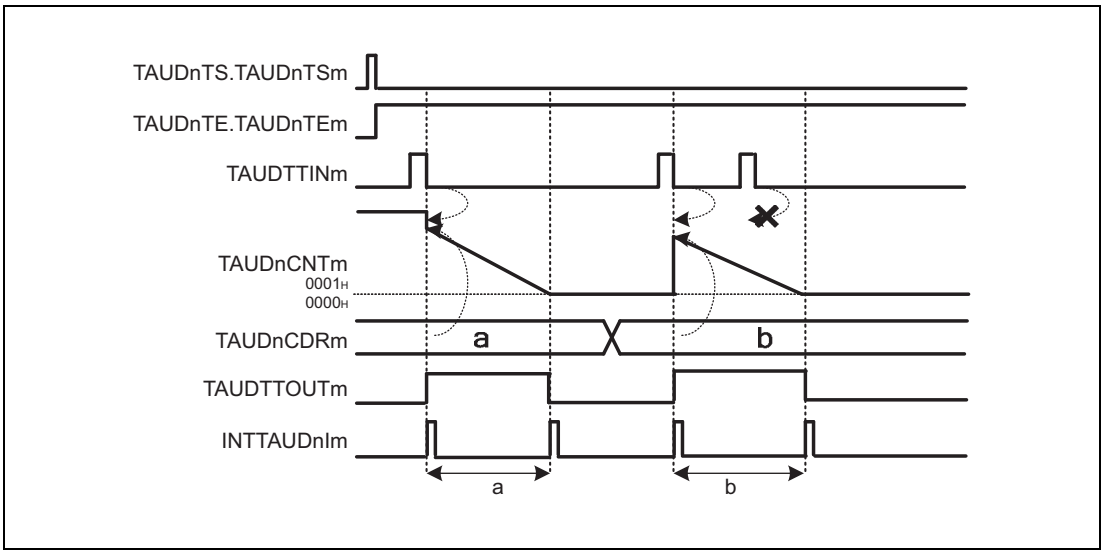


Figure 23.51 General Timing Diagram of One-Pulse Output Function

23.12.6.4 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 23.71 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 to this bit, 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 23.72 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 to these bits, 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 23.73 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: Set/reset mode
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 23.47, 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 23.74 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	

23.12.6.5 Operating Procedure for One-Pulse Output Function

Table 23.75 Operating Procedure for One-Pulse Output Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.71, Contents of the TAUDnCMORm Register for One-Pulse Output Function**, and **Table 23.72, 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 23.73, 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

23.12.7 TAUDTTINm Input Pulse Interval Measurement Function

23.12.7.1 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 signal.

Prerequisites

- The operating mode should be set to capture mode. See **Table 23.77, 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 overflow. 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 23.76 Effects of Overflow

	When Overflow Occurs		When a Valid TAUDTTINm Input is Detected	
TAUDnCMORm. TAUDnCOS[1:0]	**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 23.9, 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.

23.12.7.2 Equations

TAUDTTINm input pulse interval = count clock cycle × [(TAUDnCSRm.TAUDnOV F × ($FFFF_H$ + 1)) + TAUDnCDRm capture value + 1]

23.12.7.3 Block Diagram and General Timing Diagram

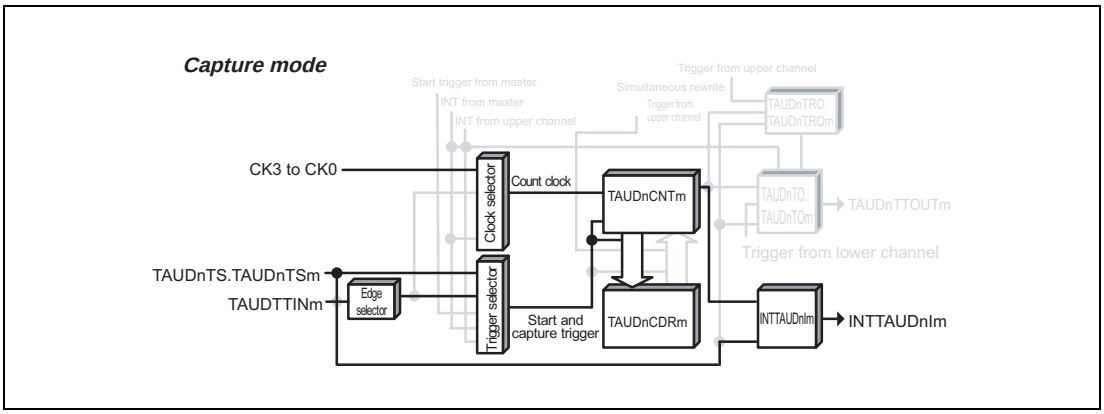


Figure 23.52 Block Diagram of TAUDTTINm Input Pulse Interval Measurement Function

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).

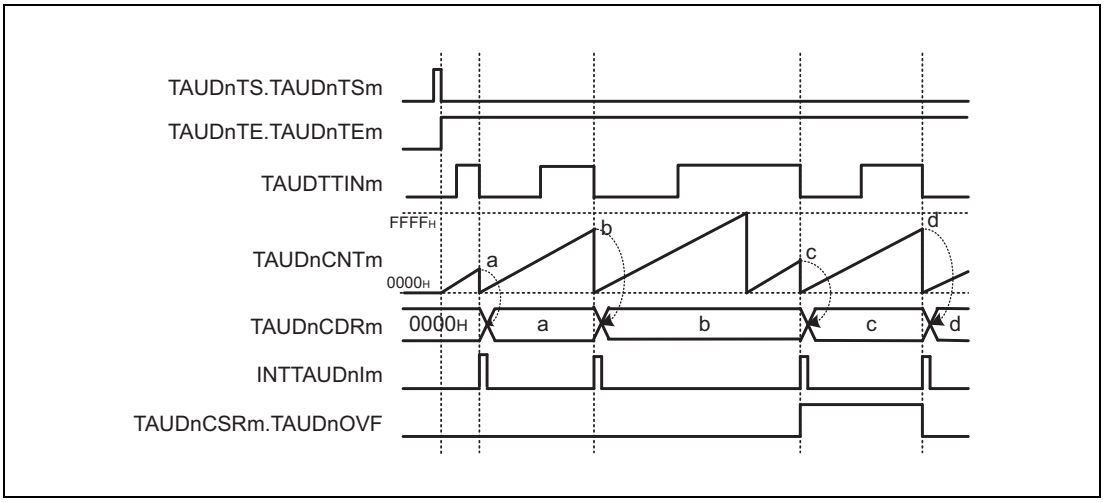


Figure 23.53 General Timing Diagram of TAUDTTINm Input Pulse Interval Measurement Function

23.12.7.4 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 23.77 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 the external capture trigger.
7, 6	TAUDnCOS[1:0]	See **Table 23.76, Effects of Overflow**.
5	Reserved	When read, the value after reset is returned. When writing to this bit, 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 23.78 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 to these bits, 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

(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 23.79 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	

23.12.7.5 Operating Procedure for TAUDTTINm Input Pulse Interval Measurement Function

Table 23.80 Operating Procedure for TAUDTTINm Input Pulse Interval Measurement Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.77, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Measurement Function**, and **Table 23.78, 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

23.12.7.6 Specific Timing Diagrams: Overflow Operation

(1) TAUDnCMORm.TAUDnCOS[1:0] = 00_B

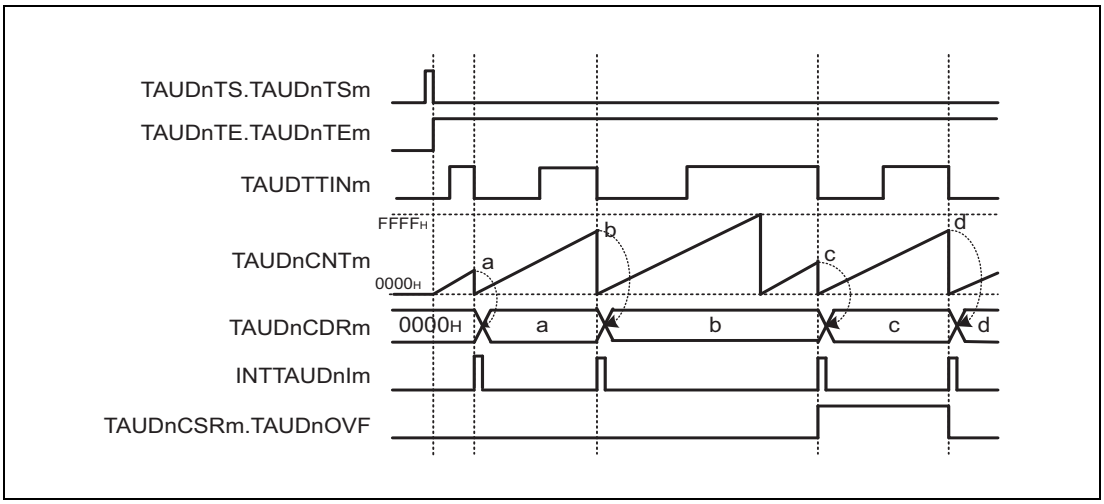


Figure 23.54 TAUDnCMORm.TAUDnCOS[1:0] = 00_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- 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 to1.
- 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

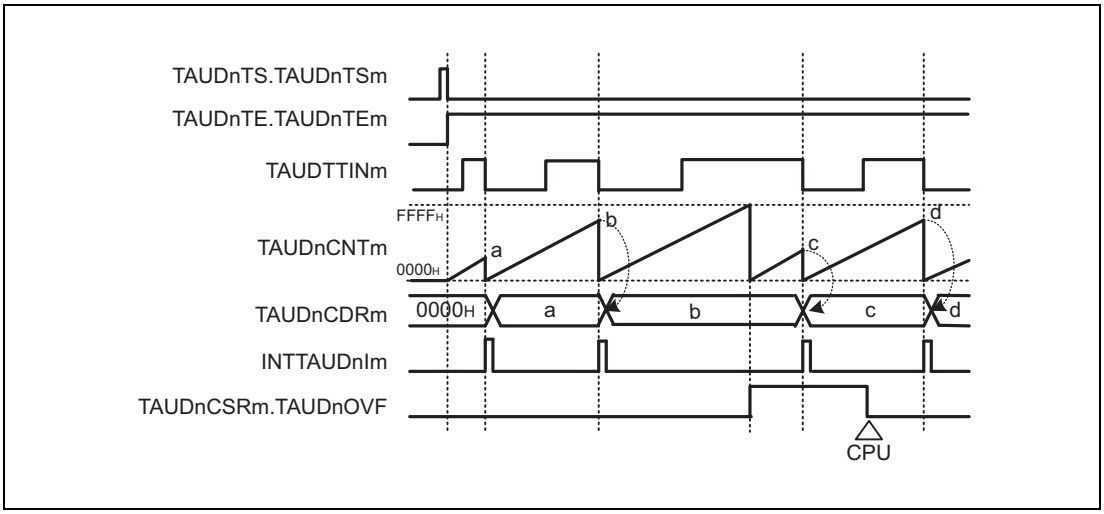


Figure 23.55 TAUDnCMORm.TAUDnCOS[1:0] = 01_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUDnCDRm remains unchanged and TAUDnCSRm.TAUDnOVF is set to1.

- 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

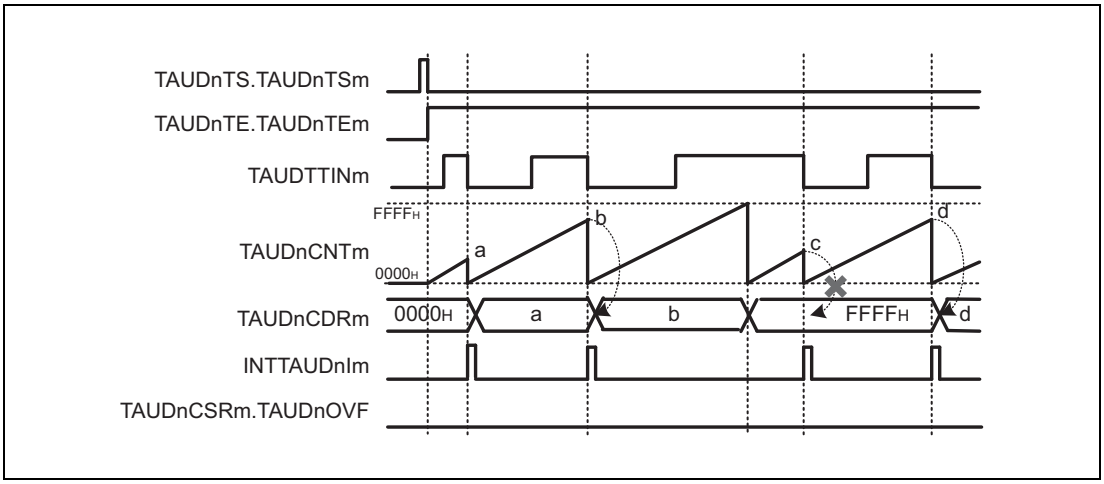


Figure 23.56 TAUDnCMORm.TAUDnCOS[1:0] = 10_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- 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

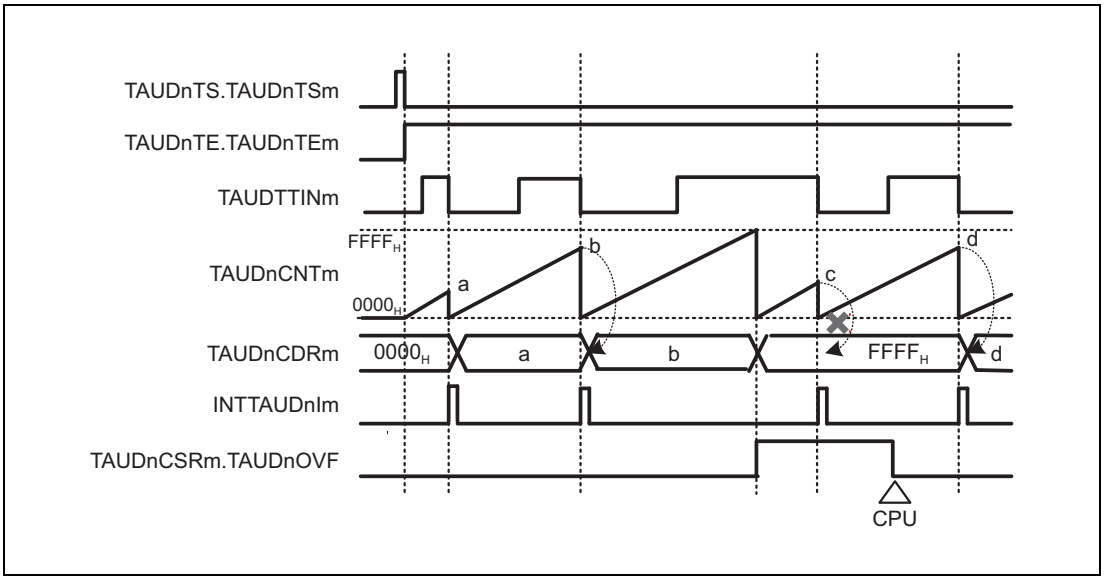


Figure 23.57 TAUDnCMORm.TAUDnCOS[1:0] = 11_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B

- 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.

23.12.8 TAUDTTINm Input Signal Width Measurement Function

23.12.8.1 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 23.82, 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 23.81 Effects of Overflow

	When Overflow Occurs		When a Valid TAUDTTINm Input Stop Edge is Detected	
TAUDnCMORm.TAUDnCOS[1:0]	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.COS[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.

23.12.8.2 Equations

TAUDTTINm input signal width = count clock cycle × [(TAUDnCSRm.TAUDnOVF × ($FFFF_H$ + 1)) + TAUDnCDRm capture value + 1]

23.12.8.3 Block Diagram and General Timing Diagram

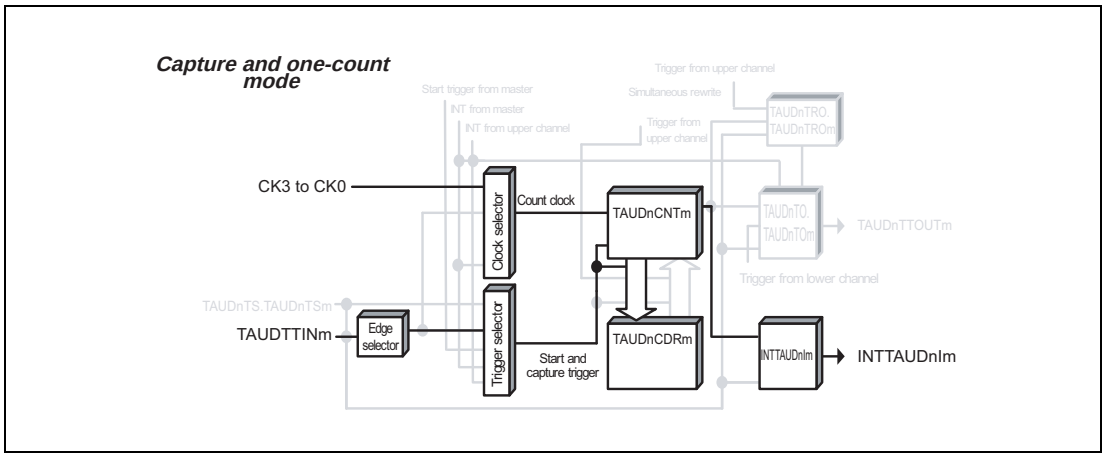


Figure 23.58 Block Diagram of TAUDTTINm Input Signal Width Measurement Function

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)

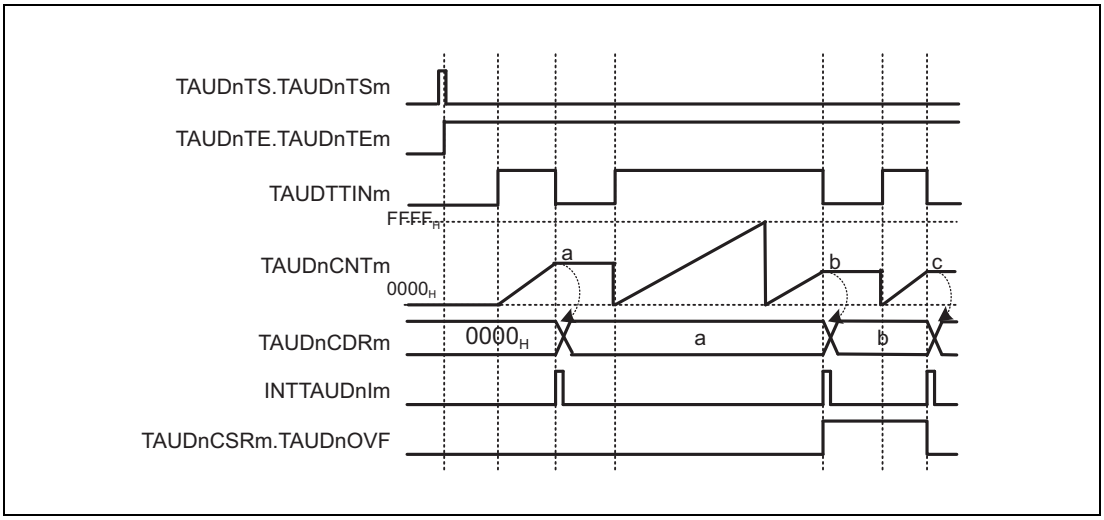


Figure 23.59 General Timing Diagram of TAUDTTINm Input Signal Width Measurement Function

23.12.8.4 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 23.82 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 23.81, Effects of Overflow**.
5	Reserved	When read, the value after reset is returned. When writing to this bit, 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 23.83 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 to these bits, 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 23.84 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	

23.12.8.5 Operating Procedure for TAUDTTINm Input Signal Width Measurement Function

Table 23.85 Operating Procedure for TAUDTTINm Input Signal Width Measurement Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.82, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Measurement Function**, and **Table 23.83, 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. TAUDnCSC.CLOV 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 TUAnCNTm 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

23.12.8.6 Specific Timing Diagrams: Overflow Operation

(1) TAUDnCMORm.TAUDnCOS[1:0] = 00_B

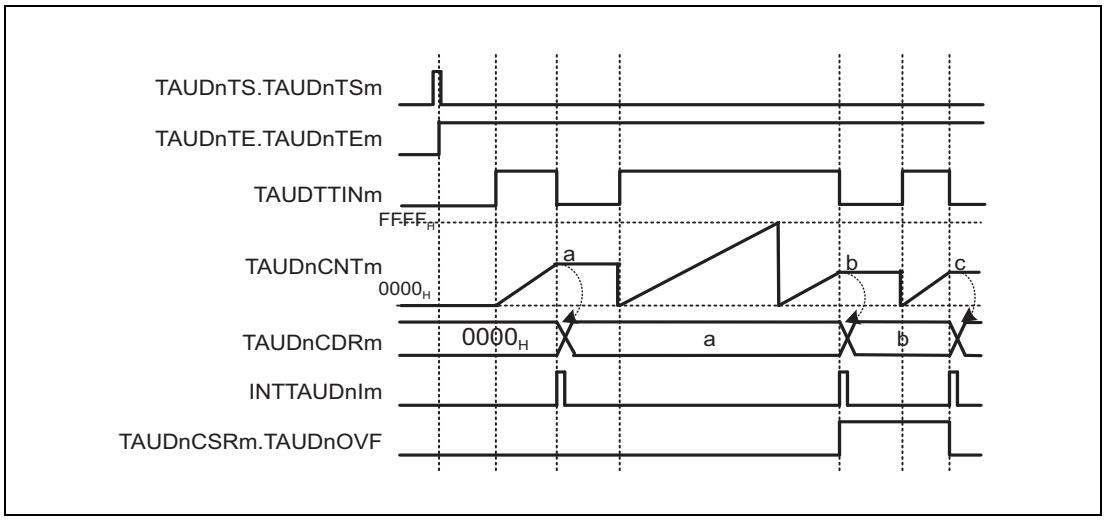


Figure 23.60 TAUDnCMORm.TAUDnCOS[1:0] = 00_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- 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

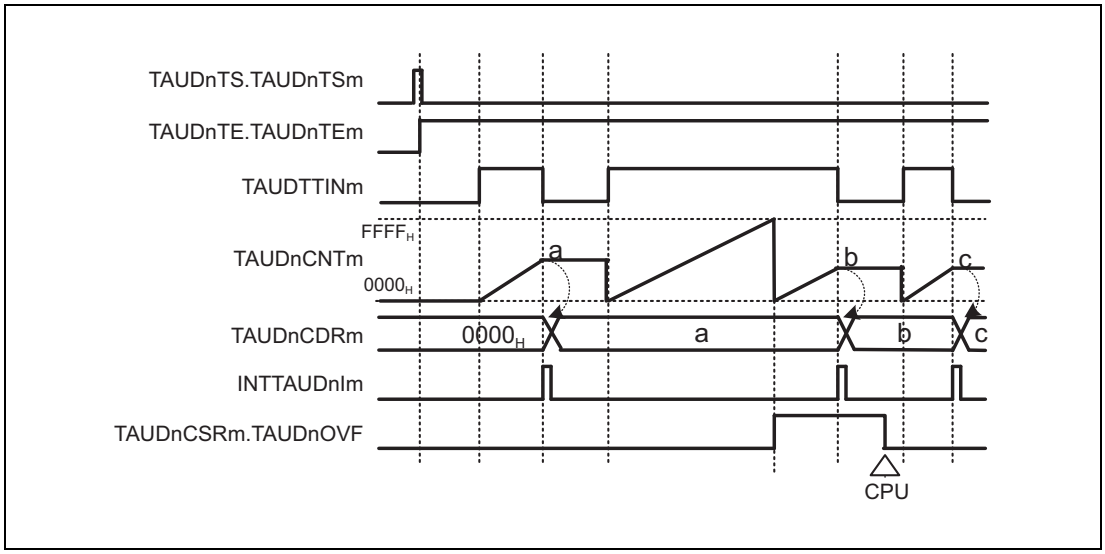


Figure 23.61 TAUDnCMORm.TAUDnCOS[1:0] = 01_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- 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

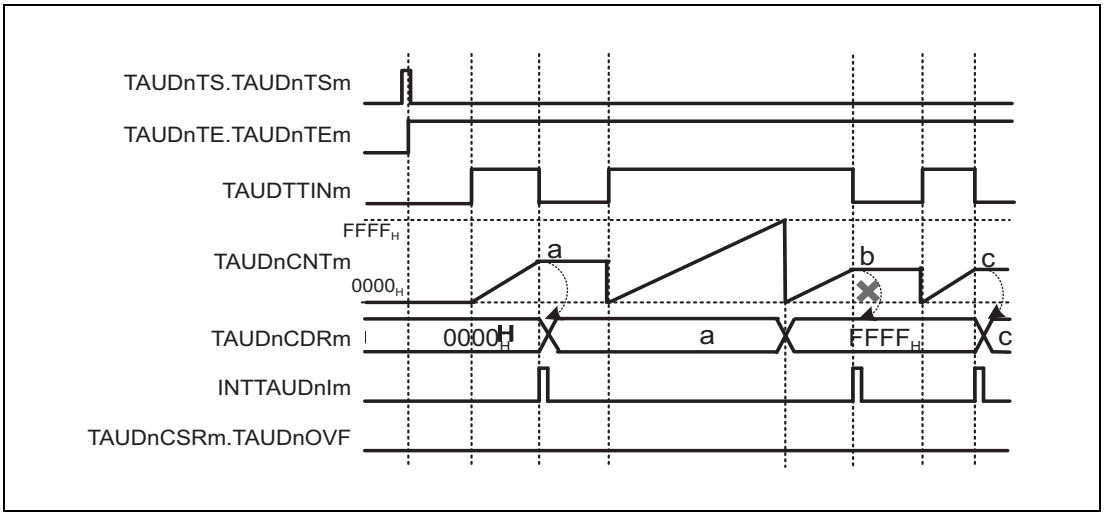


Figure 23.62 TAUDnCMORm.TAUDnCOS[1:0] = 10_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- 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

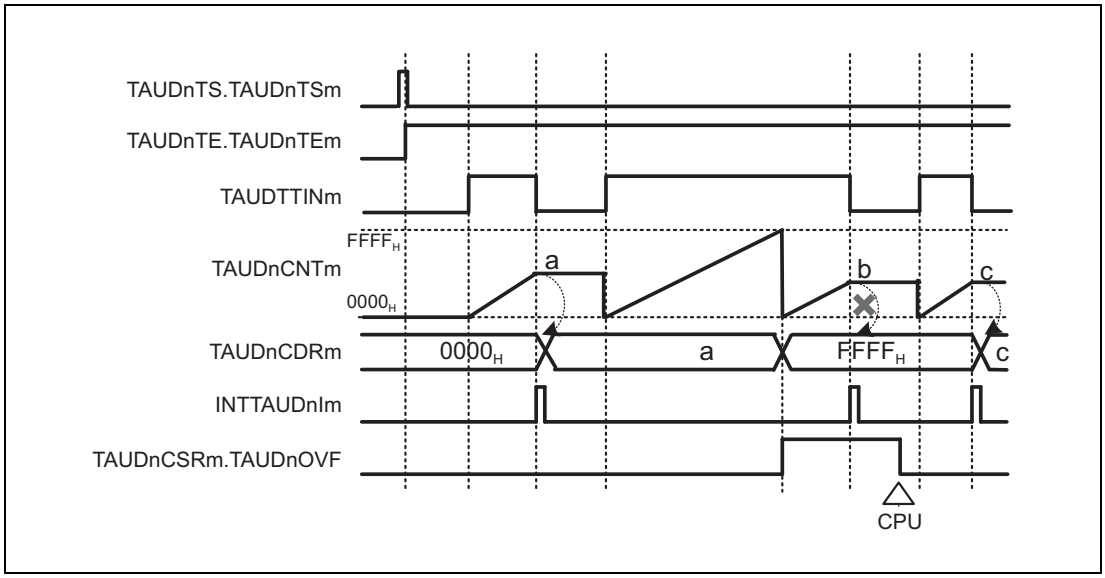


Figure 23.63 TAUDnCMORm.TAUDnCOS[1:0] = 11_B, TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 11_B

- 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.

23.12.9 TAUDTTINm Input Position Detection Function

23.12.9.1 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 23.86, 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 (INTTAUDnlm) is generated. The count operation continues.

When the counter reaches $FFFF_H$, the counter restarts from 0000_H.

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 23.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

23.12.9.2 Equations

Functional duration at a TAUDTTINm input pulse =

count clock cycle × (TAUDnCDRm capture value + 1)

23.12.9.3 Block Diagram and General Timing Diagram

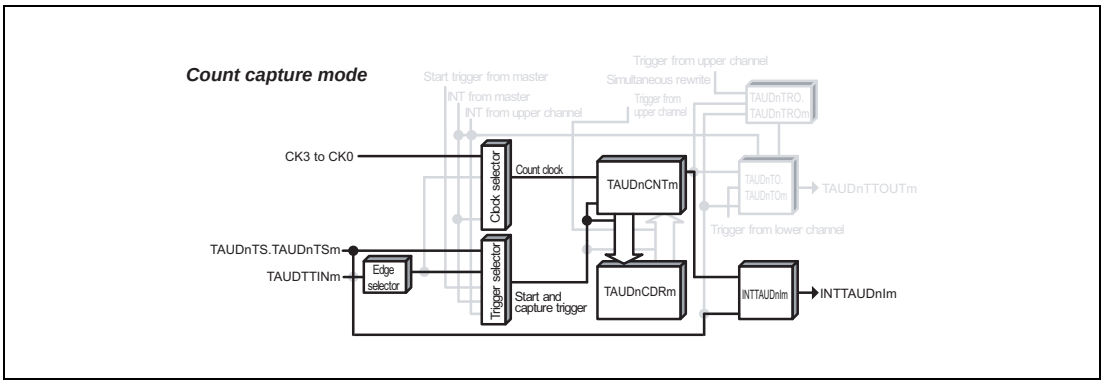


Figure 23.64 Block Diagram of TAUDTTINm Input Position Detection Function

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)

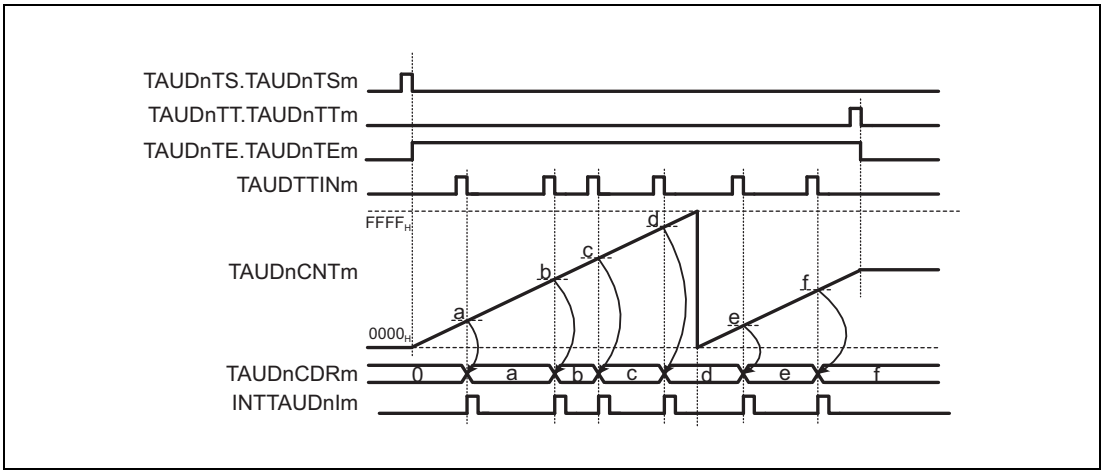


Figure 23.65 General Timing Diagram of TAUDTTINm Input Position Detection Function

23.12.9.4 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 23.86 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 to this bit, 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 23.87 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 to these bits, 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

(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 23.88 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	

23.12.9.5 Operating Procedure for TAUDTTINm Input Position Detection Function

Table 23.89 Operating Procedure for TAUDTTINm Input Position Detection Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.86, Contents of the TAUDnCMORm Register for TAUDTTINm Input Position Detection Function**, and **Table 23.87, 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.TIS[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. • INTTAUDnIm occurs. • 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

23.12.9.6 Specific Timing Diagrams

(1) Operation stop and restart

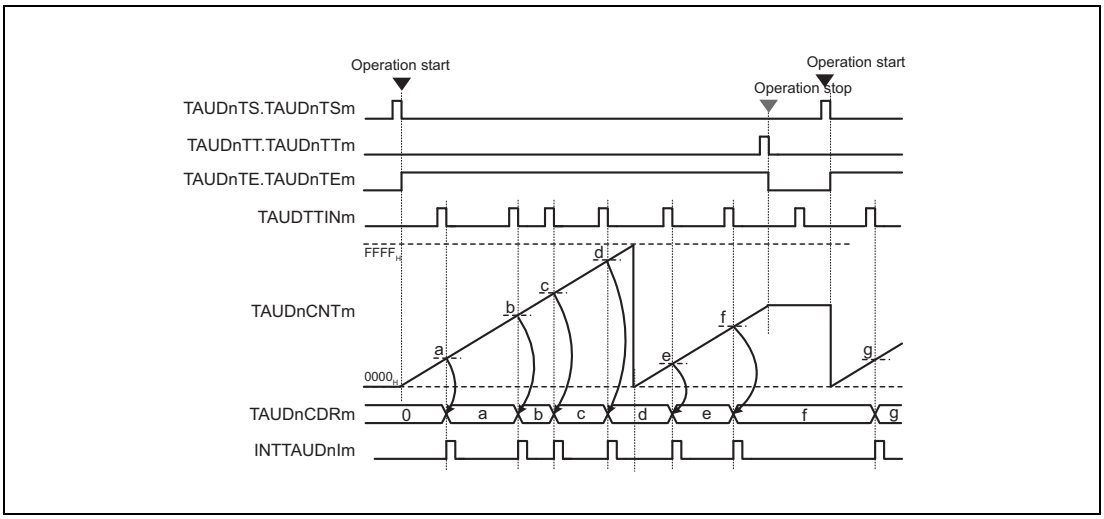


Figure 23.66 Operation Stop and Restart (TAUDnCMORm.TAUDnMD0 = 0, TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- 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.

23.12.10 TAUDTTINm Input Period Count Detection Function

23.12.10.1 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 23.90, 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.

23.12.10.2 Equations

Cumulative TAUDTTINm input width =
count clock cycle × (TAUDnCDRm capture value + 1)

23.12.10.3 Block Diagram and General Timing Diagram

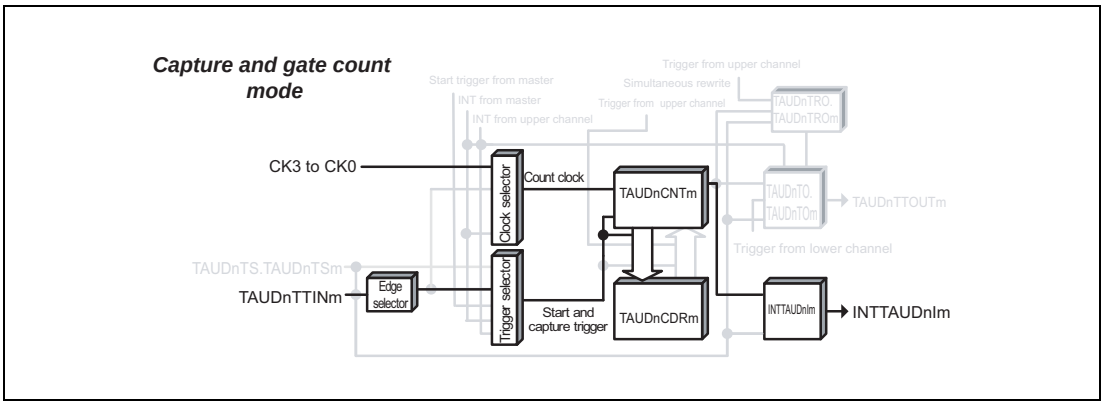


Figure 23.67 Block Diagram of TAUDTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

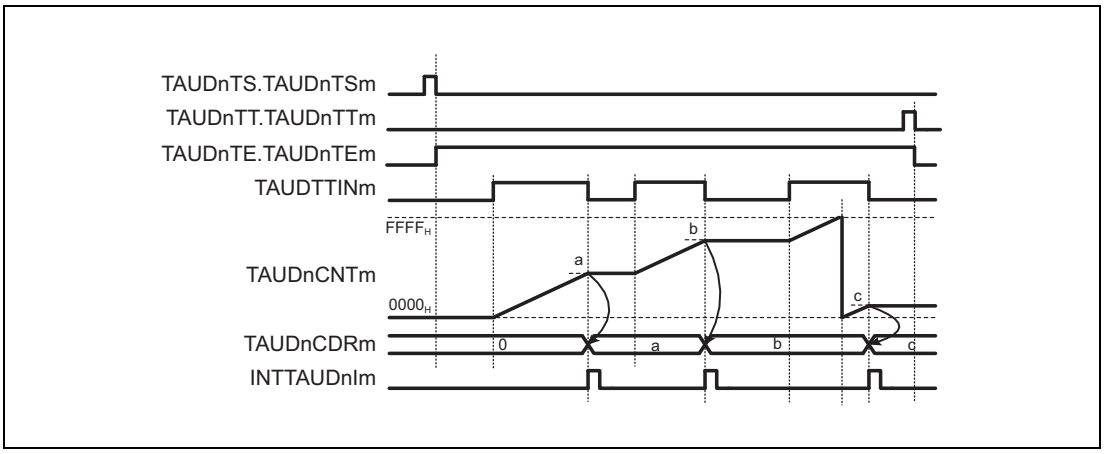


Figure 23.68 General Timing Diagram of TAUDTTINm Input Period Count Detection Function

23.12.10.4 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 23.90 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 to this bit, 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 23.91 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 to these bits, 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 23.92 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	

23.12.10.5 Operating Procedure for TAUDTTINm Input Period Count Detection Function

Table 23.93 Operating Procedure for TAUDTTINm Input Period Count Detection Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.90, Contents of the TAUDnCMORm Register for TAUDTTINm Input Period Count Detection Function**, and **Table 23.91, 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

23.12.10.6 Specific Timing Diagrams

(1) Operation stop and restart

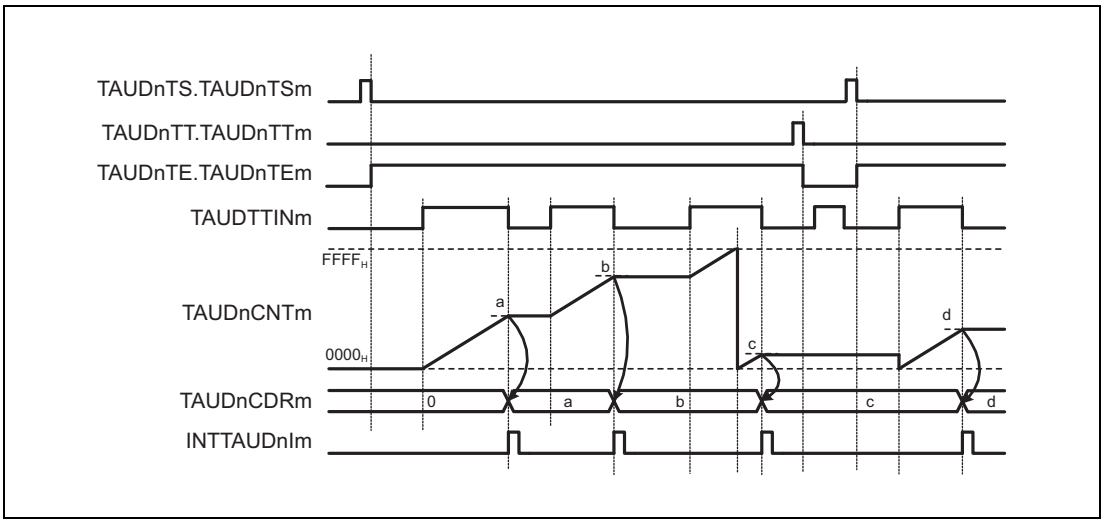


Figure 23.69 Operation Stop and Restart (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

- 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.

23.12.11 TAUDTTINm Input Pulse Interval Judgment Function

23.12.11.1 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 23.94, 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.

23.12.11.2 Block Diagram and General Timing Diagram

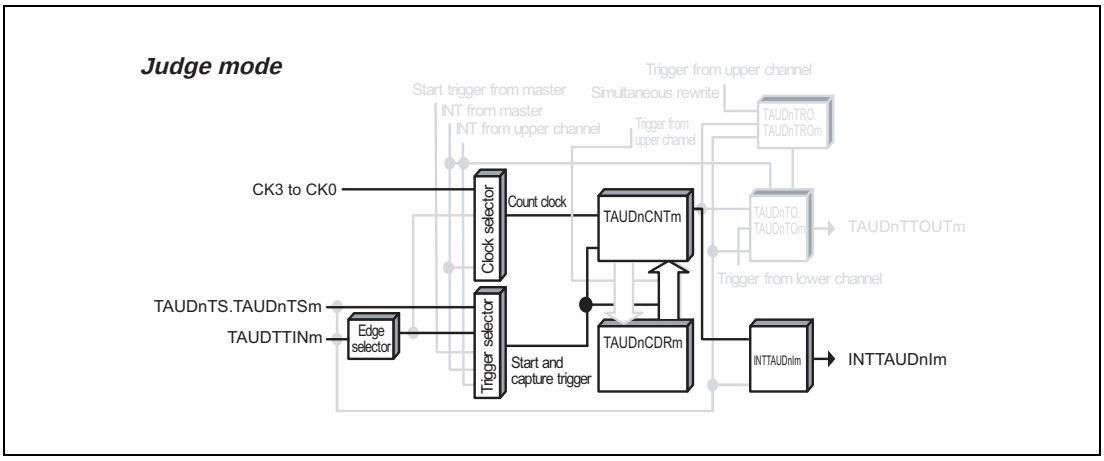


Figure 23.70 Block Diagram of TAUDTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

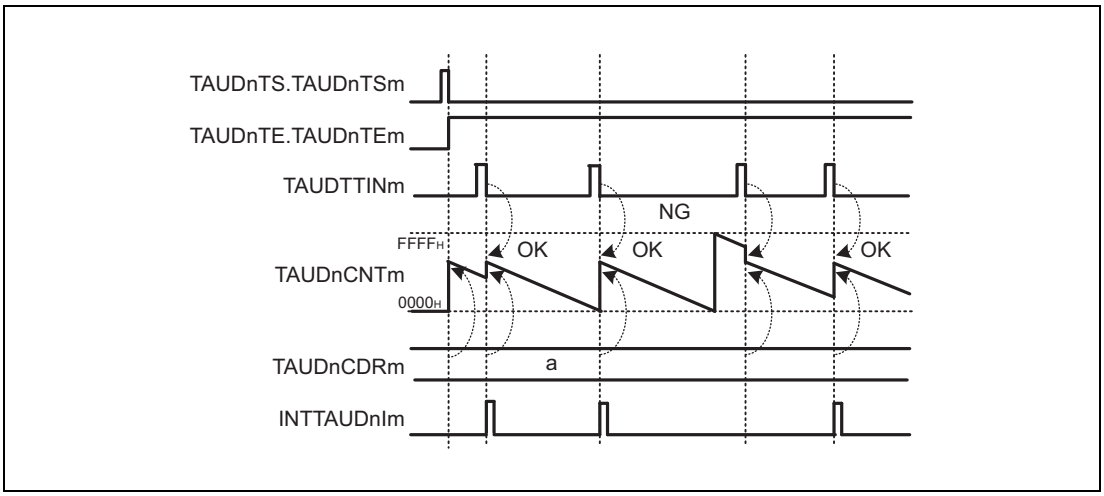


Figure 23.71 General Timing Diagram of TAUDTTINm Input Pulse Interval Judgment Function

23.12.11.3 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 23.94 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 to this bit, 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 23.95 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 to these bits, 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 23.96 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	

23.12.11.4 Operating Procedure for TAUDTTINm Input Pulse Interval Judgment Function

Table 23.97 Operating Procedure for TAUDTTINm Input Pulse Interval Judgment Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.94, Contents of the TAUDnCMORm Register for TAUDTTINm Input Pulse Interval Judgment Function**, and **Table 23.95, 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

23.12.12 TAUDTTINm Input Signal Width Judgment Function

23.12.12.1 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 23.98, 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.

23.12.12.2 Block Diagram and General Timing Diagram

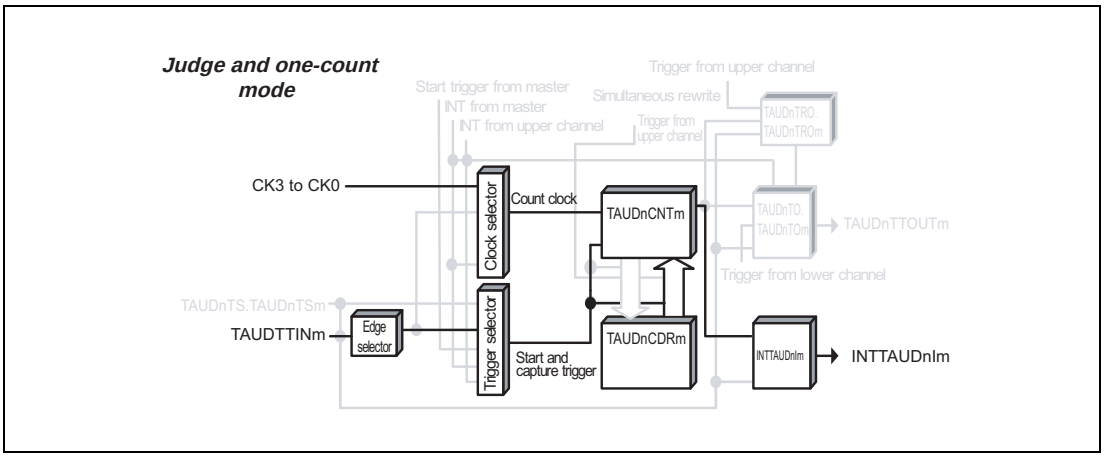


Figure 23.72 Block Diagram of TAUDTTINm Input Signal Width Judgment Function

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)

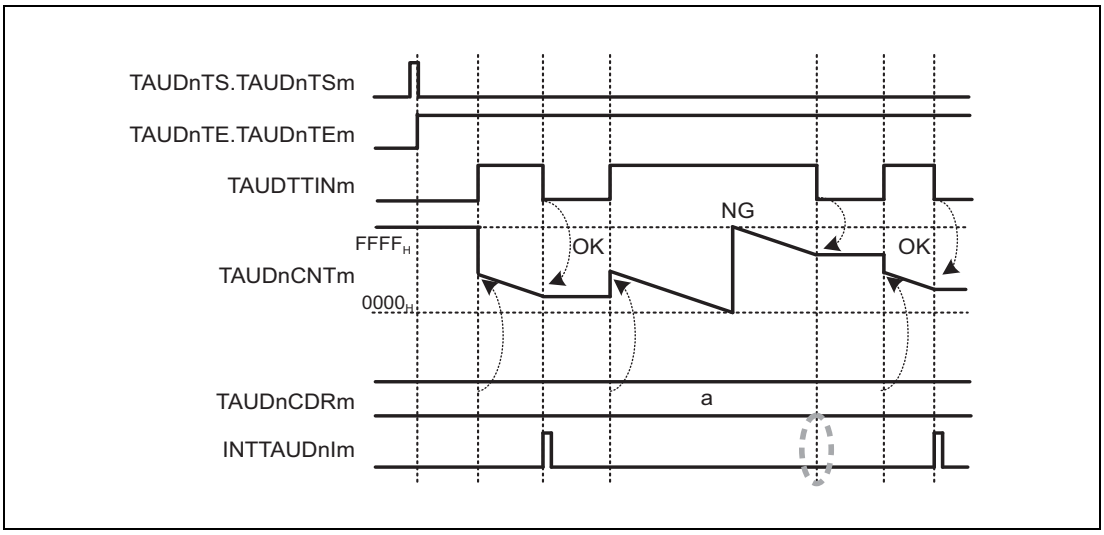


Figure 23.73 General Timing Diagram of TAUDTTINm Input Signal Width Judgment Function

23.12.12.3 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 23.98 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 to this bit, 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 23.99 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 to these bits, 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 23.100 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	

23.12.12.4 Operating Procedure for TAUDTTINm Input Signal Width Judgment Function

Table 23.101 Operating Procedure for TAUDTTINm Input Signal Width Judgment Function

	Operation	TAUDn Status
Initial Channel Setting	Set TAUDnCMORm and TAUDnCMURm registers as described in **Table 23.98, Contents of the TAUDnCMORm Register for TAUDTTINm Input Signal Width Judgment Function**, and **Table 23.99, 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

23.12.13 Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

23.12.13.1 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 23.102, 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 written 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 reloads $FFFF_H$ 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.

23.12.13.2 Block Diagram and General Timing Diagram

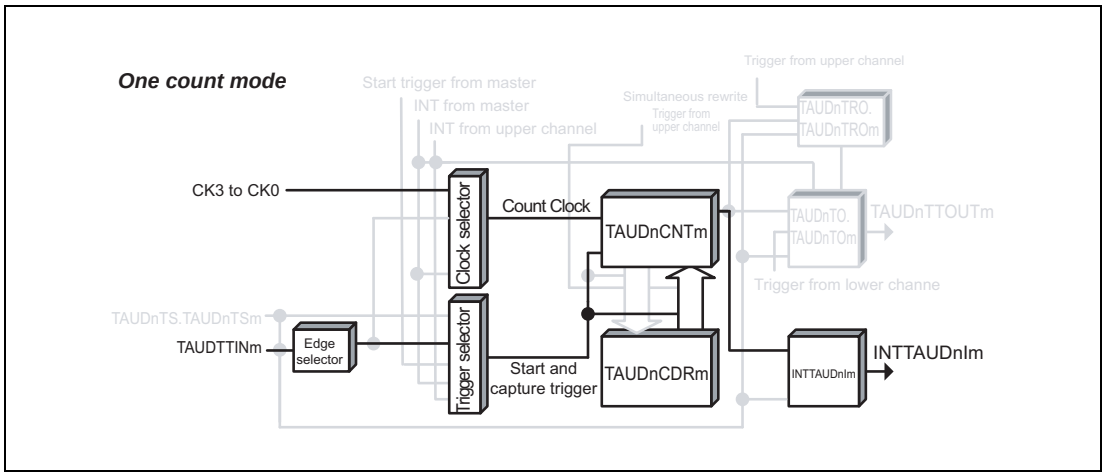


Figure 23.74 Block Diagram for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

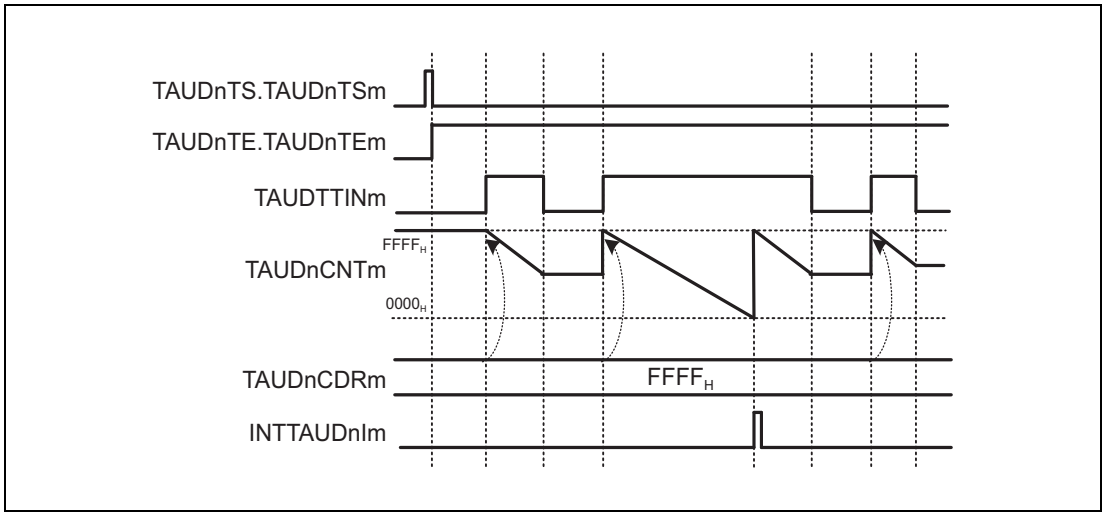


Figure 23.75 General Timing Diagram for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

23.12.13.3 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 23.102 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 to this bit, 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 23.103 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 to these bits, 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 23.104 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	

23.12.13.4 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)

Table 23.105 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 23.102, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Width Measurement)**, and **Table 23.103, 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 during count operation: • 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

23.12.14 Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

23.12.14.1 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 23.106, 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 written 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 written 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.

23.12.14.2 Block Diagram and General Timing Diagram

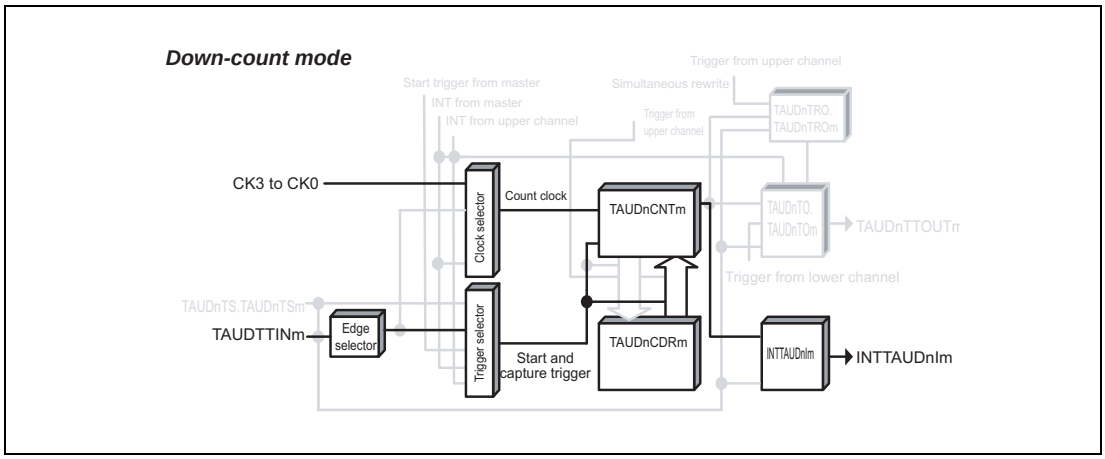


Figure 23.76 Block Diagram for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Detection of rising and falling edges = high width measurement (TAUDnCMURm.TAUDnTIS[1:0] = 11_B)

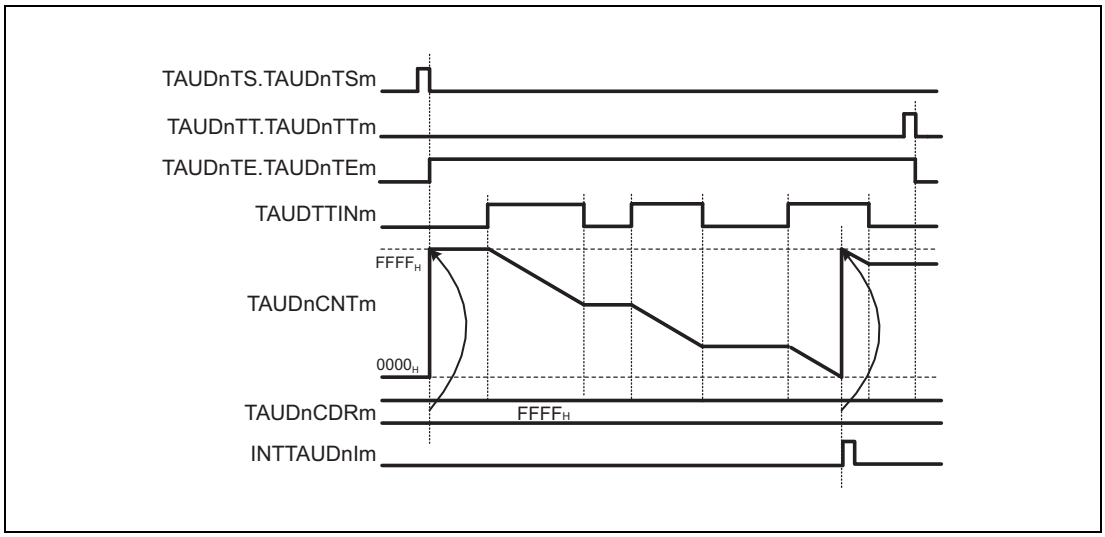


Figure 23.77 General Timing Diagram for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

23.12.14.3 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 23.106 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 to this bit, 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 23.107 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 to these bits, 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 23.108 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	

23.12.14.4 Operating Procedure for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)

Table 23.109 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 23.106, Contents of the TAUDnCMORm Register for Overflow Interrupt Output Function (during TAUDTTINm Input Period Count Detection)**, and **Table 23.107, 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. • TAUDnCDRm 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 stop value. When TAUDTTINm input start edge is detected during count operation: • 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

23.12.15 One-Phase PWM Output Function

23.12.15.1 Overview

Summary

This function adds dead time to a TAUDTTINm input signal. The resulting PWM signal is output via TAUDnTTOUTm 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 23.111, 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 23.7, 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 23.110 TAUDTTOUTm to which Dead Time is Added and State of TAUDTTINm

TAUDnCMUR.TAUDnTISm	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 TAUDnTTOUTm 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.

23.12.15.2 Block Diagram and General Timing Diagram

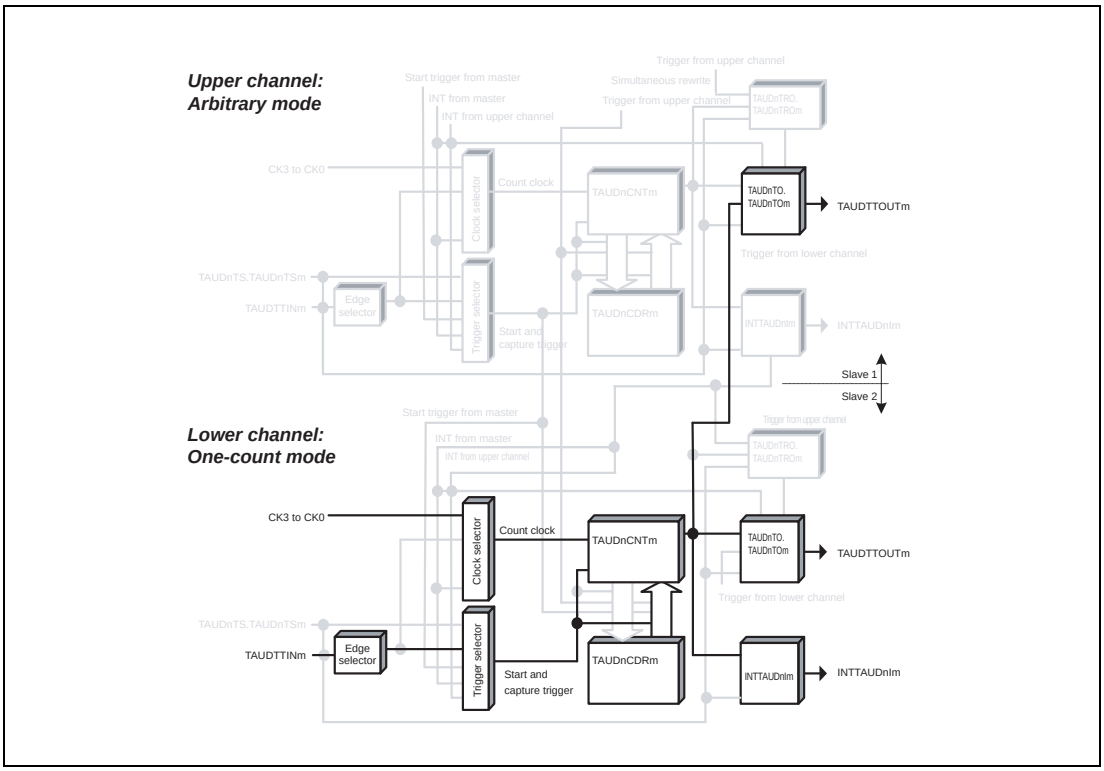


Figure 23.78 Block Diagram of One-Phase PWM Output Function

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 a high active.

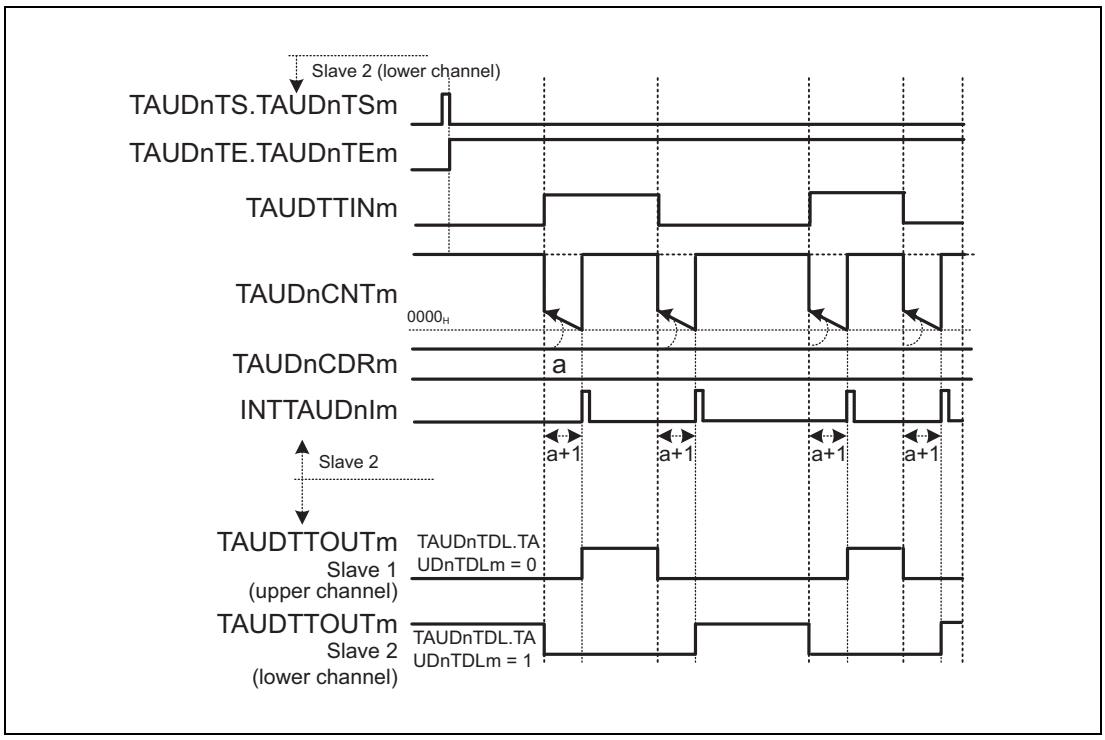


Figure 23.79 General Timing Diagram of One-Phase PWM Output Function

23.12.15.3 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 23.111 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 to this bit, 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 23.112 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 to these bits, 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 23.113 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: Uses an interrupt as a TAUDTTOUTm set trigger and a valid TAUDTTINm edge as a TAUDTTOUTm reset trigger. 1: Uses a valid TAUDTTINm edge as a TAUDTTOUTm set trigger and an interrupt as a TAUDTTOUTm reset trigger.
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 23.114 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	

23.12.15.4 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 23.115 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.

23.12.15.5 Operating Procedure for One-phase PWM Output Function

Table 23.116 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 23.111, Contents of the TAUDnCMORm Register for the Lower Channel of the One-Phase PWM Output Function**, and **Table 23.112, 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 23.12.15.4, Register Settings for Upper Channels**. Set the value of TAUDnCDRm register. Set channel output mode by setting the control bits as described in **Table 23.113, 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

23.13 Independent Channel Real-Time Functions

This section describes functions that output the value of the TAUDnTRO.TAUDnTROm bit in real time.

23.13.1 Real-Time Output Function Type 1

23.13.1.1 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 23.117, 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 23.7, 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, 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.
- 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 23.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

23.13.1.2 Equations

INTTAUDnIm generation cycle = count clock cycle × (TAUDnCDRm value + 1)

23.13.1.3 Block Diagram and General Timing Diagram

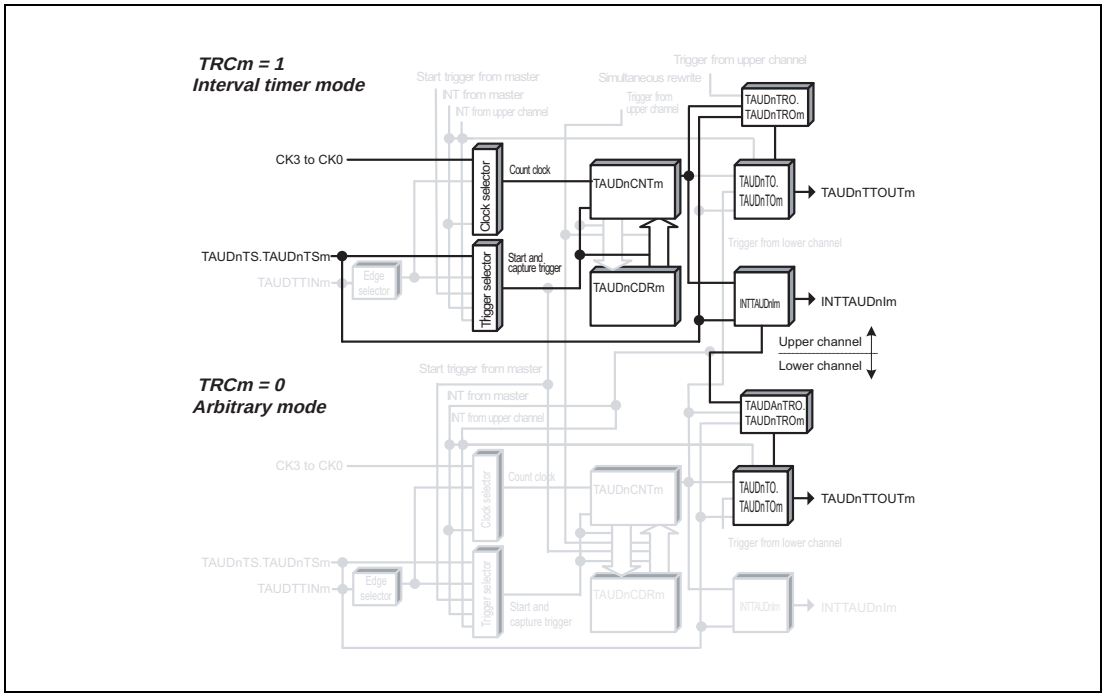


Figure 23.80 Block Diagram of Real-Time Output Function Type 1

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

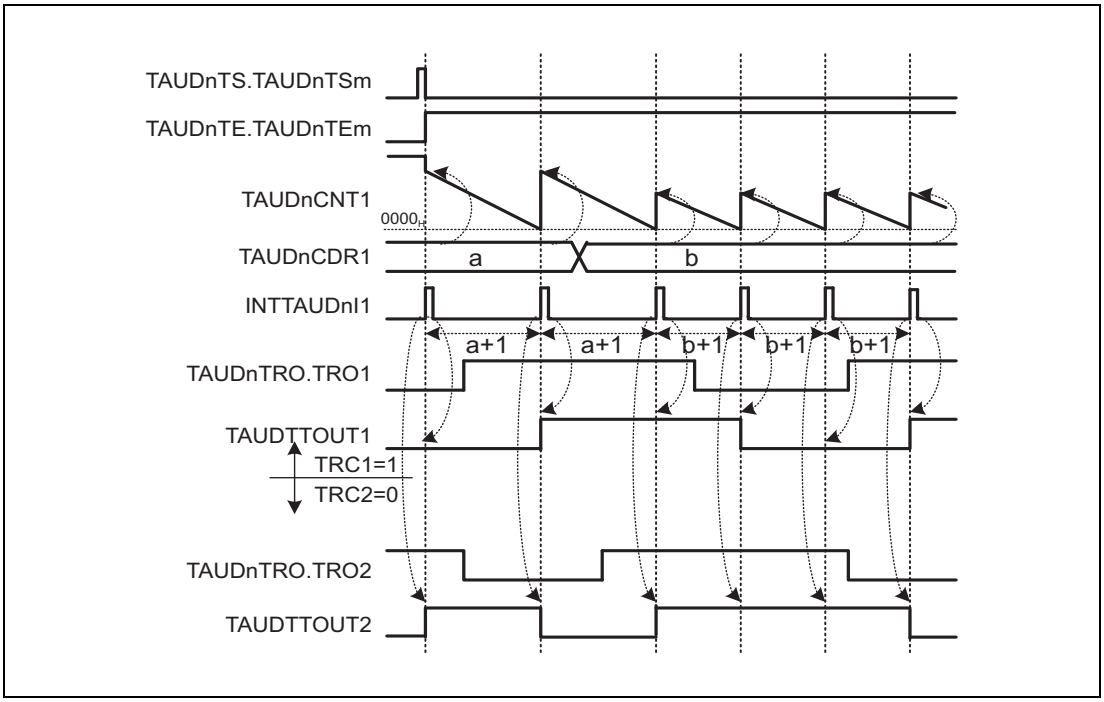


Figure 23.81 General Timing Diagram of Real-Time Output Function Type 1

23.13.1.4 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 23.117 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	TAUDnSTS[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 to this bit, 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 23.118 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for upper channels

Table 23.119 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 23.120 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	

23.13.1.5 Register Settings for Lower Channels

(1) TAUDnCMORm for lower channels

The TAUDnCMORm register for lower channels is available for any setting.

(2) TAUDnCMURm for lower channels

The TAUDnCMURm register for lower channels is available for any setting.

(3) Channel output mode for lower channels

Table 23.121 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 is available for any setting.

23.13.1.6 Operating Procedure for Real-Time Output Function Type 1

Table 23.122 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 23.117, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 1**, and **Table 23.118, 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 23.13.1.5, 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 23.119, 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 23.121, 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. TAUDnCDRm value is loaded 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

23.13.1.7 Specific Timing Diagrams

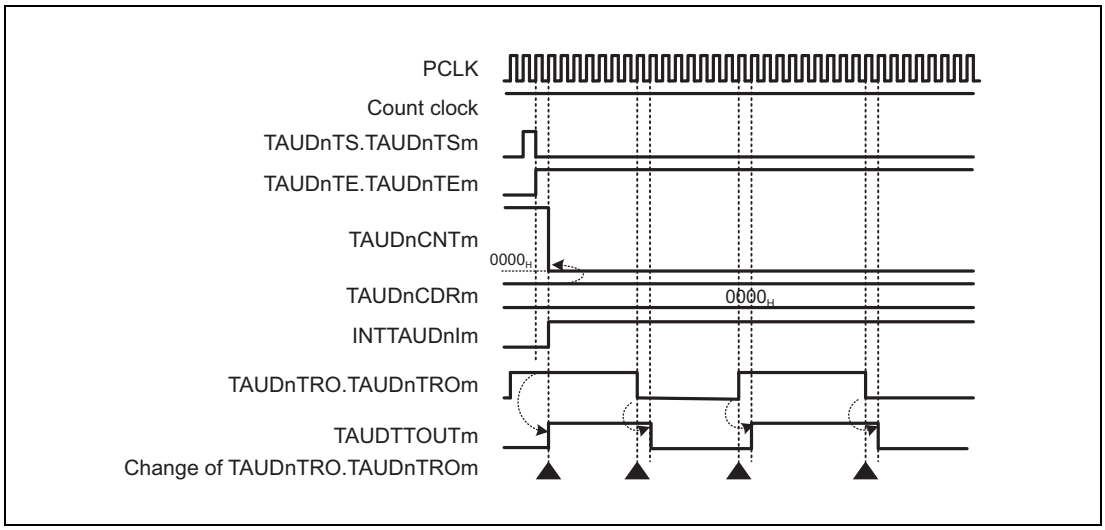


Figure 23.82 TAUDnCDRm = 0000_H, TAUDnCMORm.TAUDnMD0 = 1

- The value of TAUDTTOUTm changes according to the setting of TAUDnTRO.TAUDnTROm with a delay of one PCLK cycle.

23.13.2 Real-Time Output Function Type 2

23.13.2.1 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 interval timer mode. (See **Table 23.123, 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 23.7, 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.
- 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 23.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

23.13.2.2 Block Diagram and General Timing Diagram

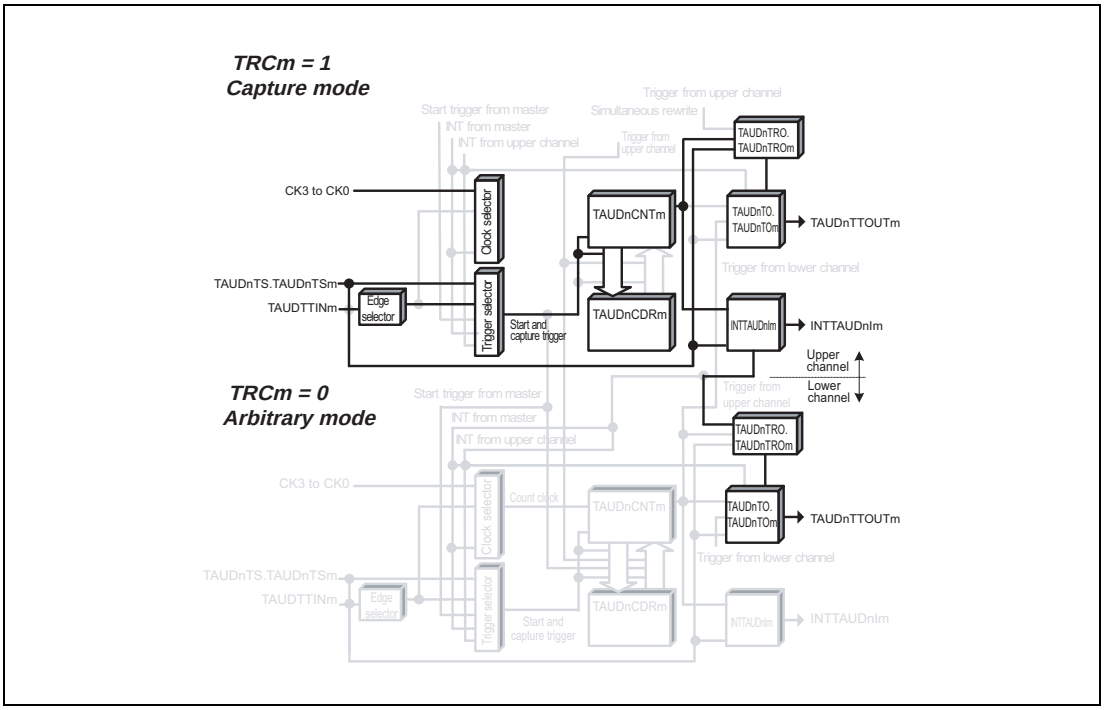


Figure 23.83 Block Diagram of Real-Time Output Function Type 2

The following settings apply to the general timing diagram.

- INTTAUDnIm is not generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 0)

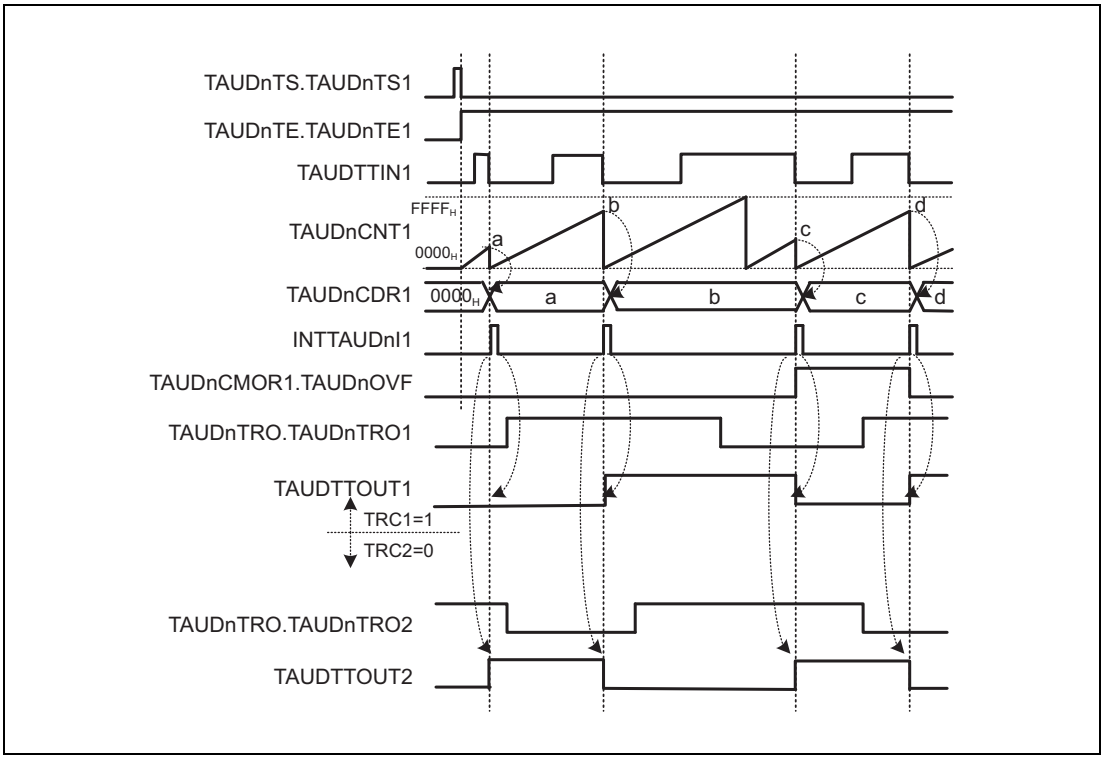


Figure 23.84 General Timing Diagram of Real-Time Output Function Type 2

23.13.2.3 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 23.123 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 to this bit, 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 23.124 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 to these bits, 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 23.125 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 23.126 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	

23.13.2.4 Register Settings for Lower Channels

(1) TAUDnCMORm for lower channels

The TAUDnCMORm register for lower channels is available for any setting.

(2) TAUDnCMURm for lower channels

The TAUDnCMURm register for lower channels is available for any setting.

(3) Channel output mode for lower channels

Table 23.127 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.

23.13.2.5 Operating Procedure for Real-Time Output Function Type 2

Table 23.128 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 23.123, Contents of the TAUDnCMORm Register for the Upper Channel of Real-Time Output Function Type 2**, and **Table 23.124, 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 23.13.2.4, 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 23.125, 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 23.127, 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

23.13.2.6 Specific Timing Diagrams

(1) Operation start and stop

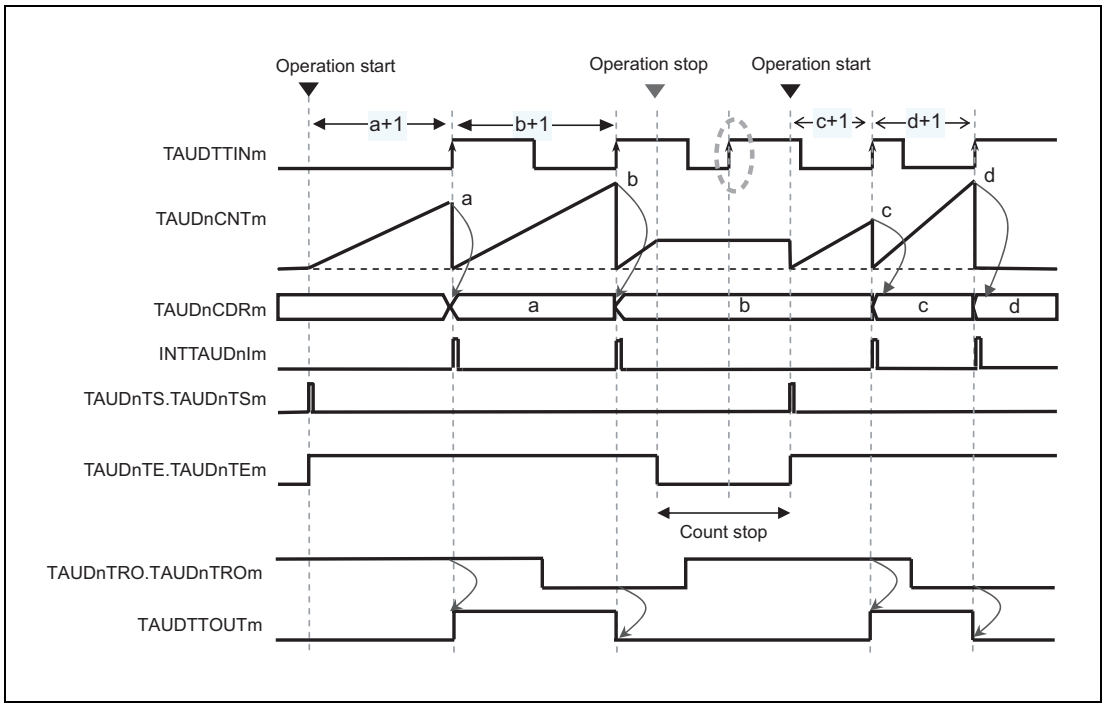


Figure 23.85 Operation Start and Stop (TAUDnCMORm.TAUDnMD0 = 0)

- 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.

23.14 Independent Channel Simultaneous Rewrite Functions

This section describes functions that carry out simultaneous rewrite.

23.14.1 Simultaneous Rewrite Trigger Generation Function Type 1

23.14.1.1 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 23.129, 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 23.46, Channel Operation Functions and Methods They Use**.
- 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 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 23.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

23.14.1.2 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 TAUDnCDRm_master + 1 should be an integer. This integer corresponds to the number of interrupts.

For triangle PWM, remember that the cycle doubles.

23.14.1.3 Block Diagram and General Timing Diagram

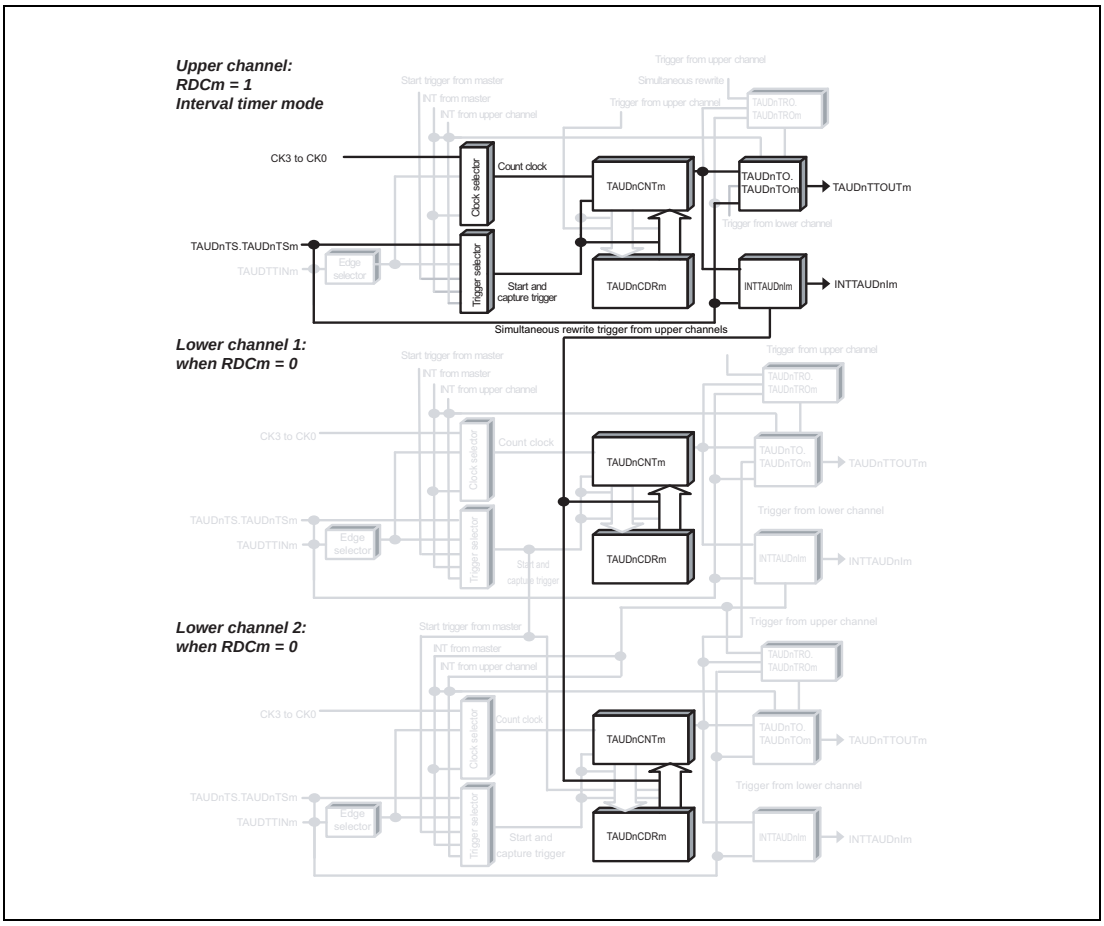


Figure 23.86 Block Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

The following settings apply to the general timing diagram.

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

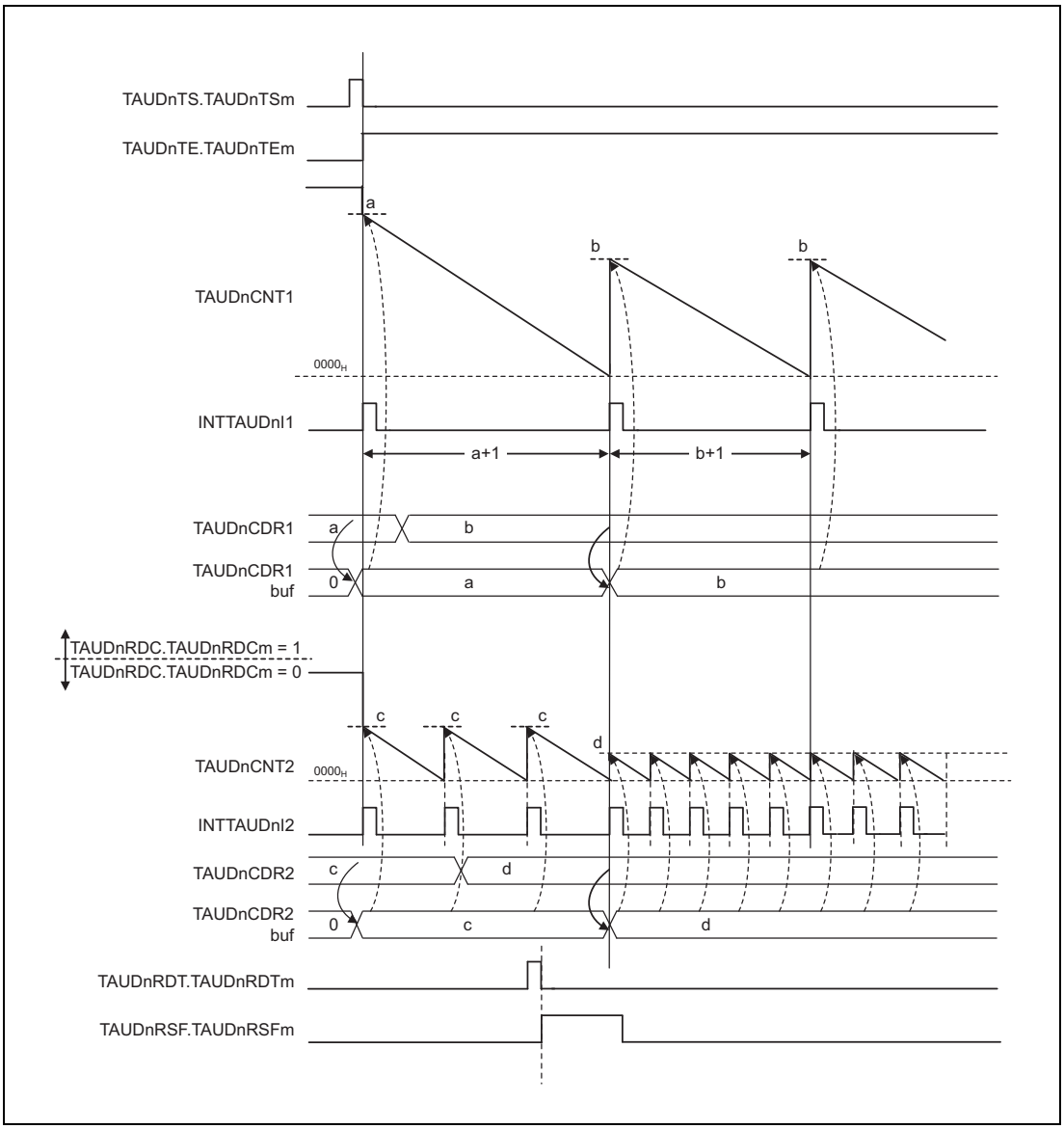


Figure 23.87 General Timing Diagram of Simultaneous Rewrite Trigger Generation Function Type 1

23.14.1.4 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 23.129 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 to this bit, 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 23.130 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 to these bits, 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 23.131 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: Follows the TAUDnRDM.TAUDnRDMm bit settings in the operating mode which can be set.
TAUDnRDC.TAUDnRDCm	1: Monitors INTTAUDnIm signal which triggers a simultaneous rewrite on the channel.

23.14.1.5 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 23.46, Channel Operation Functions and Methods They Use**).

(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 23.46, Channel Operation Functions and Methods They Use**)

(3) Channel output mode for lower channels

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 23.45, Simultaneous Rewrite Methods and when They are Triggered**.

(4) Simultaneous rewrite for lower channels

Table 23.132 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	Follows the TAUDnRDM.TAUDnRDMm bit settings in the operating mode which can be set.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

23.14.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Table 23.133 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 23.129, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 1**, and **Table 23.130, 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 23.14.1.5, 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 is 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

23.14.2 Simultaneous Rewrite Trigger Generation Function Type 2

23.14.2.1 Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is triggered by a valid TAUDTTINm input edge or the function starting.

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 23.134, 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 23.46, Channel Operation Functions and Methods They Use**.

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 23.9, TAUDTTOUTm Output and INTTAUDnIm Generation when Counter Starts or Restarts**.

23.14.2.2 Block Diagram and General Timing Diagram

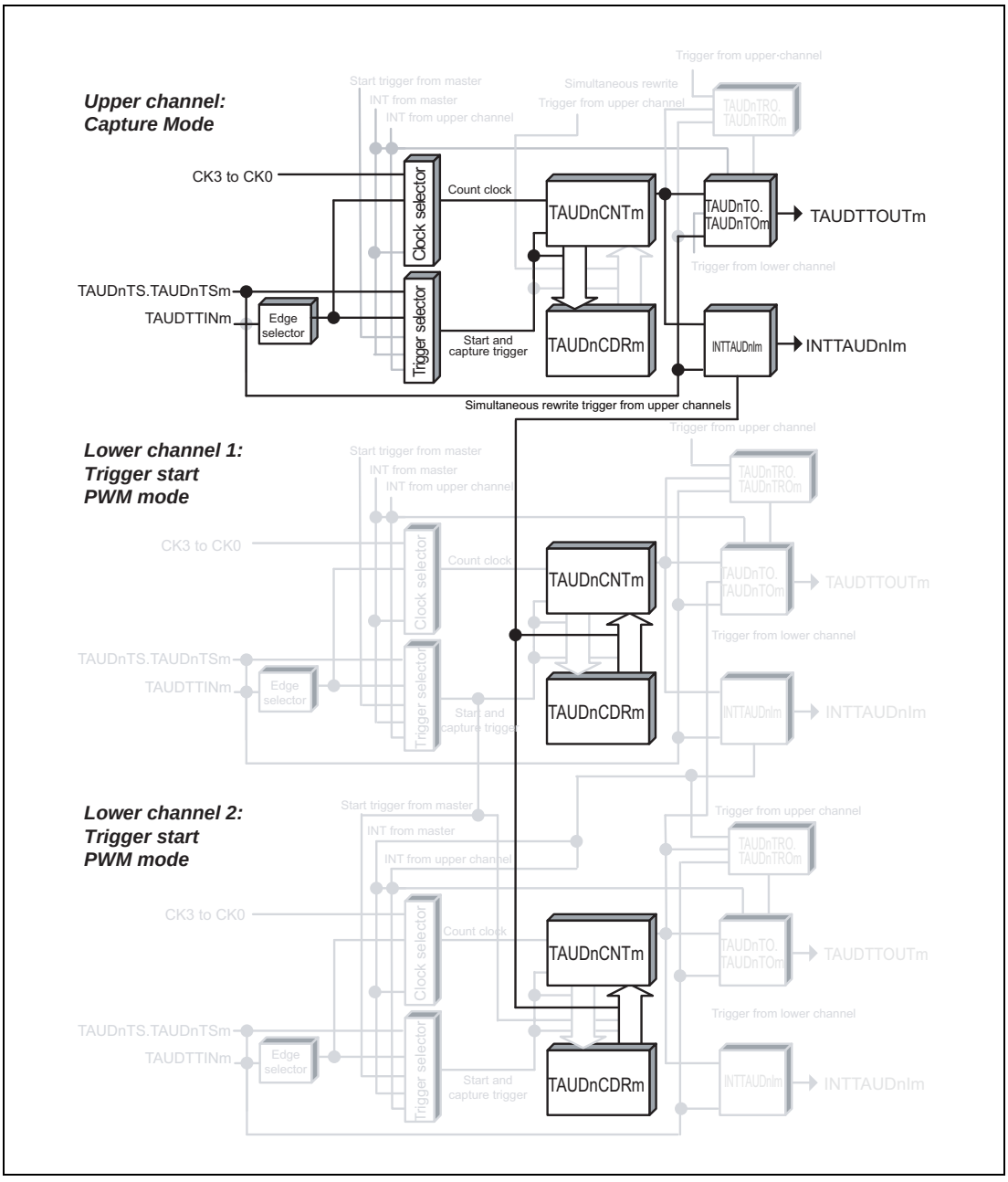


Figure 23.88 Block Diagram for Simultaneous Rewrite Trigger Generation Function Type 2

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 (channel 1) generates simultaneous rewrite trigger.

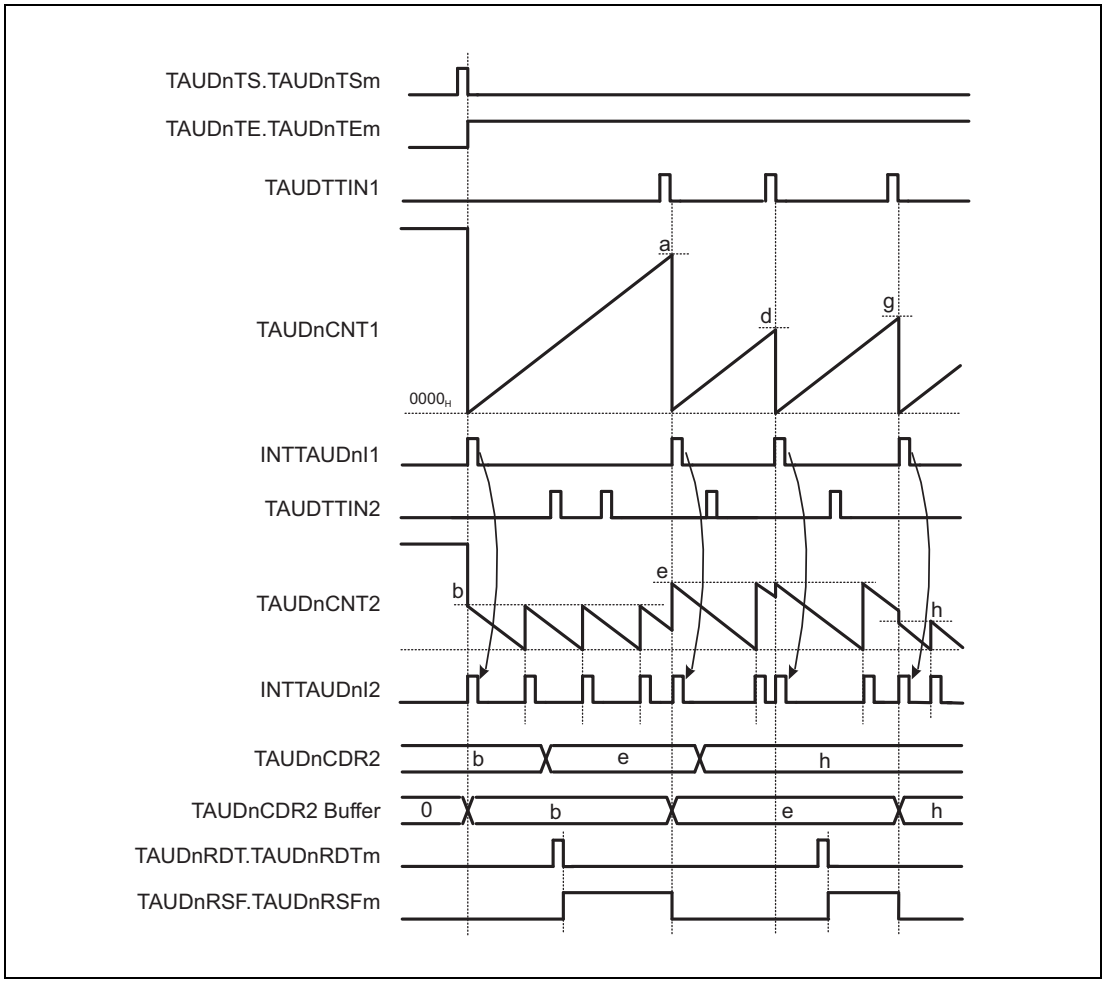


Figure 23.89 General Timing Diagram for Simultaneous Rewrite Trigger Generation Function Type 2

23.14.2.3 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]		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 23.134 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 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 to this bit, 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 23.135 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 to these bits, 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 23.136 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.

23.14.2.4 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 23.137 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 to this bit, 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 23.138 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 to these bits, 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 23.139 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.

23.14.2.5 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 2

Table 23.140 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 23.134, Contents of the TAUDnCMORm Register for the Upper Channel of Simultaneous Rewrite Trigger Generation Function Type 2** and **Table 23.135, 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 23.137, Contents of the TAUDnCMORm Register for the Lower channel of Simultaneous Rewrite Trigger Generation Function Type 2** and **Table 23.138, 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

23.15 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 23.2, Overview**

This section describes functions that generate PWM signals at regular intervals.

23.15.1 PWM Output Function

23.15.1.1 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 23.141, 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 23.144, 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 23.7, 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 INTTAUDnlm 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 INTTAUDnlm (start of the next pulse cycle) of the master channel.

The counter can stop operating 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 23.6, Simultaneous Rewrite**.

23.15.1.2 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

23.15.1.3 Block Diagram and General Timing Diagram

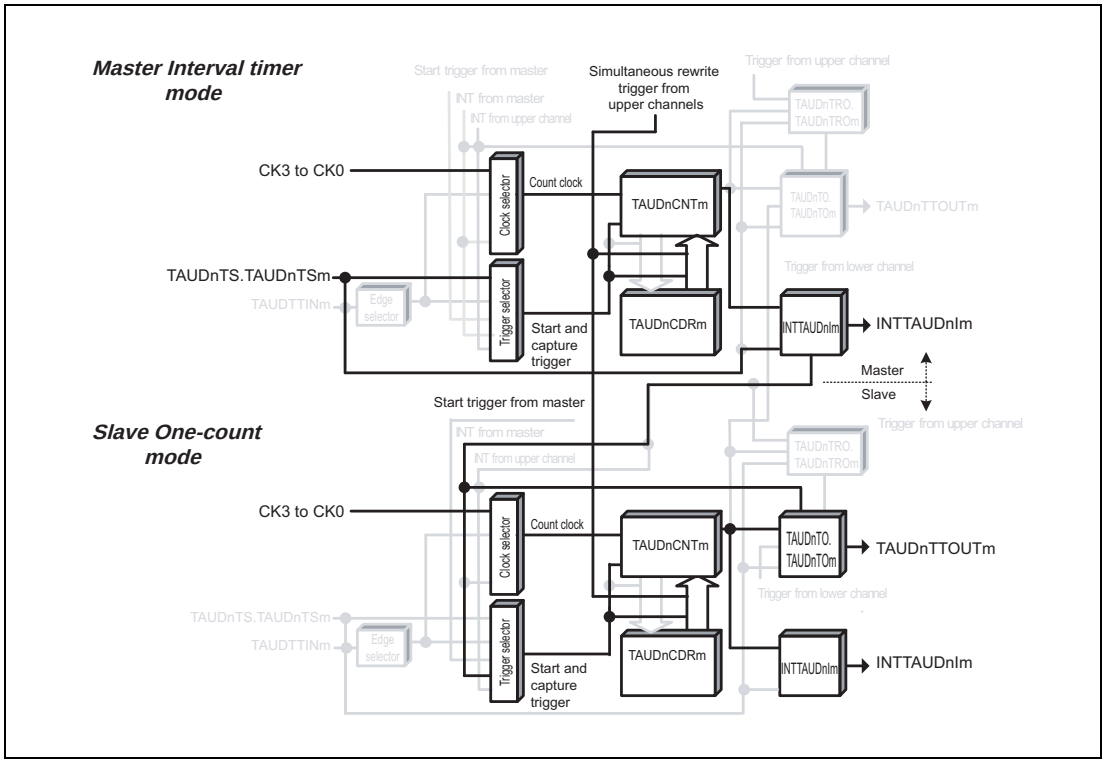


Figure 23.90 Block Diagram of PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

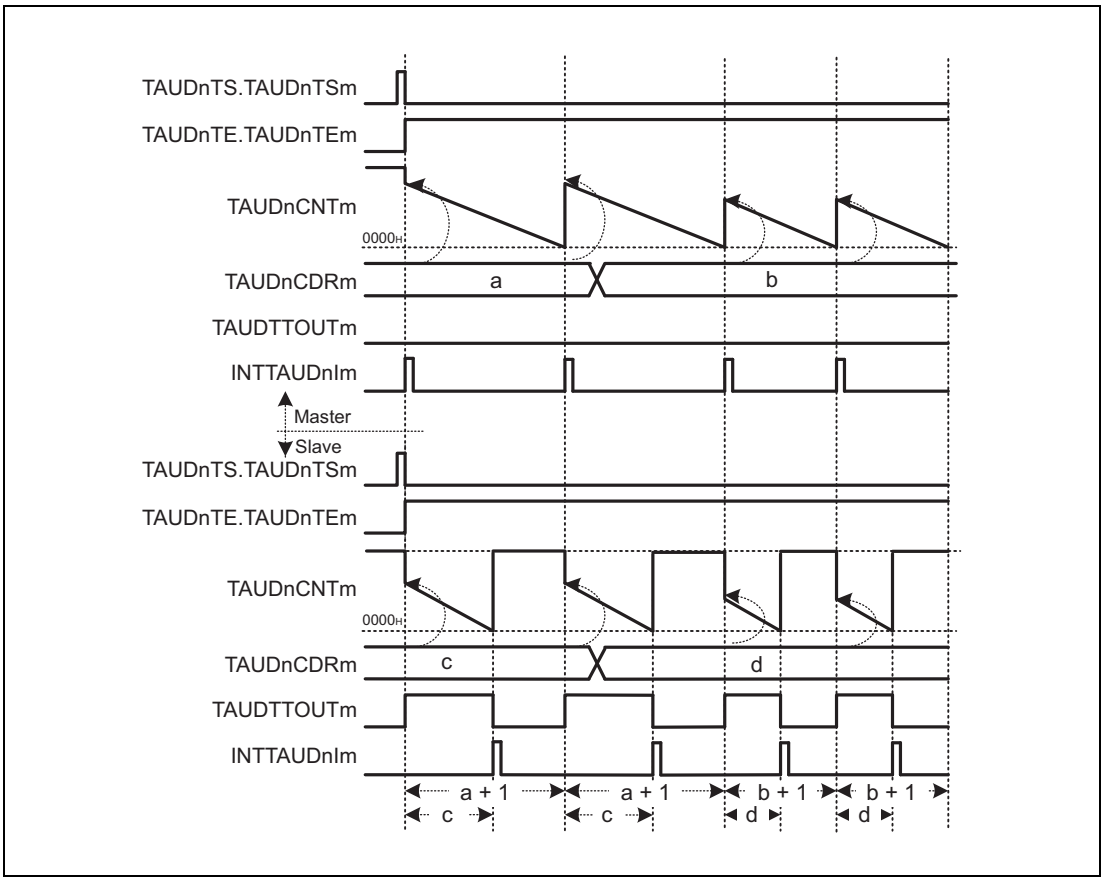


Figure 23.91 General Timing Diagram of PWM Output Function

NOTES

1. The interval between the slave channel starting to count and an interrupt being generated is the value of corresponding TAUDnCDRm, whereas for the master channel the interval is the value of the 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.

23.15.1.4 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]		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 23.141 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 to this bit, 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 23.142 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 to these bits, 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 23.143 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 an upper channel higher than the master channel that operates with **Section 23.14.1, 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.

23.15.1.5 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]		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 23.144 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 to this bit, 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 23.145 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channels

Table 23.146 Control Bit Settings in Independent 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 23.147 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.

23.15.1.6 Operating Procedure for PWM Output Function

Table 23.148 Operating Procedure for PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.1.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.1.5, 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

23.15.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

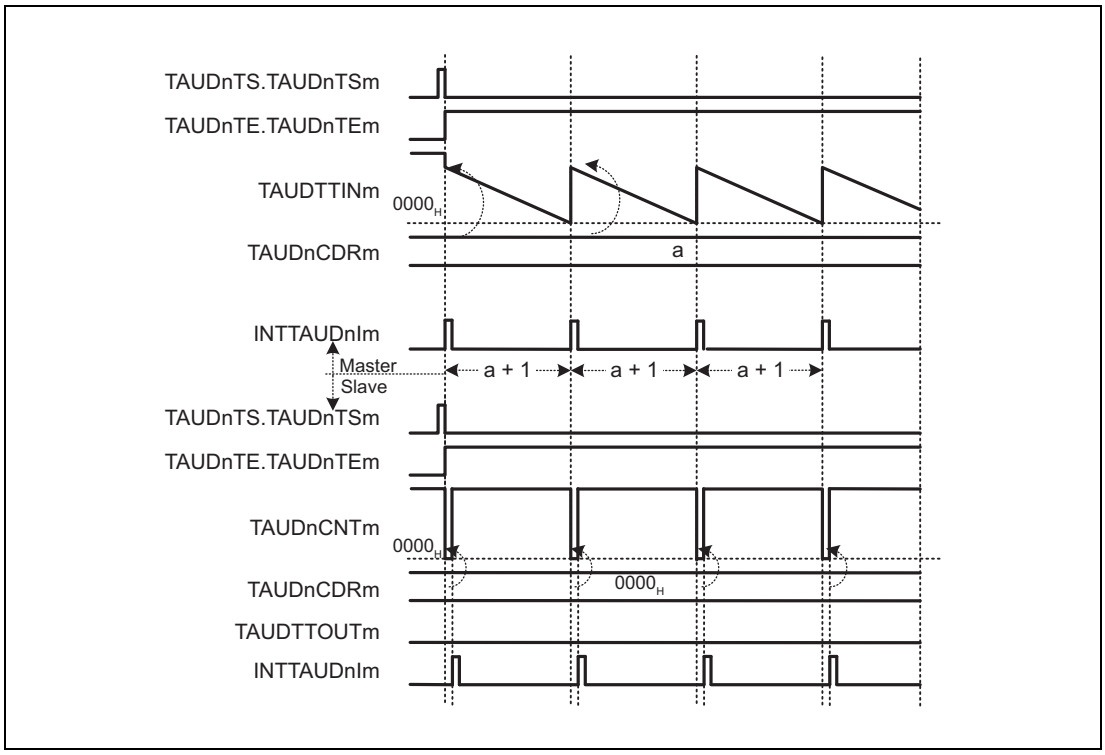


Figure 23.92 TAUDnCDRm (Slave) = 0000_H, Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- 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%

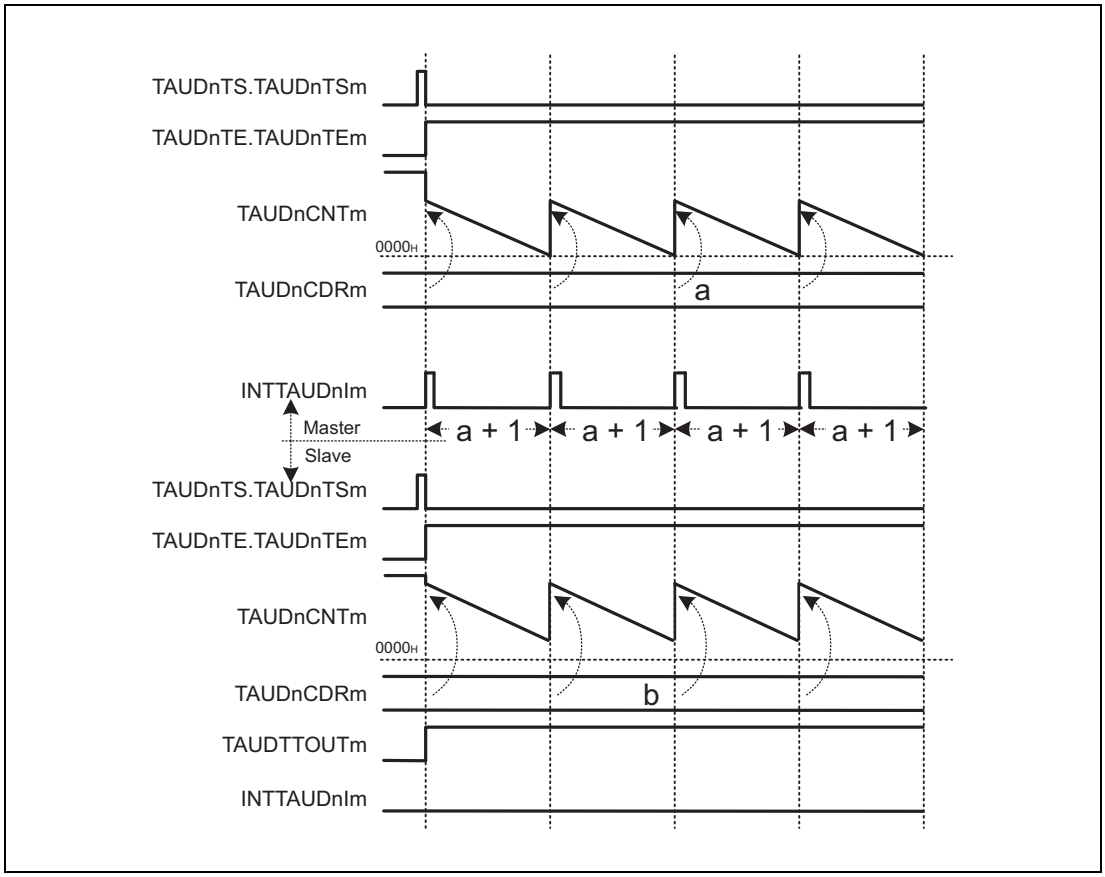


Figure 23.93 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- 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

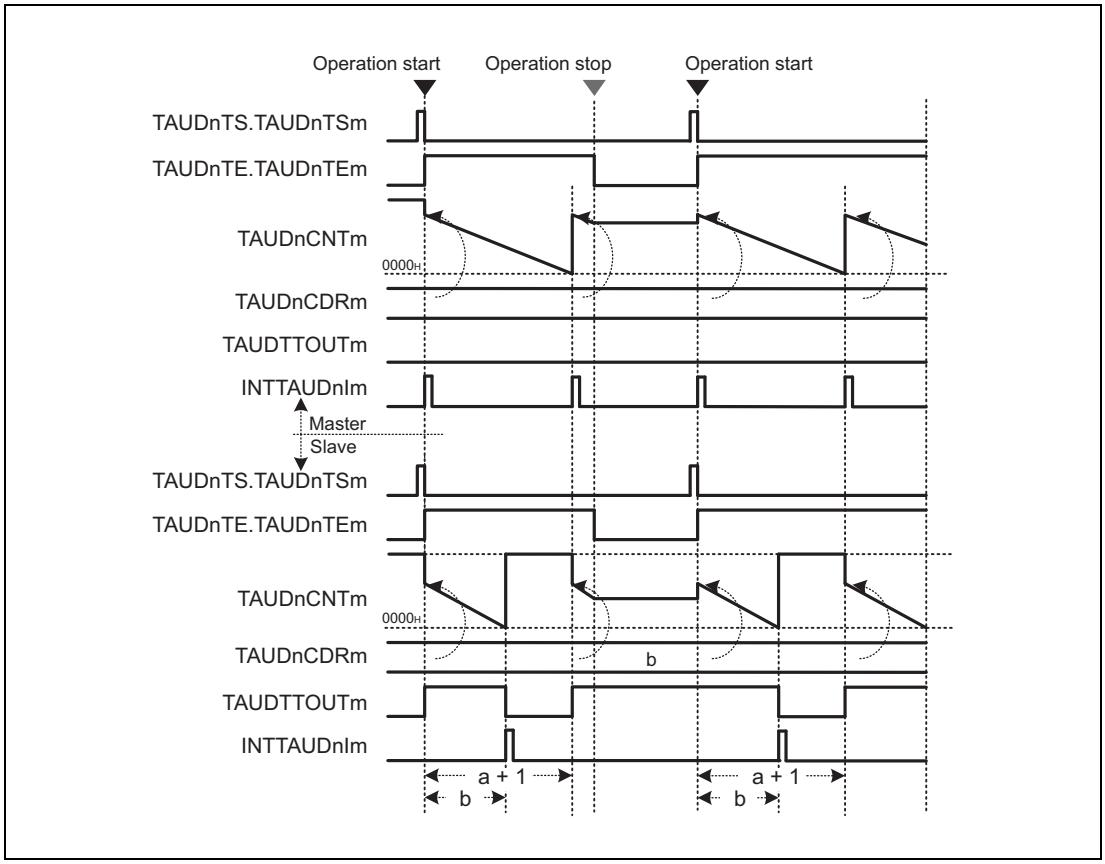


Figure 23.94 Operation Stop and Restart
Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)

- 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. TAUDnCNTm of master and slave channels reload the current values of TAUDnCDRm and start to count down from these values.

23.15.2 One-Shot Pulse Output Function

23.15.2.1 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 23.149, 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 23.152, 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 23.7, 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. This sets TAUDnTE.TAUDnTEm, 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 23.6, Simultaneous Rewrite**.

23.15.2.2 Equations

Delay from trigger input to pulse output
= (TAUDnCDRm (master) + 1) × count clock cycle

Pulse width = (TAUDnCDRm (slave)) × count clock cycle

23.15.2.3 Block Diagram and General Timing Diagram

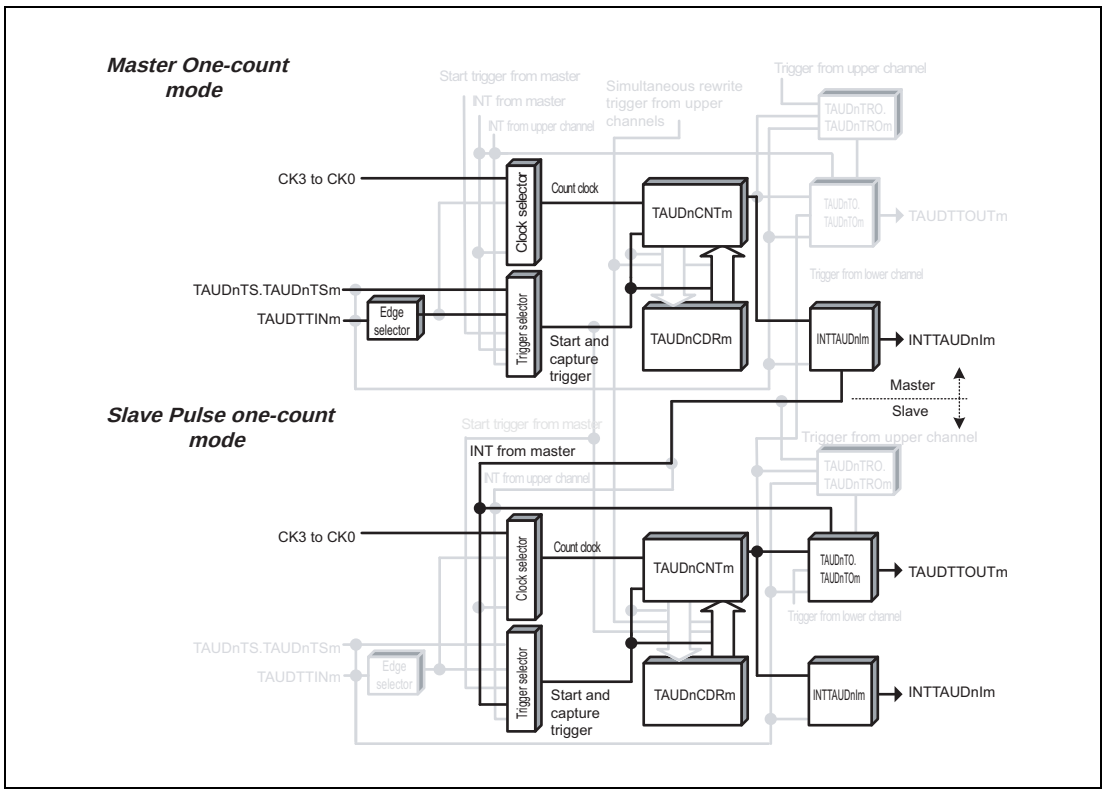


Figure 23.95 Block Diagram of One-Shot Pulse Output Function

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)

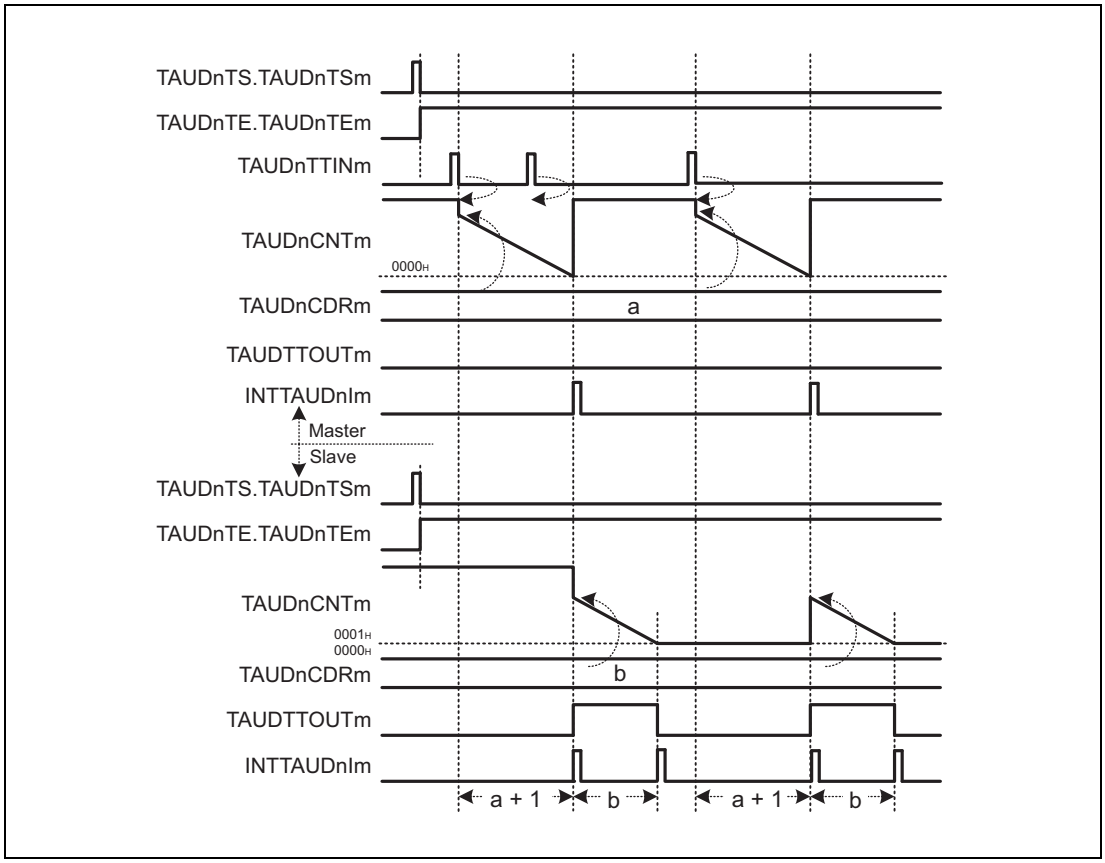


Figure 23.96 General Timing Diagram of One-Shot Pulse Output Function

23.15.2.4 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]		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 23.149 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 to this bit, 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 start trigger detection while counting. 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 23.150 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 to these bits, 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 23.151 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.

23.15.2.5 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 23.152 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 to this bit, 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 23.153 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the slave channel

Table 23.154 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 23.155 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.

23.15.2.6 Operating Procedure for One-Shot Pulse Output Function

Table 23.156 Operating Procedure for One-Shot Pulse Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.2.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm/ TAUDnCMURm register and channel output mode as described in **Section 23.15.2.5, 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 perform counting down. • 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

23.15.2.7 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)

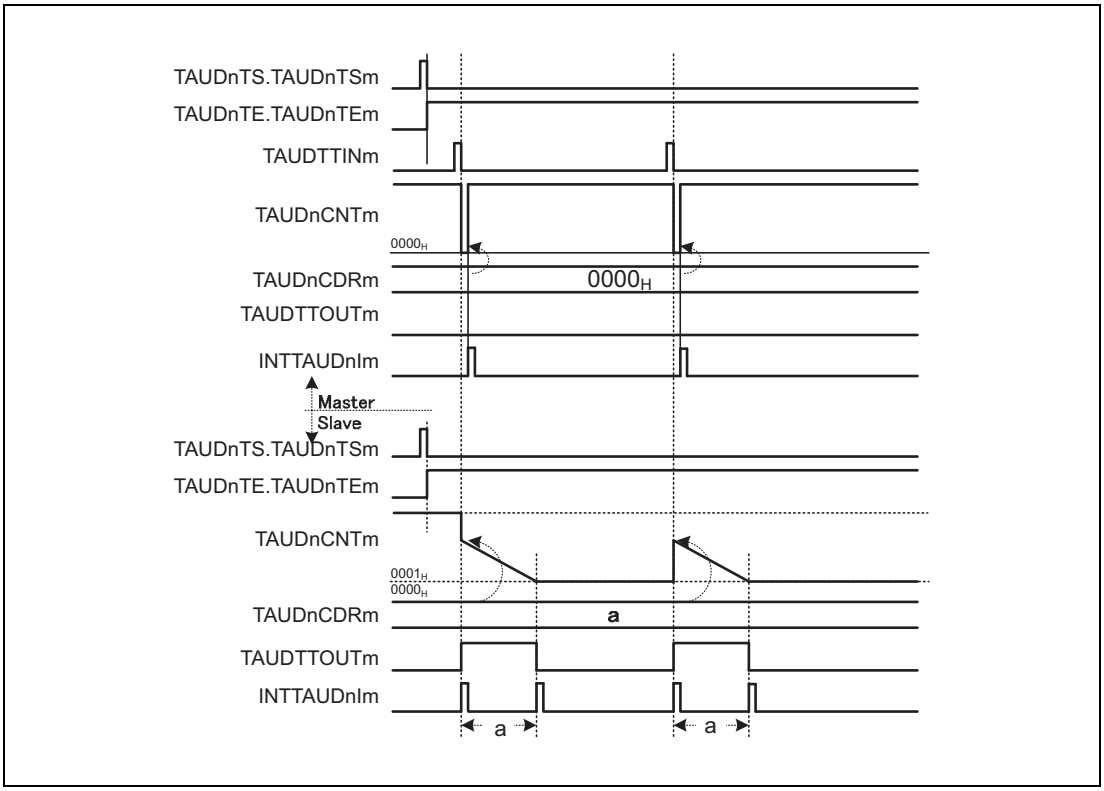


Figure 23.97 TAUDnCDRm (Master) = 0000_H

- 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)

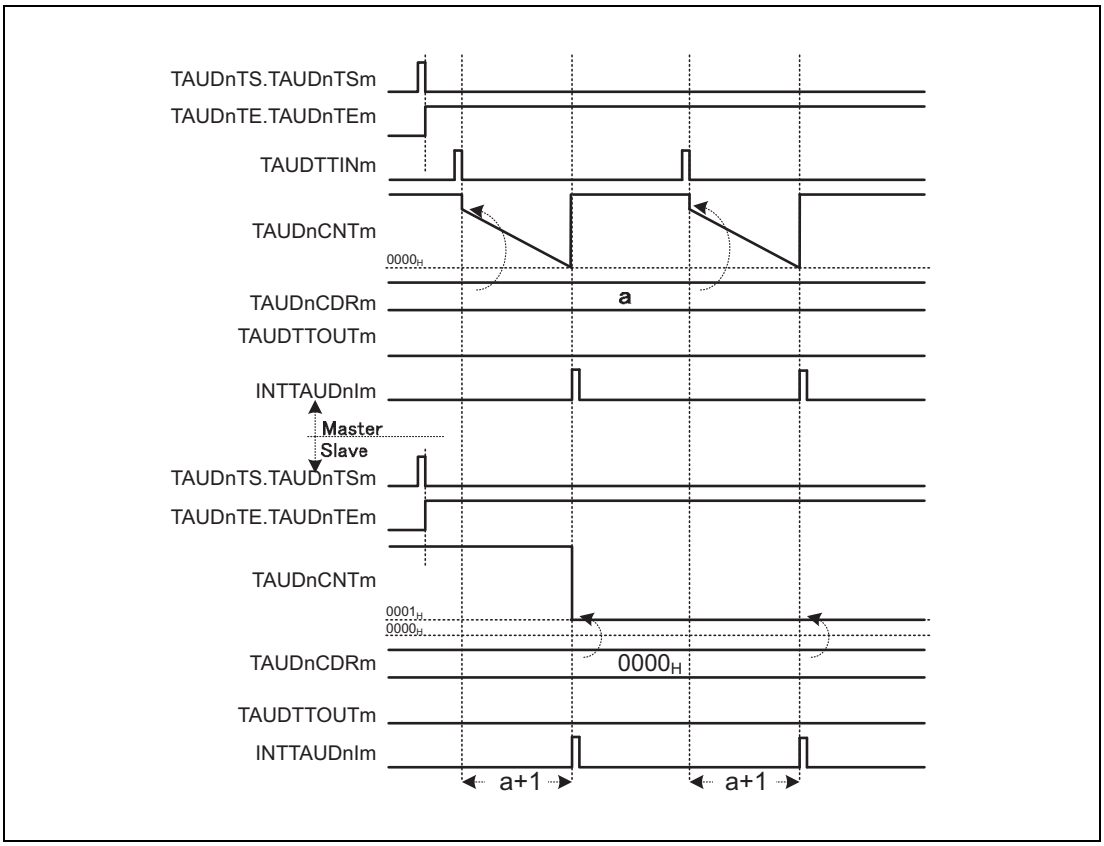


Figure 23.98 TAUDnCDRm (Slave) = 0000_H

- 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)

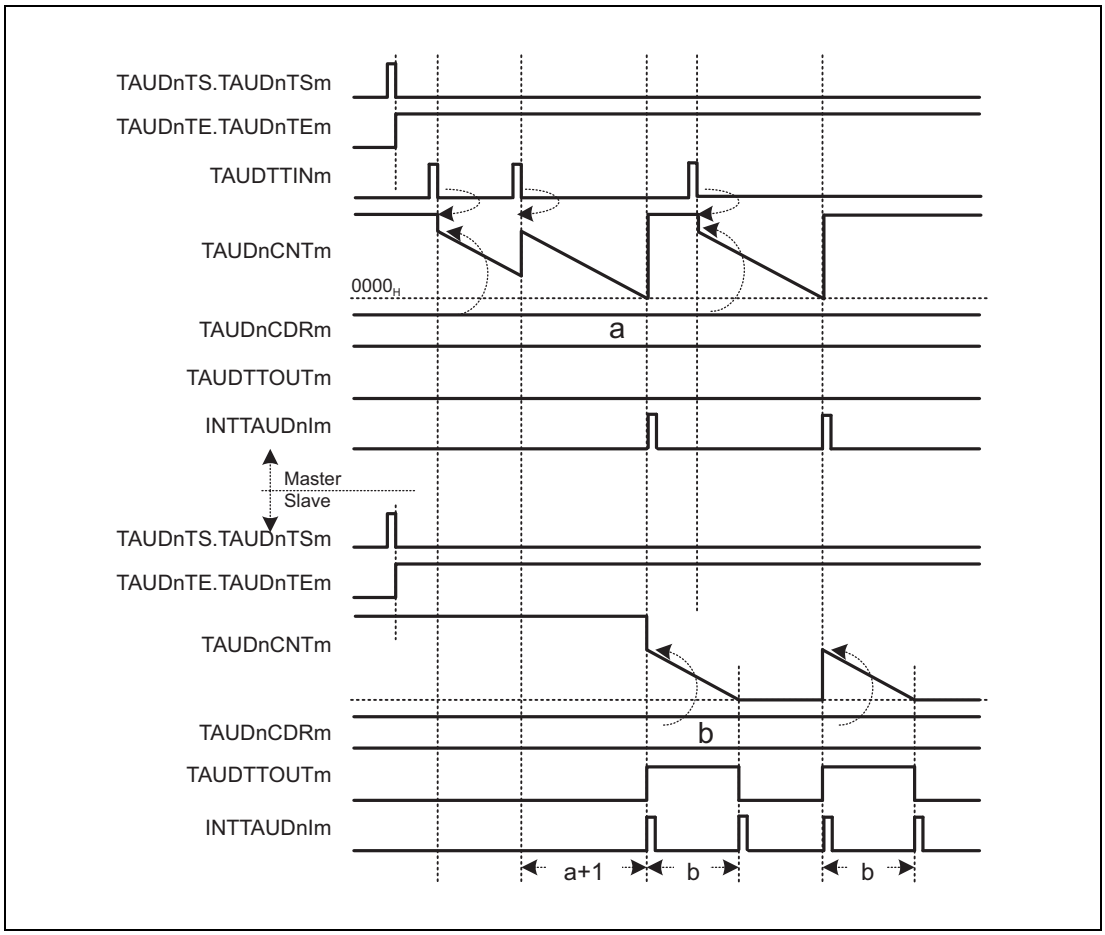


Figure 23.99 TAUDnCMORm.TAUDnMD0 = 1

- 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)

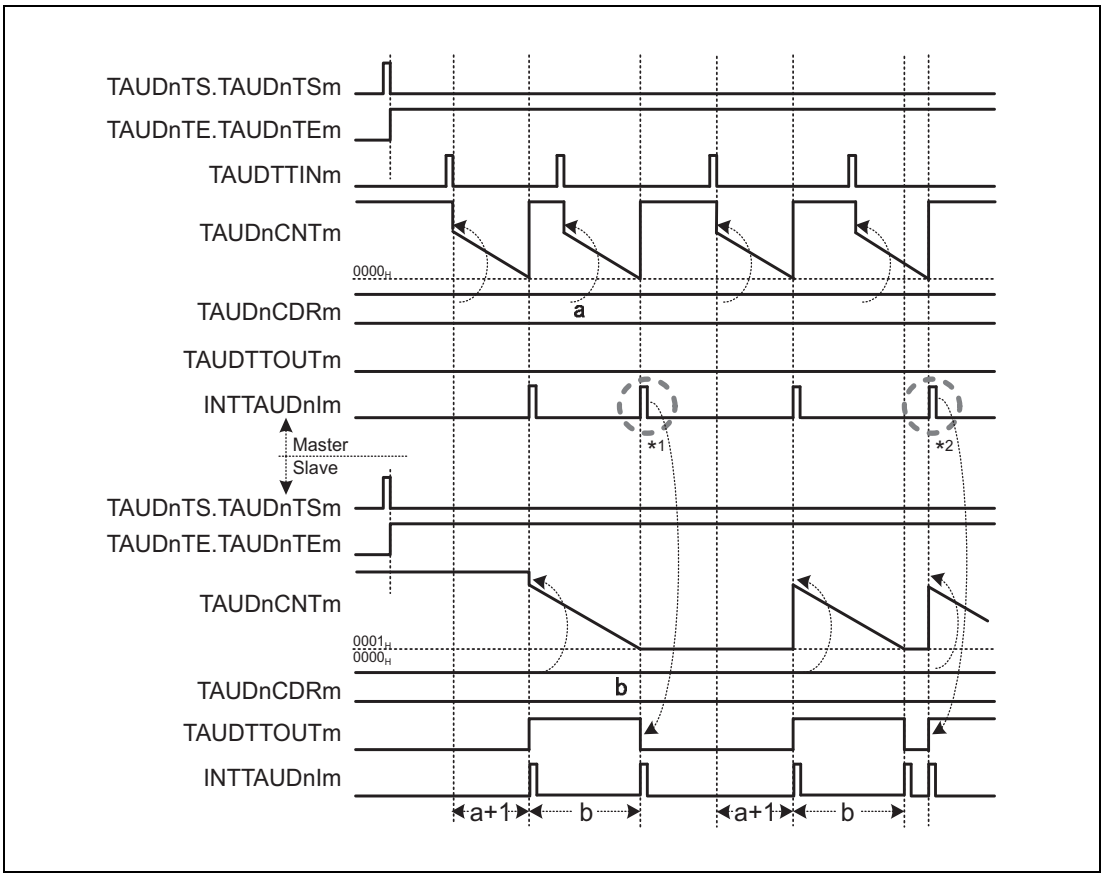


Figure 23.100 TAUDTTINm input interval ≤ Delay Time + Pulse Width +1

- 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.

23.15.3 Trigger Start PWM Output Function

23.15.3.1 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 23.157, 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 23.160, 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 23.7, 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, 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 realized 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) reload the current TAUDnCDRm values.
  If a valid TAUDTTINm input edge is detected, the counter of the master channel reloads 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 23.6, Simultaneous Rewrite**.

23.15.3.2 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

23.15.3.3 Block Diagram and General Timing Diagram

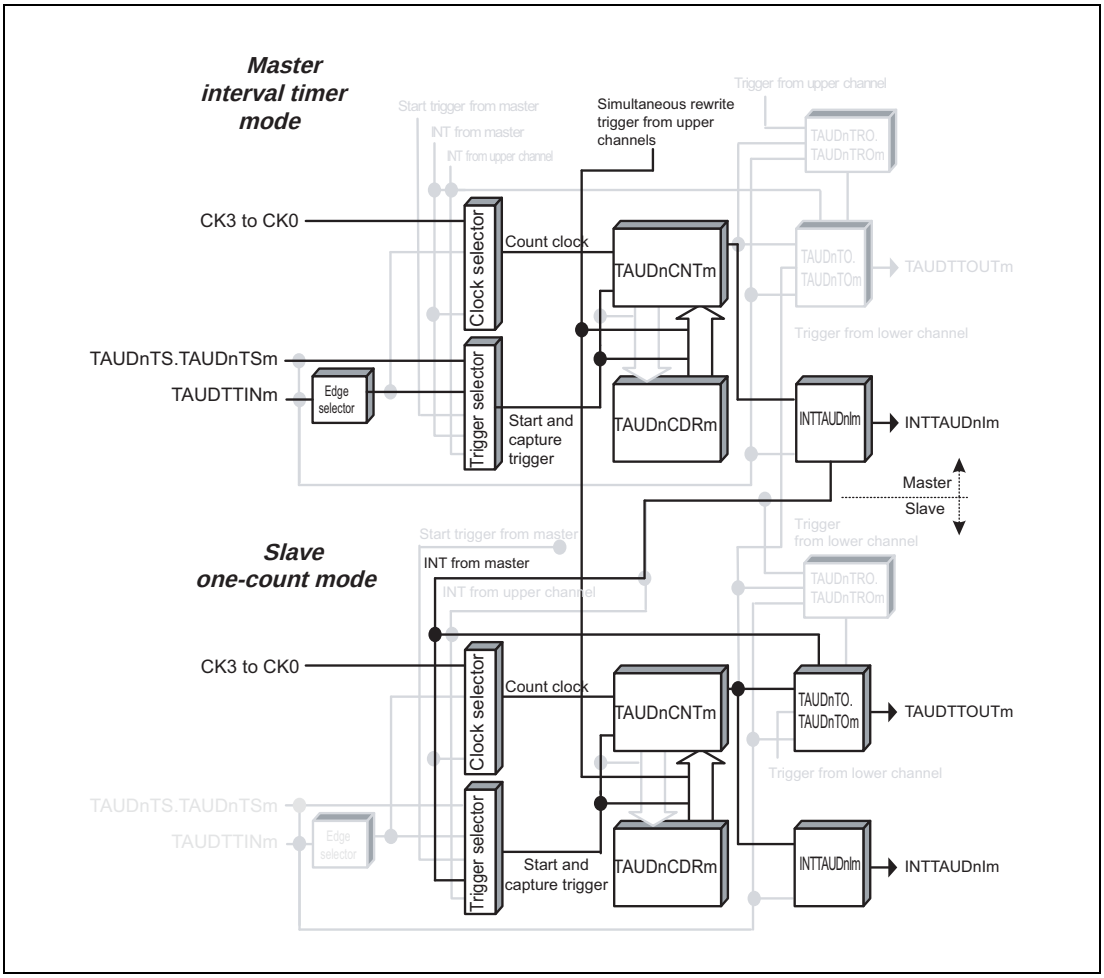


Figure 23.101 Block Diagram for Trigger Start PWM Output Function

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)

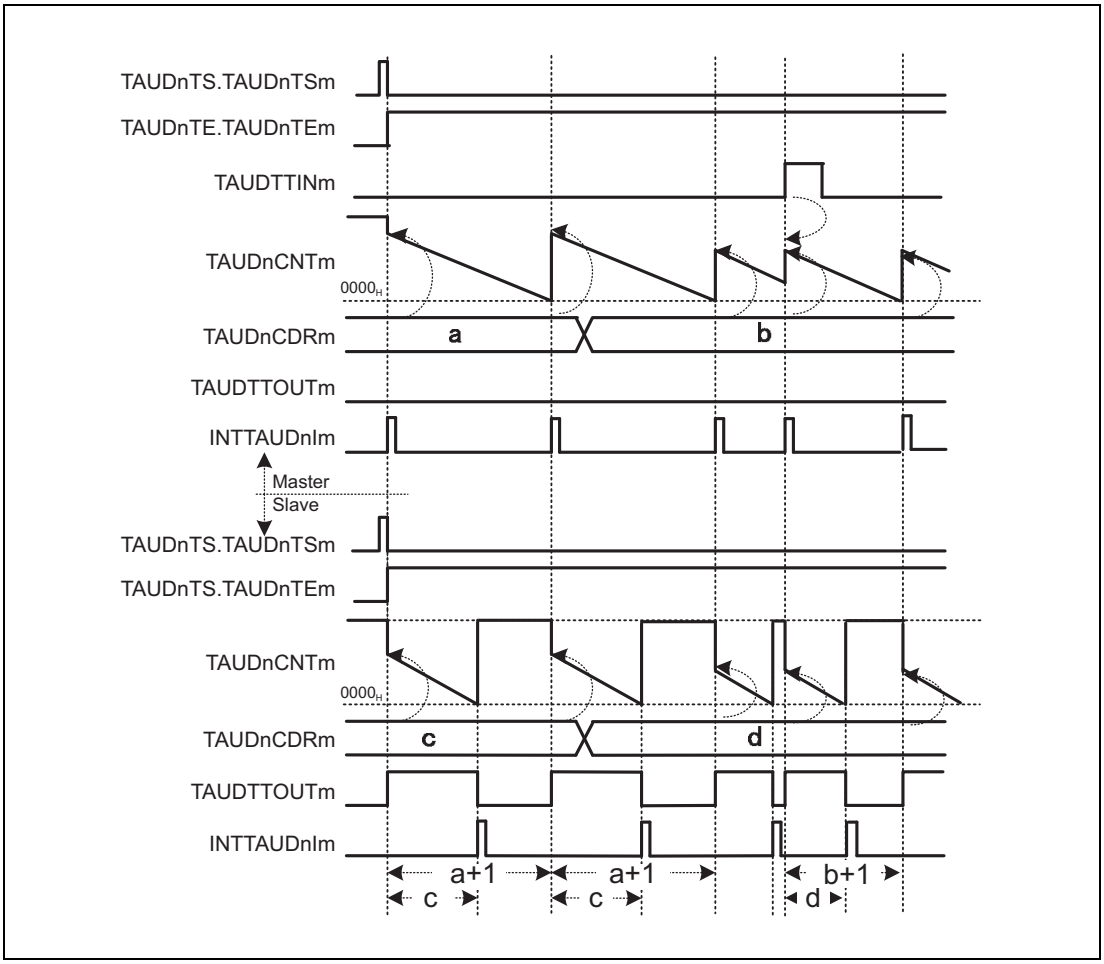


Figure 23.102 General Timing Diagram for Trigger Start PWM Output Function

NOTE

TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

23.15.3.4 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]		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 23.157 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 to this bit, 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 23.158 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 to these bits, 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 23.159 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.

23.15.3.5 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]		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 23.160 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 to this bit, 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 23.161 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the slave channel

Table 23.162 Control Bit Settings in Independent 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 23.163 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.

23.15.3.6 Operating Procedure for Trigger Start PWM Output Function

Table 23.164 Operating Procedure for Trigger Start PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.3.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.3.5, 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) reloads 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 the counter is counting down: • TAUDnCNTm (master and slave) reloads 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

23.15.3.7 Specific Timing Diagrams

(1) Duty cycle = 0%

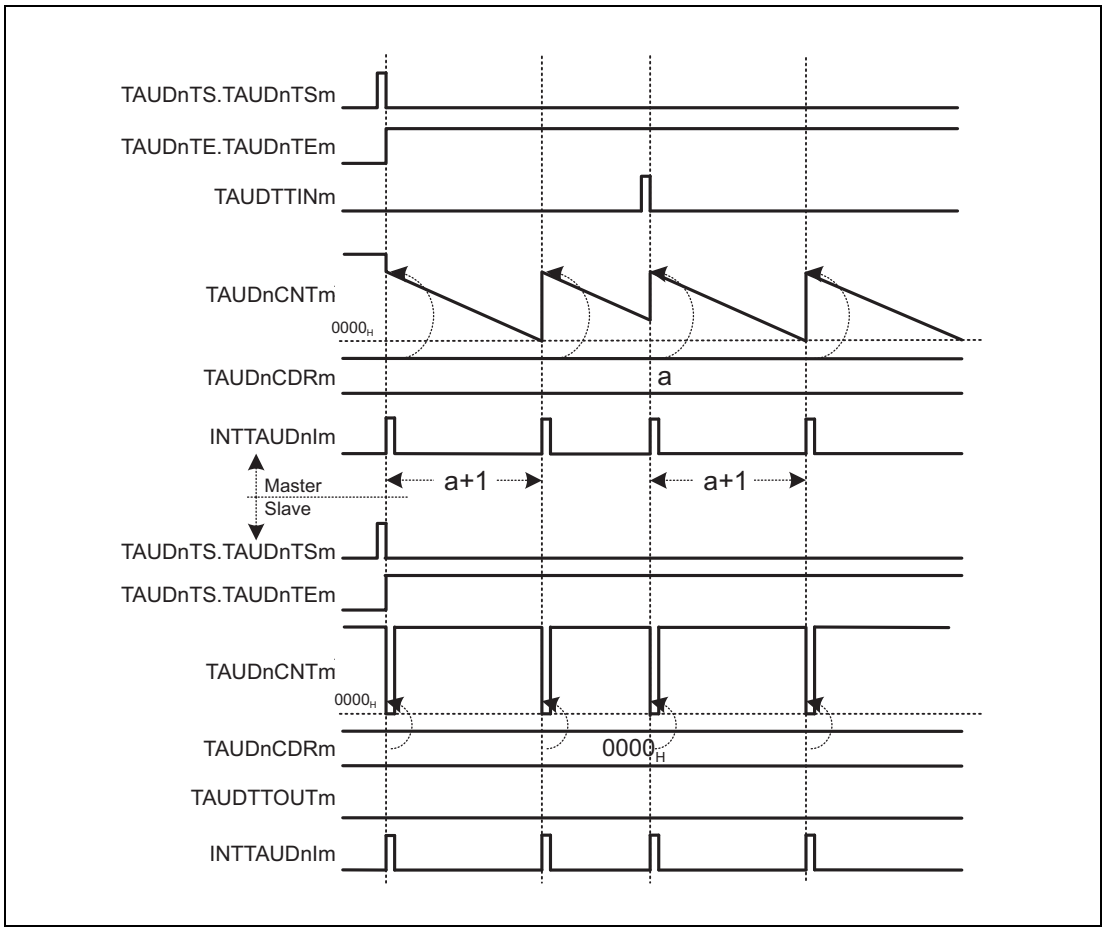


Figure 23.103 TAUDnCDRm (Slave) = 0000_H, Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0) Detection of Falling Edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- Every time the master channel generates an interrupt (INTTAUDnIm), 0000_H is written 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%

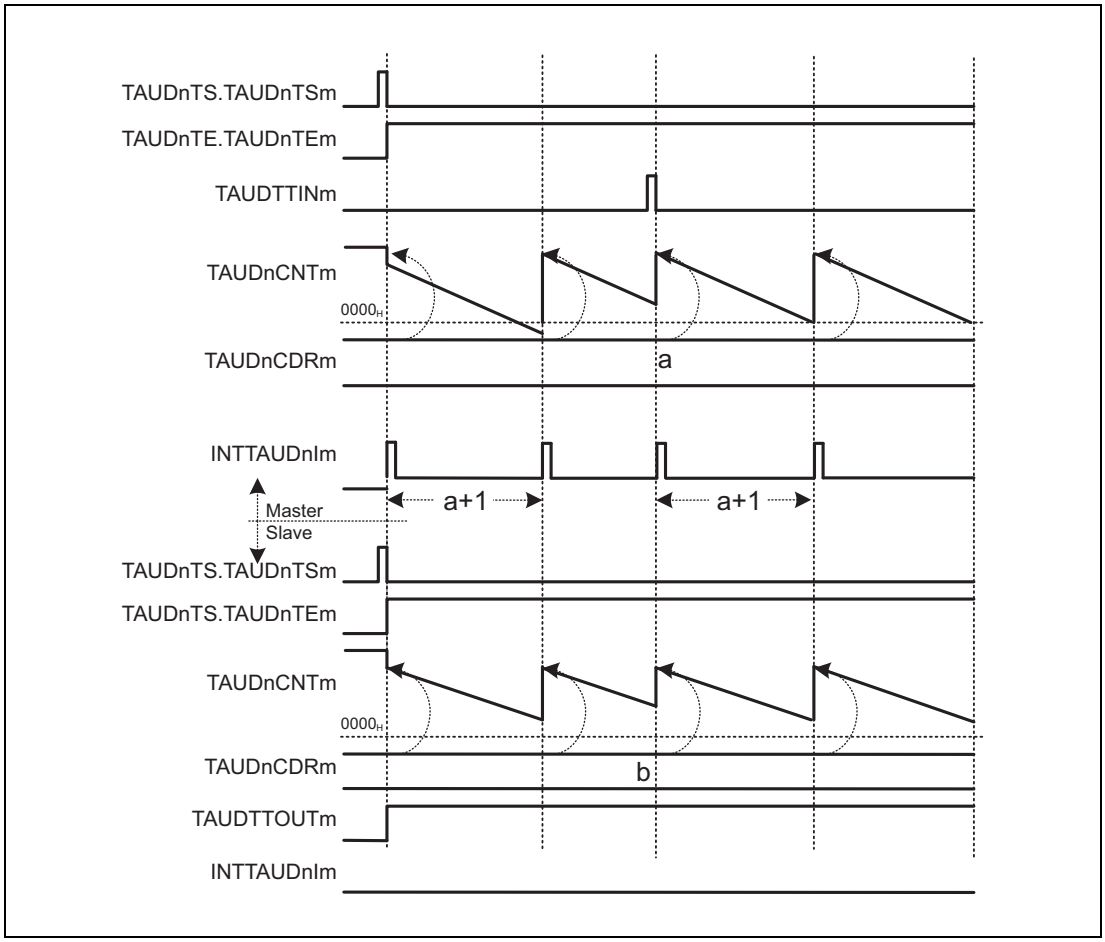


Figure 23.104 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1, Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0) Falling Edge Detection (TAUDnCMURm.TIS[1:0] = 00_B)

- 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 at active state.
  The detection of a valid TAUDTTINm input edge has no effect on TAUDTTOUTm (slave).

(3) TAUDTTINm detection and active slave counter

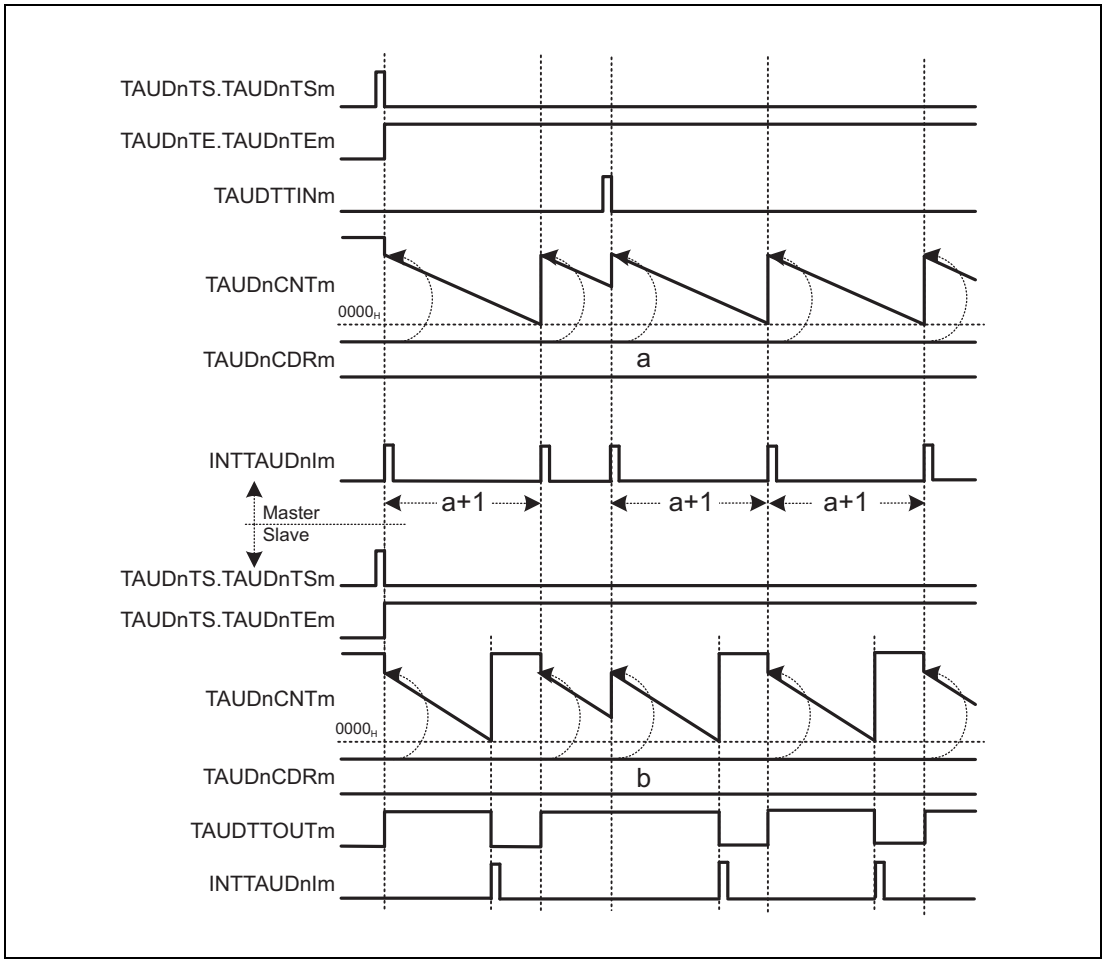


Figure 23.105 Positive Logic (TAUDnTOL.TAUDnTOLm (Slave) = 0)
Detection of Falling Edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

- If TAUDnCNTm (slave) reloads the value TAUDnCDRm (slave) while it is still counting down, TAUDTTOUTm cannot toggle and extends the duty.
  The duty does not correspond to the value of the slave's data register.

23.15.4 Delay Pulse Output Function

23.15.4.1 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 23.165, 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 23.168, Contents of the TAUDnCMORm Register for Slave Channel 1 of the Delay Pulse Output Function**.)
- The operating mode for slave channel 3 should be set to pulse one-count mode. (See **Table 23.172, Contents of the TAUDnCMORm Register for Slave Channel 2 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 23.7, Channel Output Modes**.)
- The channel output mode for slave channel 3 should be set to independent channel output mode 2. (See **Section 23.7, 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), INTTAUDnlm is generated and TAUDTTOUTm signal is reset. The counter is reset to $FFFF_H$ and waits for the next INTTAUDnlm 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 INTTAUDnlm 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 23.6, Simultaneous Rewrite**.

23.15.4.2 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).

23.15.4.3 Block Diagram and General Timing Diagram

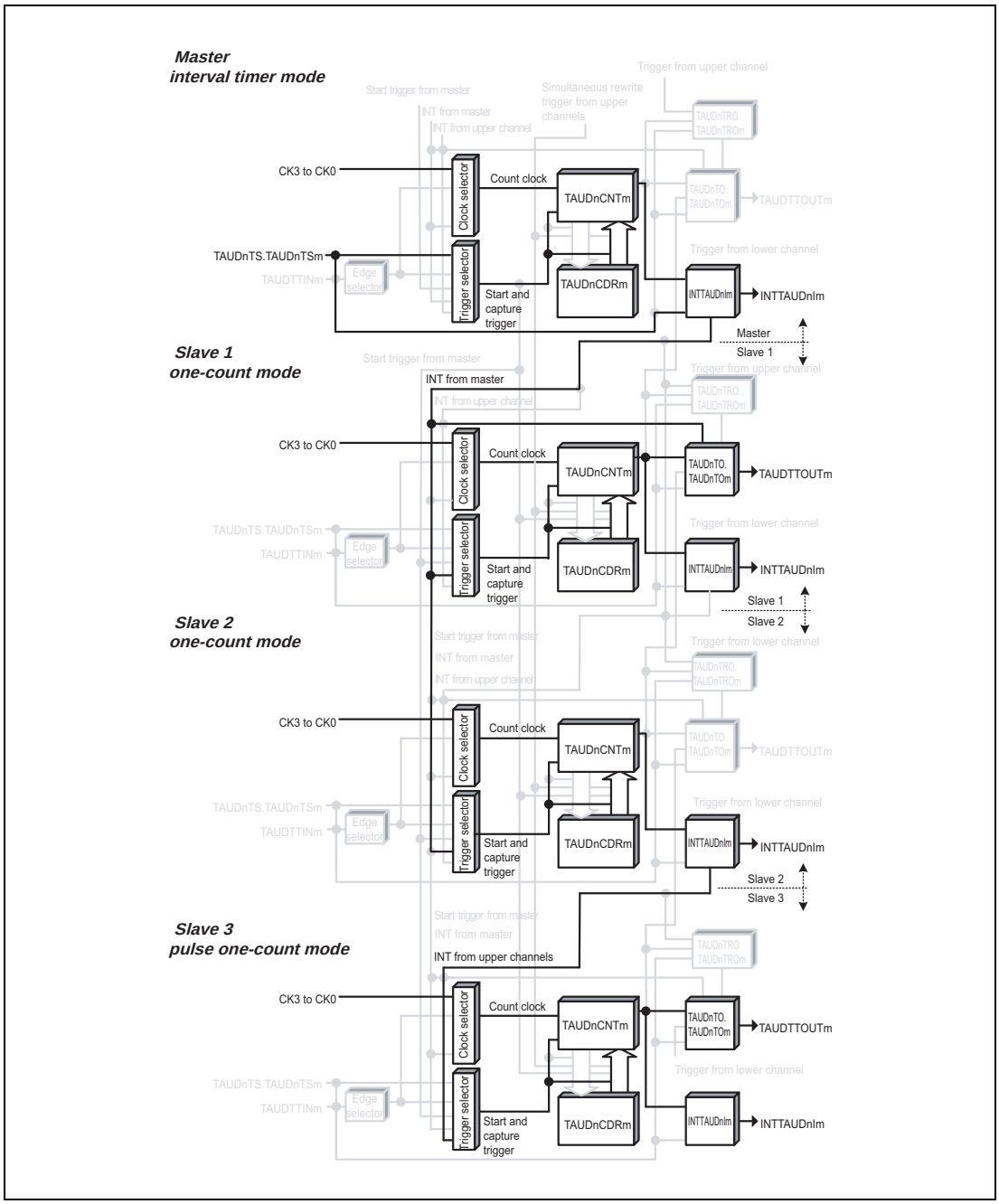


Figure 23.106 Block Diagram of Delay Pulse Output Function

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)

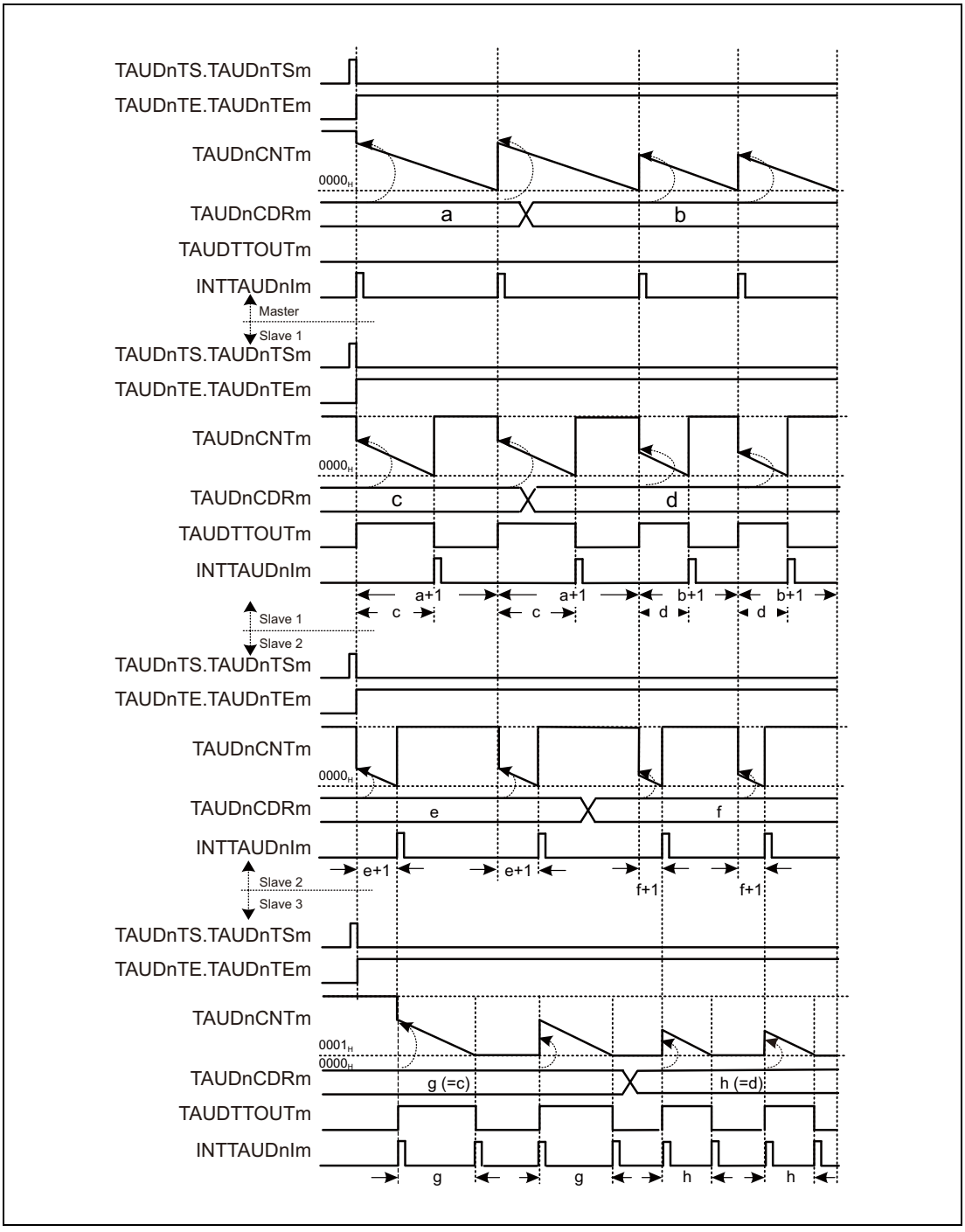


Figure 23.107 General Timing Diagram of Delay Pulse Output Function

NOTE

TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

23.15.4.4 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]		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 23.165 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 to this bit, 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 23.166 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 to these bits, 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 23.167 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.

23.15.4.5 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 23.168 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 to this bit, 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 23.169 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channel 1

Table 23.170 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 23.171 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.

23.15.4.6 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 23.172 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 to this bit, 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 23.173 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 to these bits, 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 23.174 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.

23.15.4.7 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 23.175 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 to this bit, 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 23.176 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channel 3

Table 23.177 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.TDEm = 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.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

(4) Simultaneous rewrite for slave channel 3

Both the master and slave channels should have the same simultaneous rewrite settings.

Table 23.178 Simultaneous Rewrite Settings for Slave Channel 3 of Delay Pulse Output Function

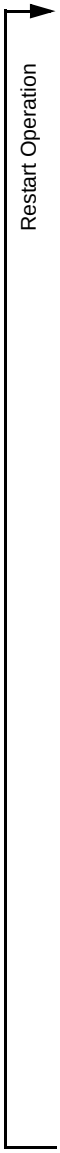
Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Enables simultaneous rewrite.
TAUDnRDS.TAUDnRDSm	0: Master channel is simultaneous rewrite control channel.
TAUDnRDM.TAUDnRDMm	0: Generates a simultaneous rewrite trigger signal when the master channel starts to count.
TAUDnRDC.TAUDnRDCm	0: Does not operate as a simultaneous rewrite trigger generation channel.

23.15.4.8 Operating Procedure for Delay Pulse Output Function

Table 23.179 Operating Procedure for Delay Pulse Output Function (1/2)

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.4.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 23.15.4.5, Register Settings for Slave Channel 1**. Slave channel 2: Set TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 23.15.4.6, Register Settings for Slave Channel 2**. Slave channel 3: Set the TAUDnCMORm and TAUDnCMURm registers and the channel output mode as described in **Section 23.15.4.7, Register Settings for Slave Channel 3**. Set the value of TAUDnCDRm register of every channel.	Channel operation is stopped.

Table 23.179 Operating Procedure for Delay Pulse Output Function (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 (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.	TAUDnCNTm of master channel and slave channels 1 and 2 load TAUDnCDRm value 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.

23.15.4.9 Specific Timing Diagrams

(1) Duty cycle (slave 3) = 100%

The following values apply to **Figure 23.108**:

- TAUDnCDRm (master) = $000A_H$
- TAUDnCDRm (slave 1) = $000B_H$
- TAUDnCDRm (slave 2) = 0000_H
- TAUDnCDRm (slave 3) = $000B_H$

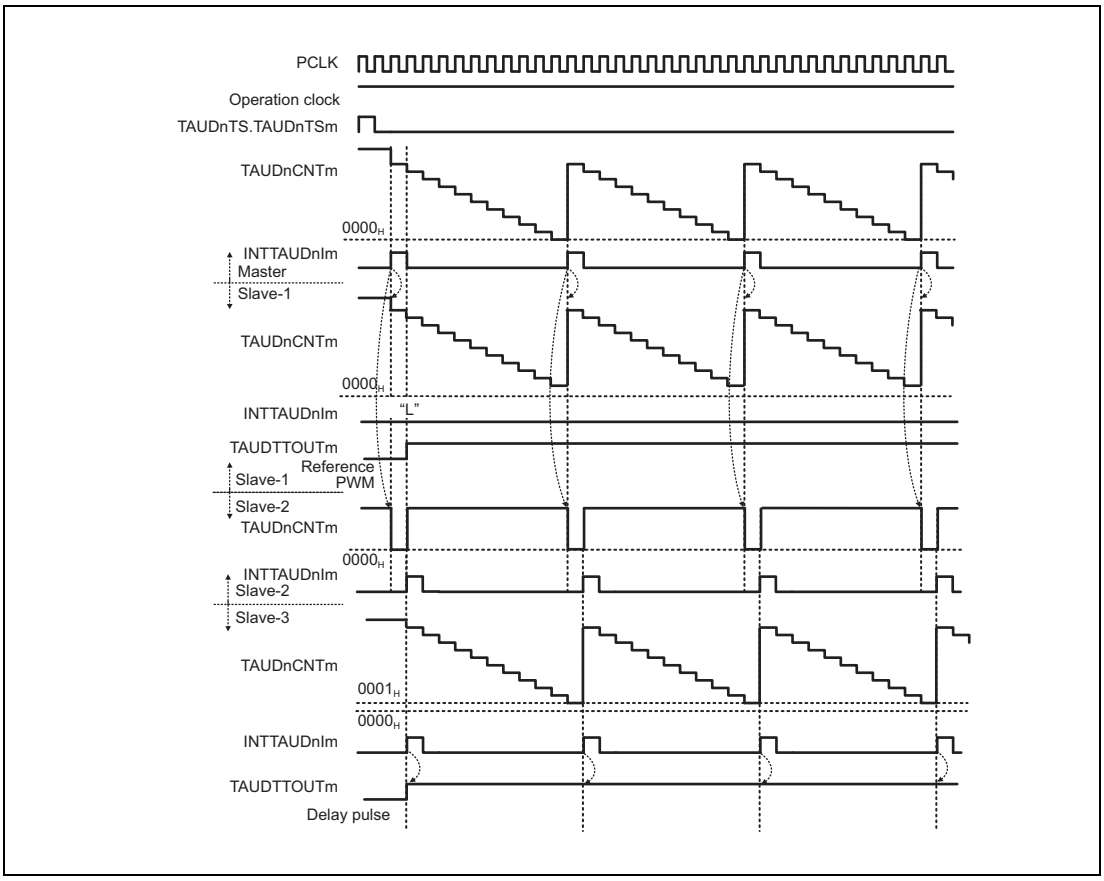


Figure 23.108 Duty Cycle (Slave 3) = 100%

- If the value of TAUDnCDRm (slave 1 and 3) is higher than the value of TAUDnCDRm (master), the counter of the slave channels 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 23.109**.

- TAUDnCDRm(master) = $000A_H$
- TAUDnCDRm(slave 1) = 0005_H
- TAUDnCDRm(slave 2) = 0000_H
- TAUDnCDRm(slave 3) = 0005_H

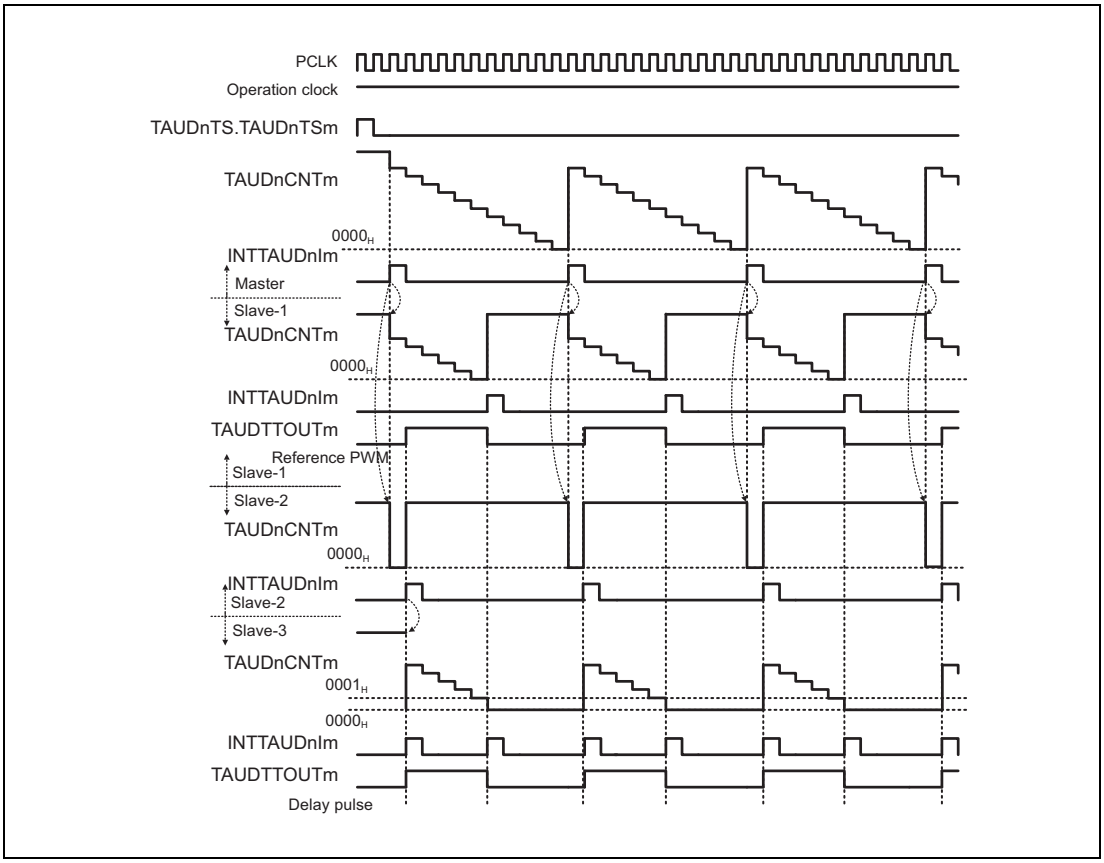


Figure 23.109 TAUDTTOUTm (Slave 1) = TAUDTTOUTm (Slave 3)

- 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.

23.15.5 Offset Trigger Output Function

23.15.5.1 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 23.180, 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 23.183, 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 23.7, 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:
  The INTTAUDnIm of master channel sets the TAUDTTOUTm (slave) signal and triggers the counter of the slave channel. 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.

23.15.5.2 Equations

Pulse width = (TAUDnCDRm (slave) + 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

23.15.5.3 Block Diagram and General Timing Diagram

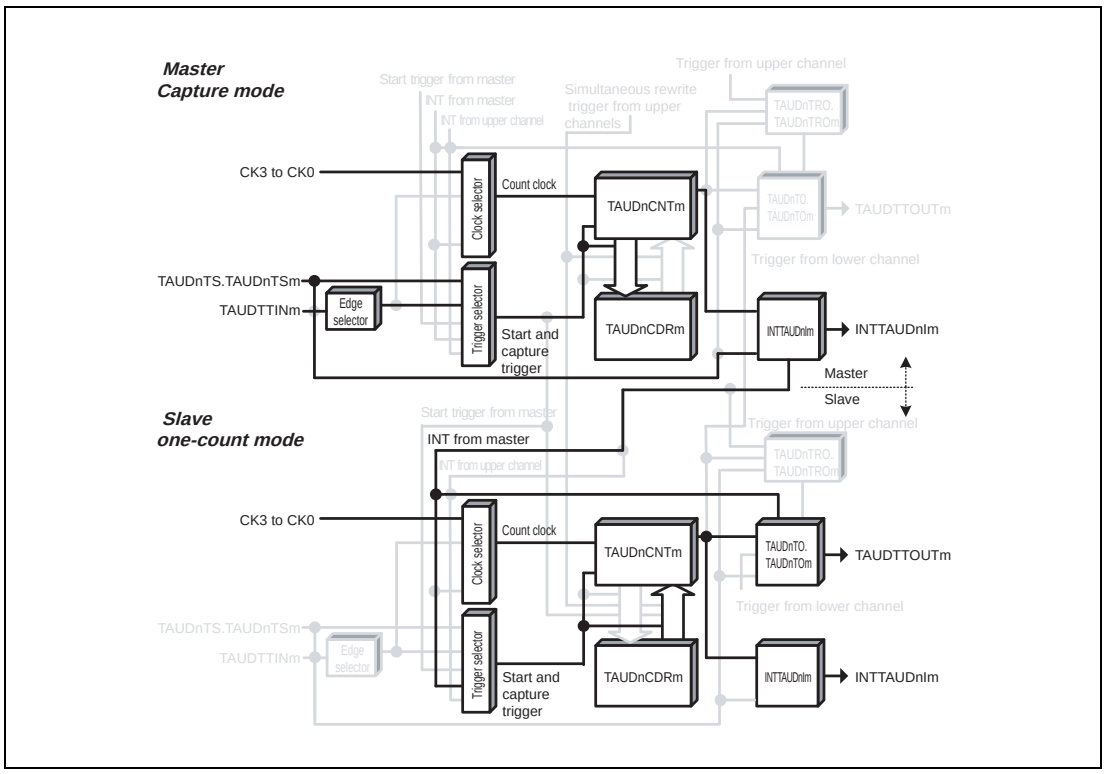


Figure 23.110 Block Diagram of Offset Trigger Output Function

The following settings apply to the general timing diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

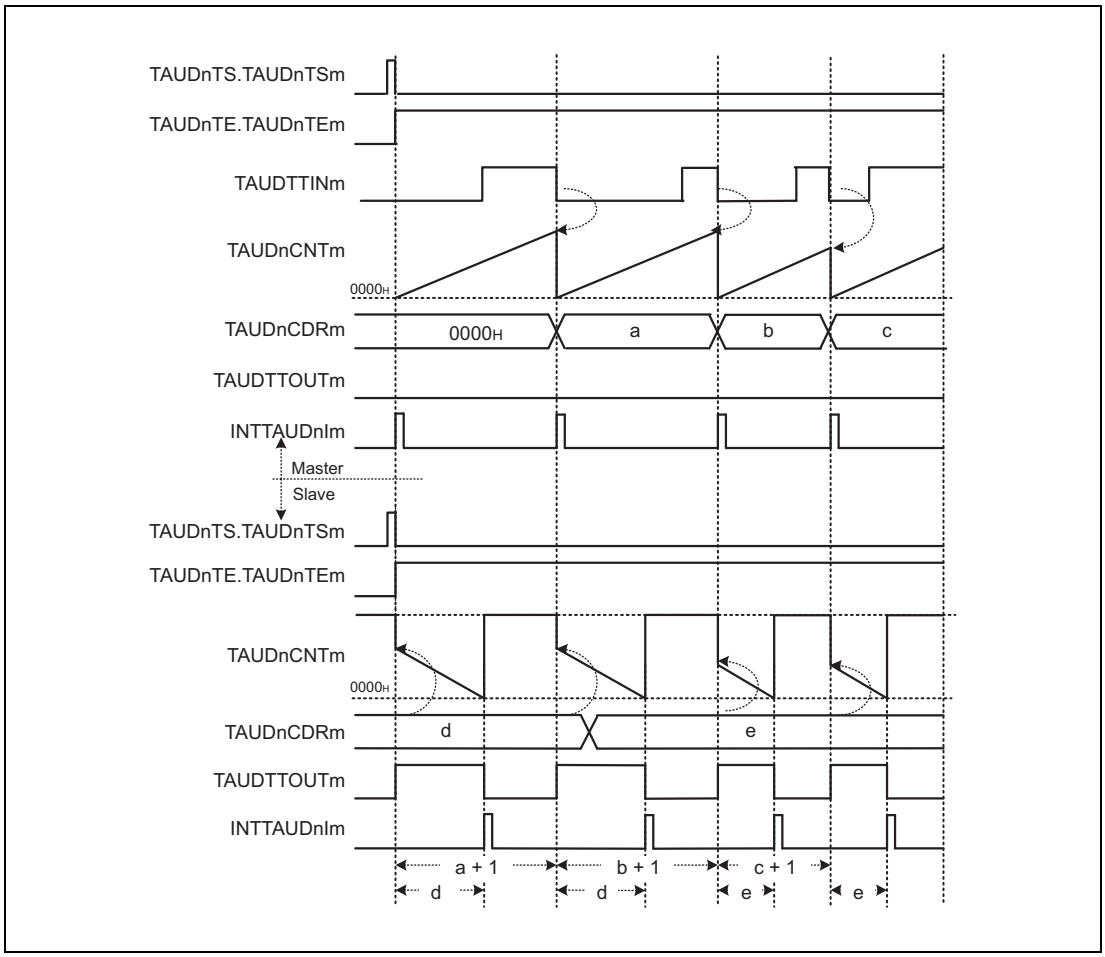


Figure 23.111 General Timing Diagram of Offset Trigger Output Function

NOTE

TAUDTTOUTm of the slave channel rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

23.15.5.4 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]		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 23.180 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 to this bit, 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 23.181 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 to these bits, 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 23.182 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	

23.15.5.5 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]		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 23.183 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 to this bit, 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 23.184 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channels

Table 23.185 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 23.186 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	

23.15.5.6 Operating Procedure for Offset Trigger Output Function

Table 23.187 Operating Procedure for Offset Trigger Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.5.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm/ TAUDnCMURm register and channel output mode as described in **Section 23.15.5.5, 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 TAUDnCSR.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

23.15.5.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to this diagram.

- Detection of falling edge (TAUDnCMURm.TAUDnTIS[1:0] = 00_B)

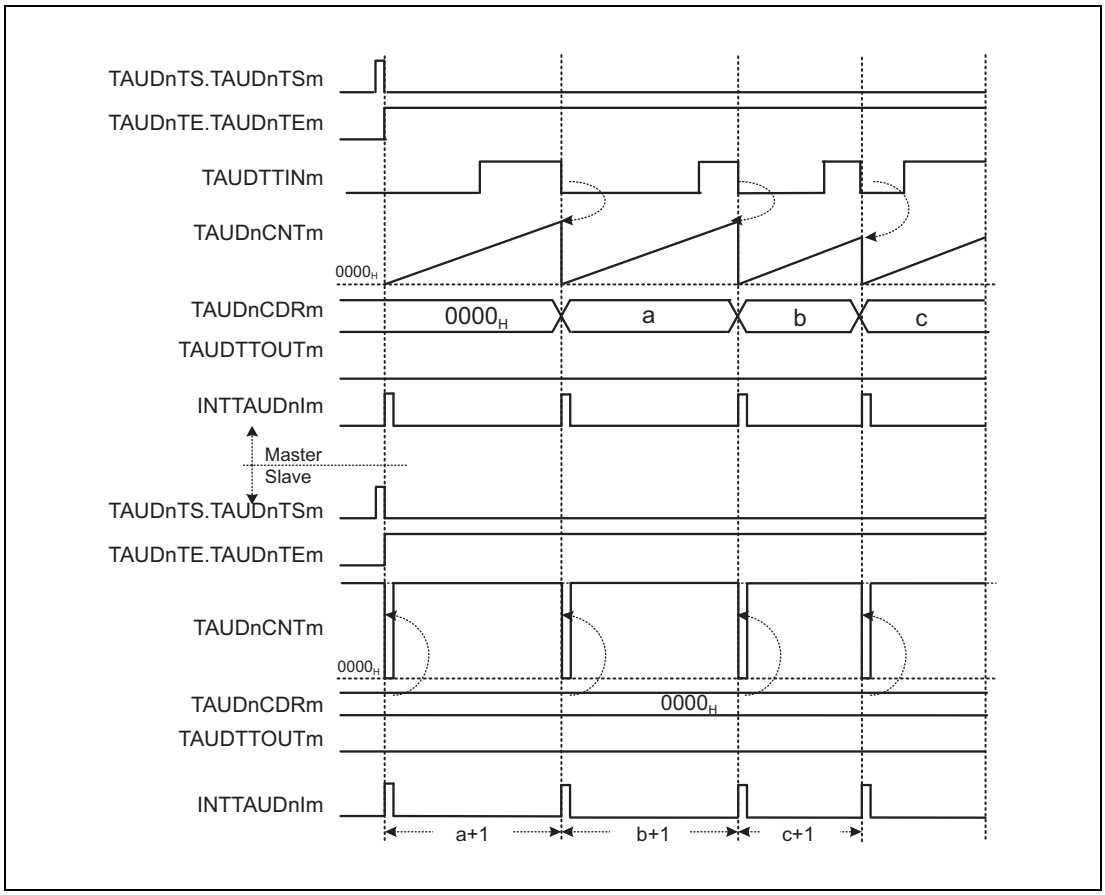


Figure 23.112 TAUDnCDRm (Slave) = 0000_H

- 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)

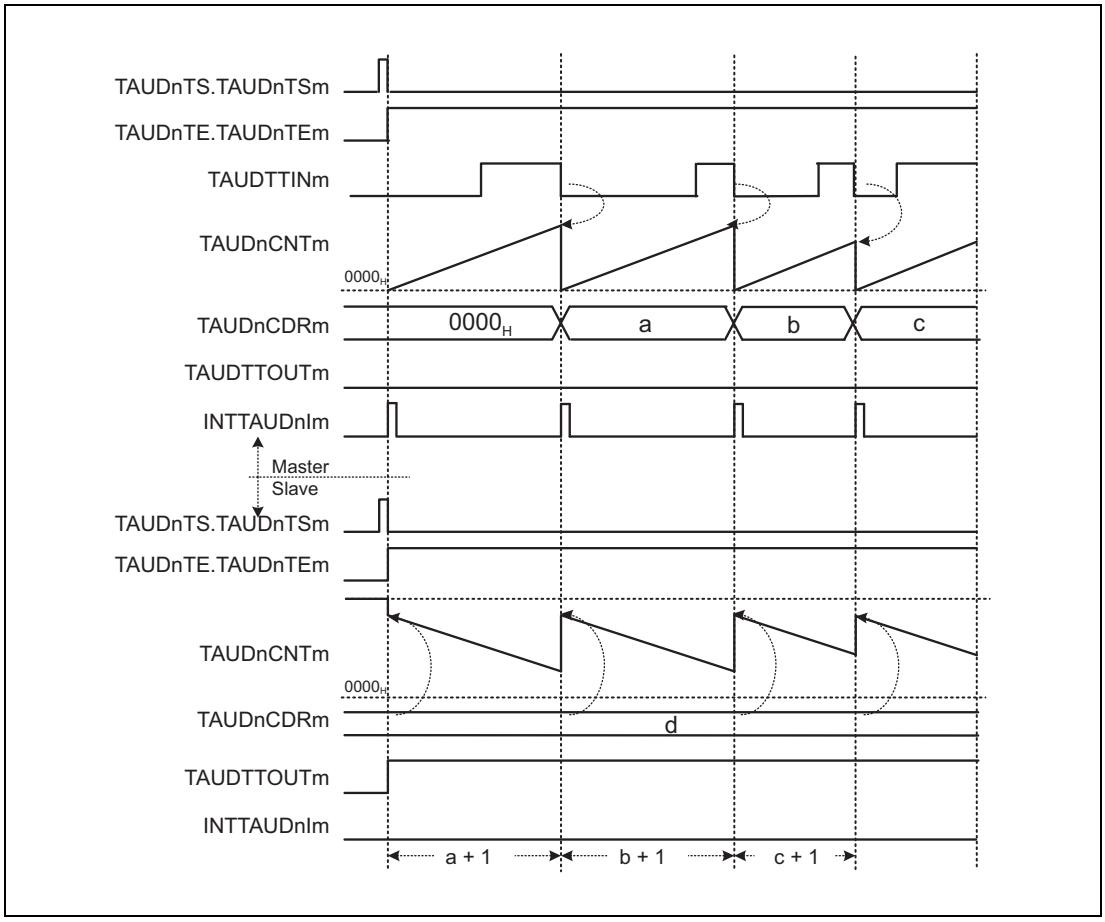


Figure 23.113 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1

- 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 at active state.

23.15.6 A/D Conversion Trigger Output Function Type 1

23.15.6.1 Overview

Summary

This function is identical to **Section 23.15.1, PWM Output Function**, except that TAUDTTOUTm is not output.

This is achieved by setting the channel output mode for the slave to independent channel output mode controlled by software.

23.15.6.2 Block Diagram and General Timing Diagram

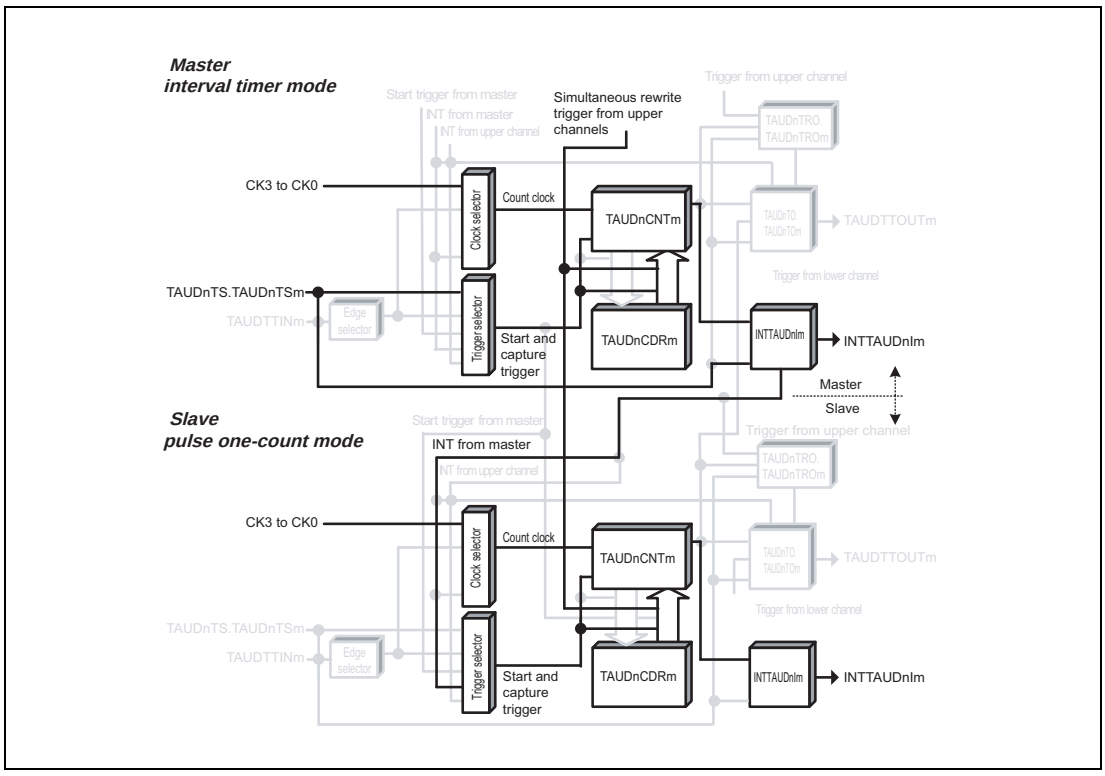


Figure 23.114 Block Diagram of A/D Conversion Trigger Output Function Type 1

The following settings apply to the general timing diagram.

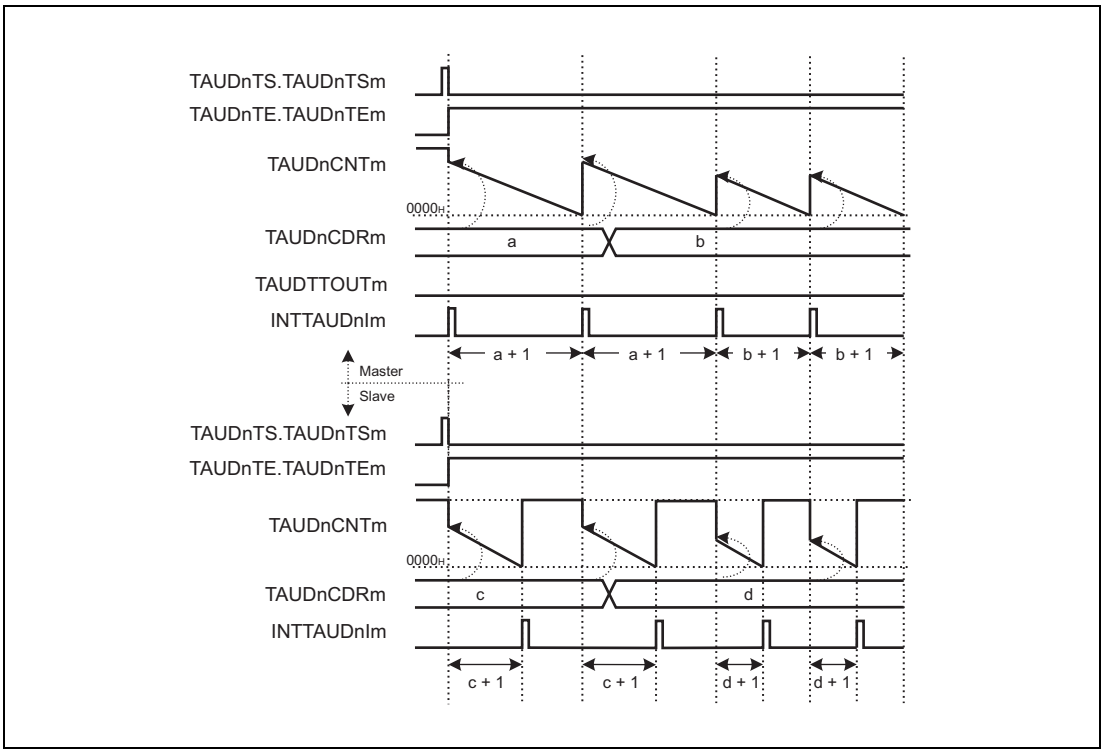


Figure 23.115 General Timing Diagram of A/D Conversion Trigger Output Function Type 1

23.15.7 Triangle PWM Output Function

23.15.7.1 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 23.188, 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 23.192, 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 23.7, Channel Output Modes**.)
- The channel output mode for slave channels should be set to synchronous channel output mode 2. (See **Section 23.7, Channel Output Modes**.)
- The following settings allows 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 INTTAUDnlm 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 23.6, Simultaneous Rewrite**.

23.15.7.2 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

23.15.7.3 Block Diagram and General Timing Diagram

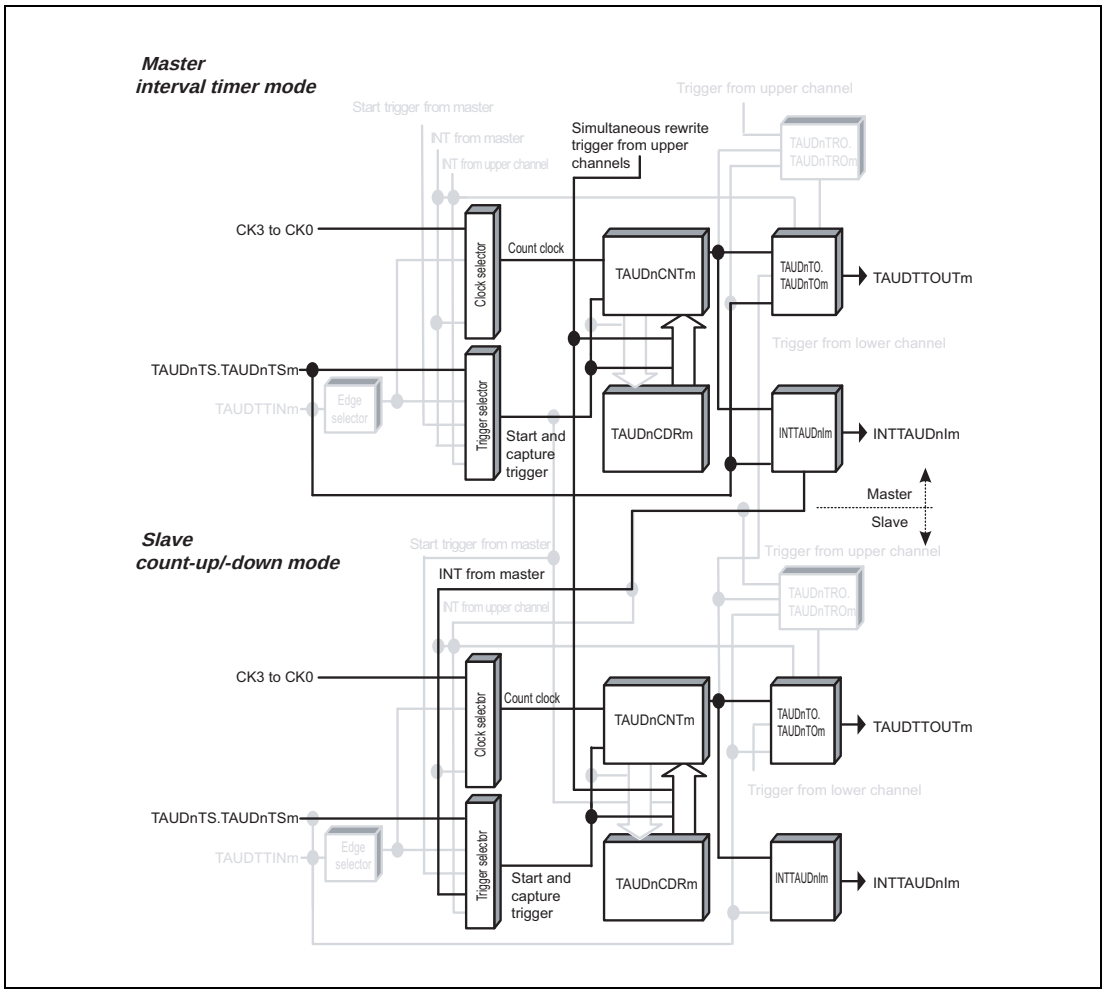


Figure 23.116 Block Diagram of Triangle PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUDnIm is generated at the beginning of operation.
    (TAUDnCMORm.TAUDnMD0 = 1)

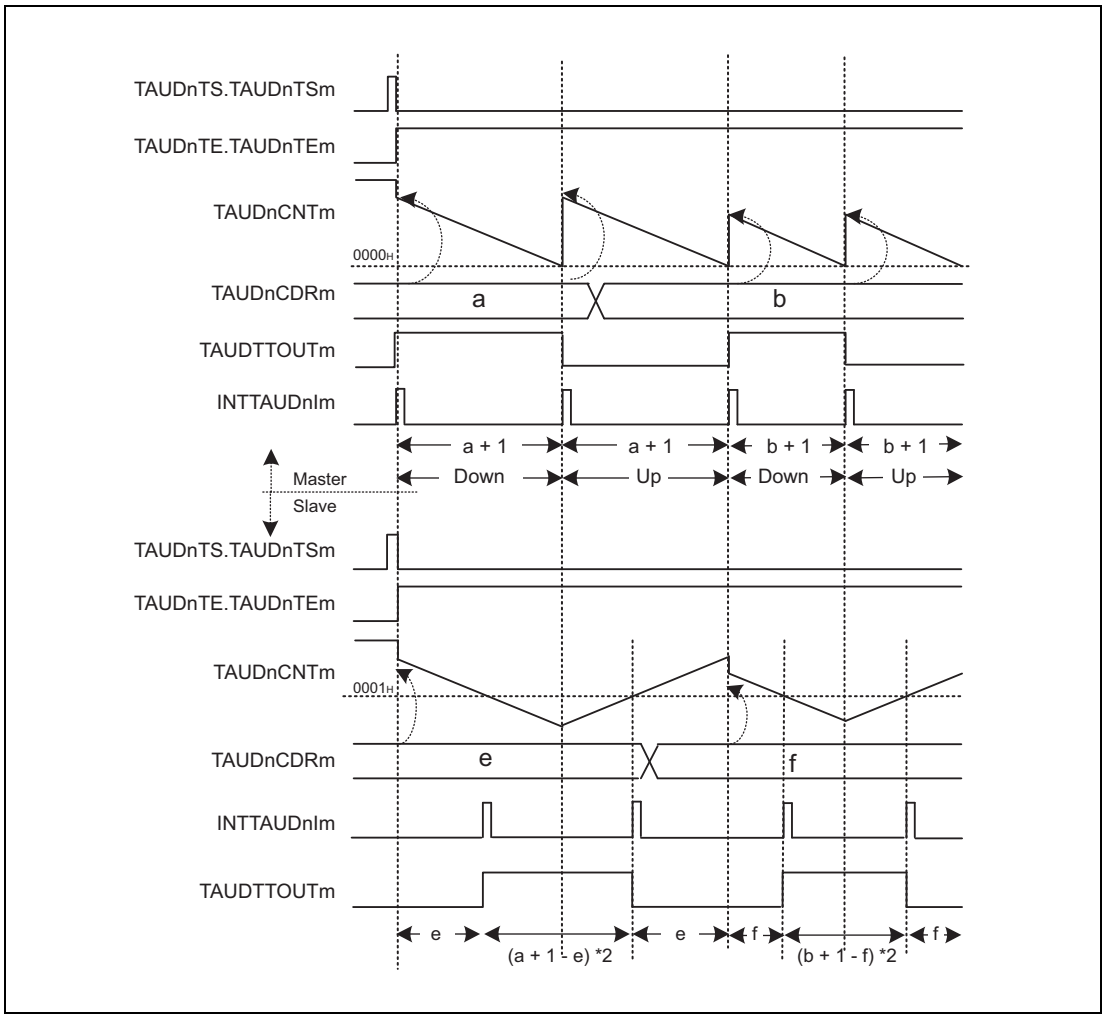


Figure 23.117 General Timing Diagram of Triangle PWM Output Function

23.15.7.4 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]		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 23.188 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 to this bit, 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 23.189 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the master channel

Table 23.190 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 23.191 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.

23.15.7.5 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 23.192 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 to this bit, 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 23.193 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channels

Table 23.194 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	1: 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 23.195 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.

23.15.7.6 Operating Procedure for Triangle PWM Output Function

Table 23.196 Operating Procedure for Triangle PWM Output Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.7.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.7.5, 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. • TAUDnTTOUTm (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

23.15.7.7 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

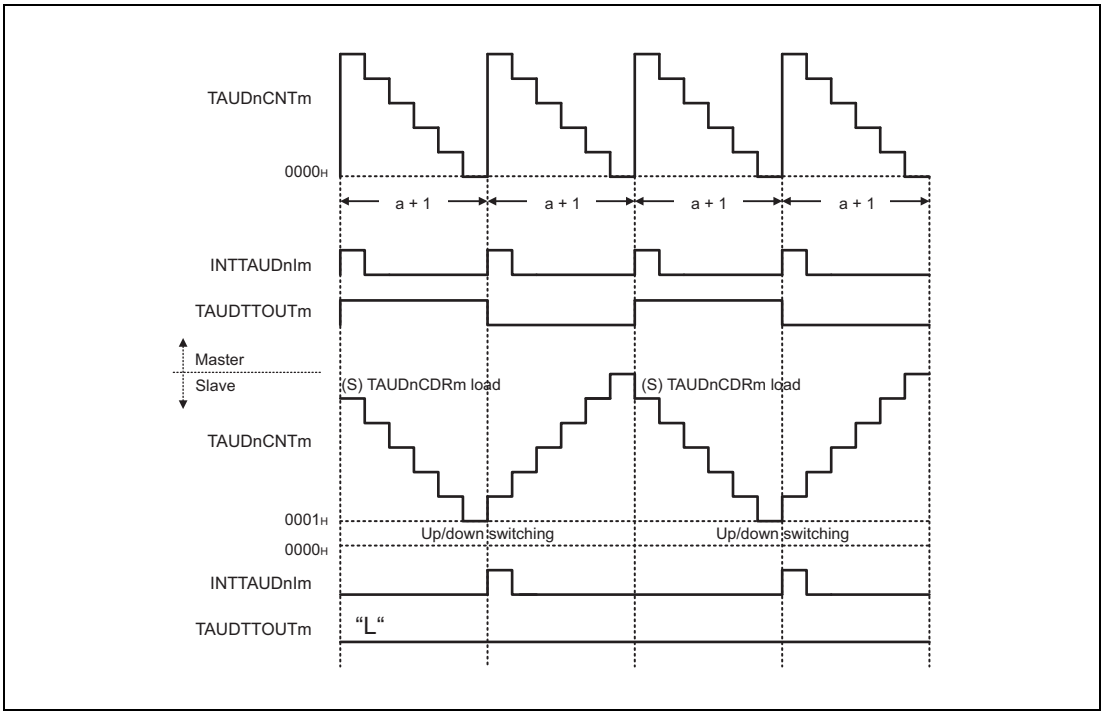


Figure 23.118 TAUDnCDRm (Slave) ≥ TAUDnCDRm (Master) + 1

- If TAUDnCDRm (slave) value is greater than TAUDnCDRm (master) value, the counter of the slave channel does not reach 0001_H 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

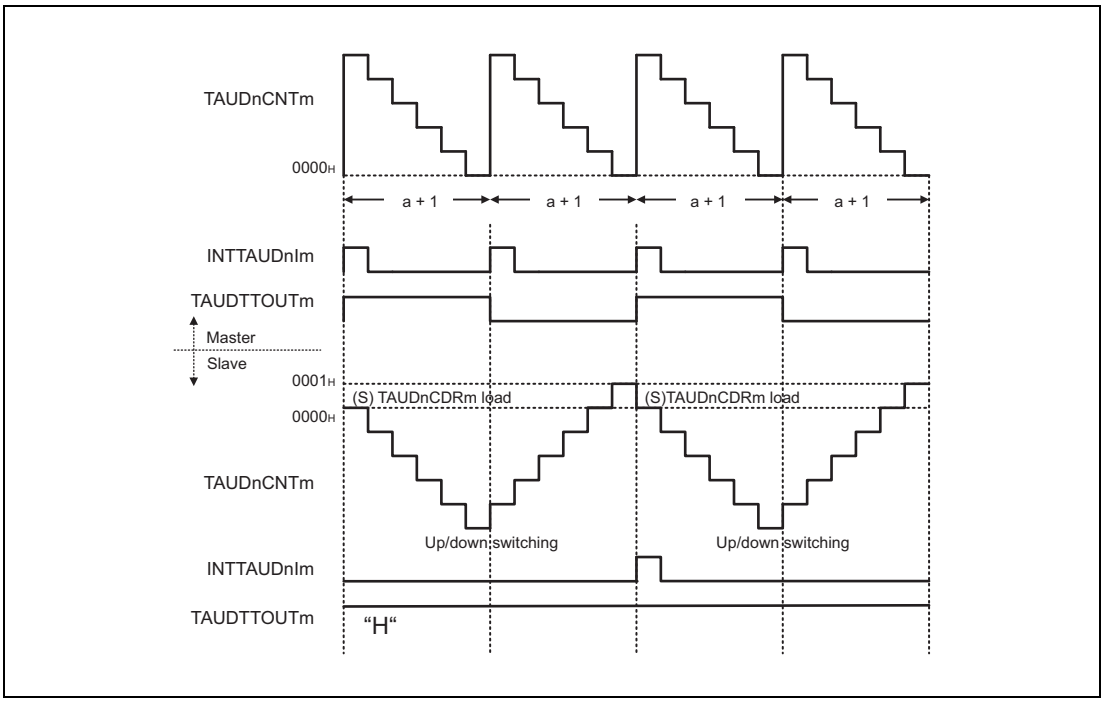


Figure 23.119 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm (slave) = 0000_H, the counter of the slave channel does not reach 0001_H while counting up. TAUDTTOUTm remains high because there is no reset signal to be detected.

23.15.8 Triangle PWM Output Function with Dead Time

23.15.8.1 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 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 23.198, 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).
- The operating mode for slave channel 2 should be set to count-up/-down mode (See **Table 23.202, 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 23.206, 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 23.7, 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 23.7, 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.

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 INTTAUDnlm of the master channel.
- 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 INTTAUDnlm of slave channel 2.

As described in **Table 23.197, Operation of TAUDTTOUTm upon Occurrence of an Interrupt on Slave Channel 2**, the set/reset timing (right 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 23.6, 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 TAUDnTOL.TAUDnTOLm or TAUDnTDL.TAUDnTDLm.

Table 23.197 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 right after interrupt occurs
1	Down	Set right after interrupt occurs
	Up	Set after elapse of dead time

23.15.8.2 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

23.15.8.3 Block Diagram and General Timing Diagram

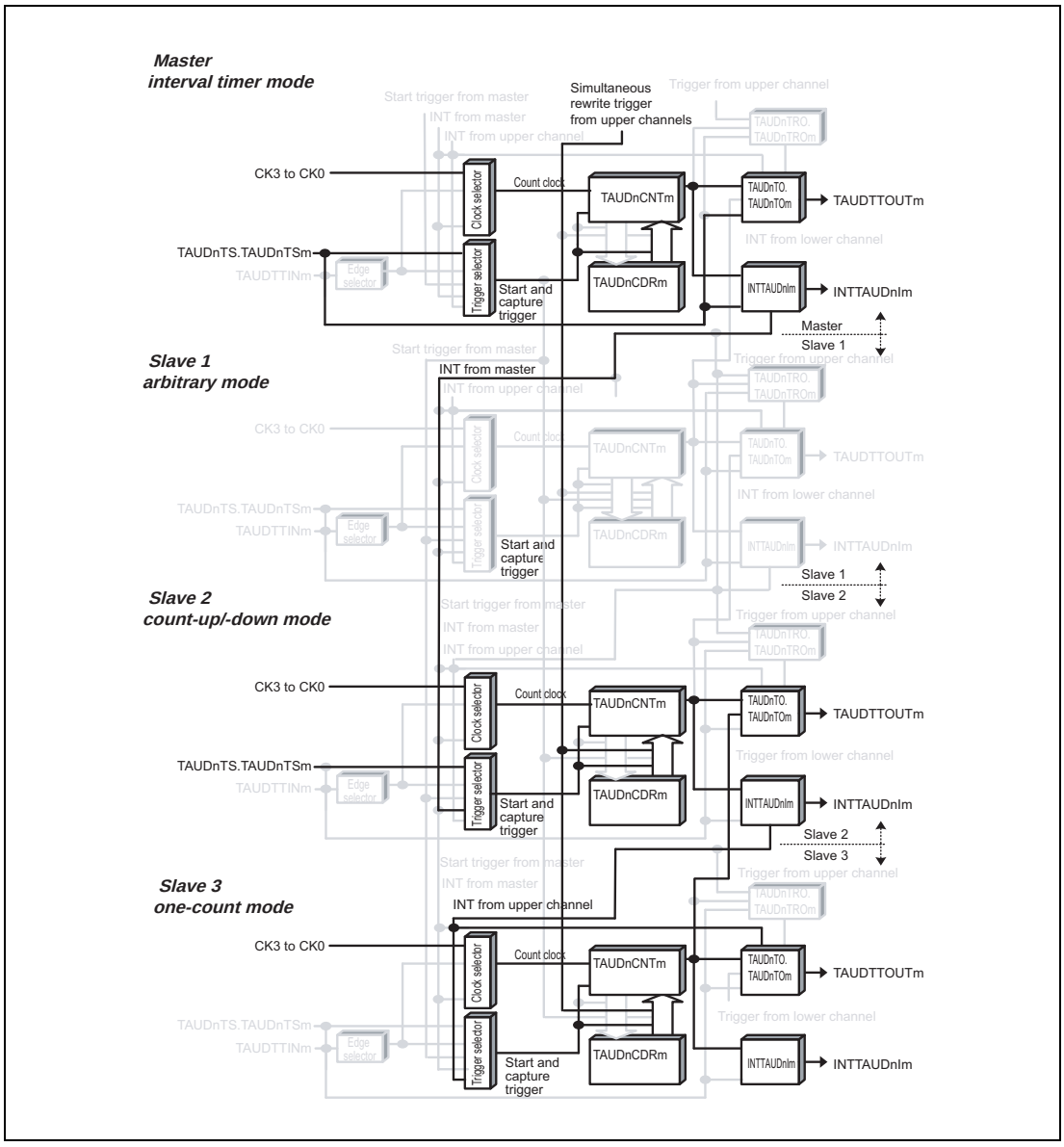


Figure 23.120 Block Diagram of Triangle PWM Output Function with Dead Time

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:
  - INTTAUDnIm generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)
  - TAUDnTDL.TAUDnTDLm = 1
  - Positive logic (TAUDnTOL.TAUDnTOLm = 0)

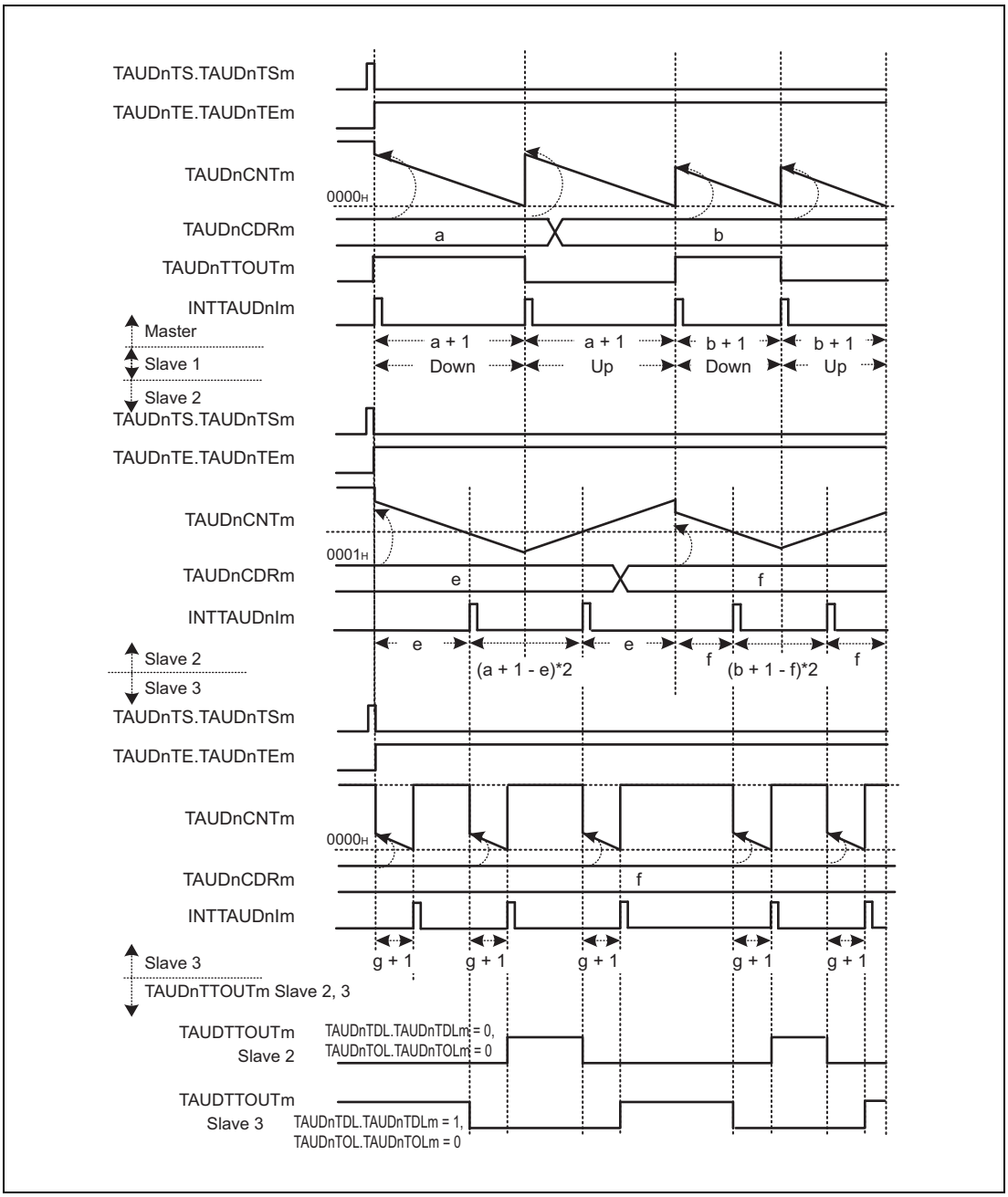


Figure 23.121 General Timing Diagram of Triangle PWM Output Function with Dead Time

23.15.8.4 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 23.198 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 to this bit, 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 23.199 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output the mode for the master channel

Table 23.200 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 23.201 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.

23.15.8.5 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 23.202 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 to this bit, 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 23.203 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channel 2

Table 23.204 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 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 23.205 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.

23.15.8.6 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 23.206 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 to this bit, 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 23.207 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for slave channel 3

Table 23.208 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 23.209 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.

23.15.8.7 Operating Procedure for Triangle PWM Output Function with Dead Time

Table 23.210 Operating Procedure for Triangle PWM Output Function with Dead Time

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.8.4, Register Settings for the Master Channel**. Slave channel 2: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.8.5, Register Settings for Slave Channel 2**. Slave channel 3: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.8.6, 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

23.15.8.8 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram.

- Slave channel 2:
  - Positive logic (TAUDnTDL.TAUDnTDLm = 0)
- Slave channel 3:
  - Negative logic (TAUDnTDL.TAUDnTDLm = 1)

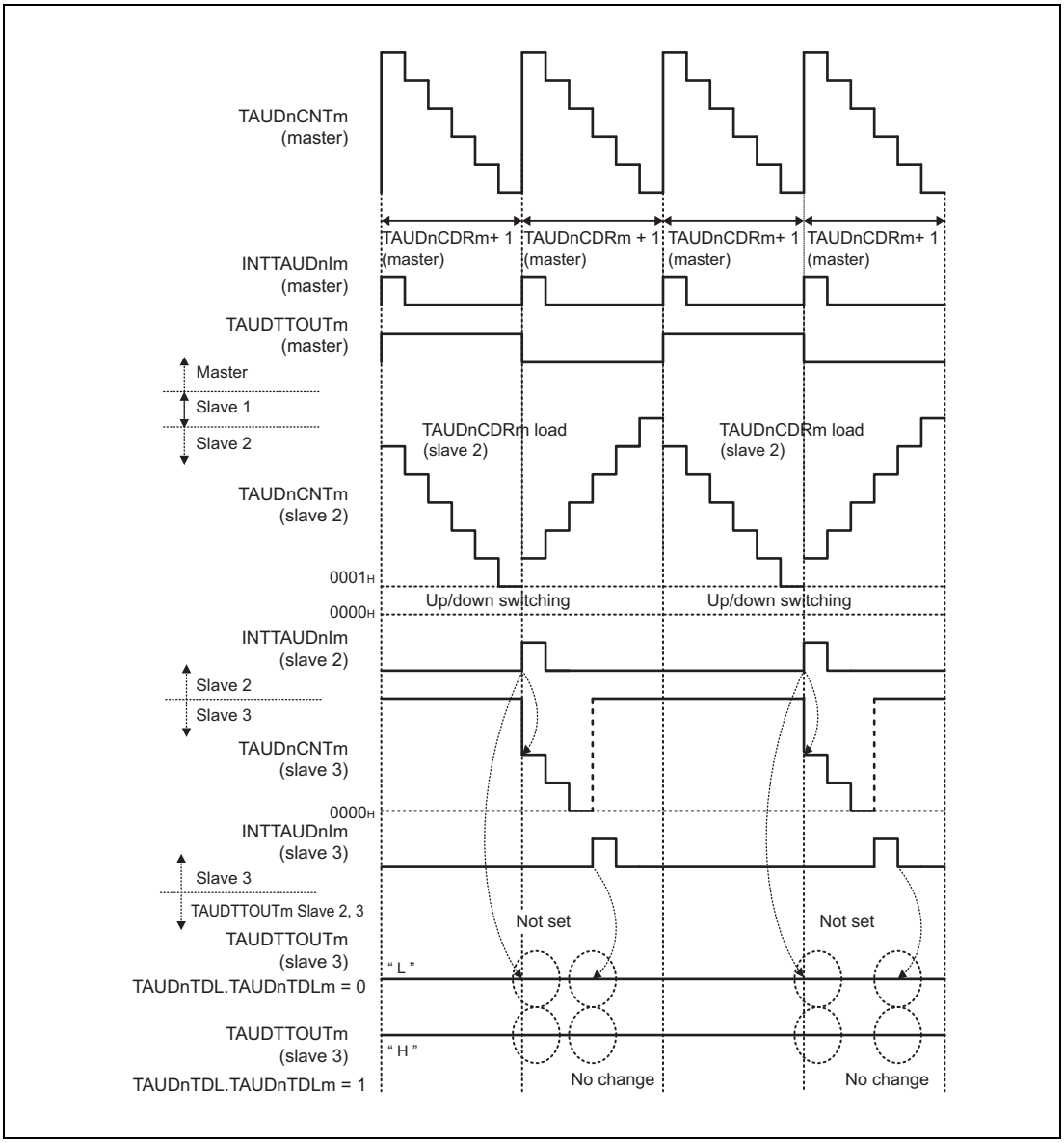


Figure 23.122 TAUDnCDRm (Slave 2) ≥ TAUDnCDRm (Master) + 1

- 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.

- Slave channel 2:
  - Positive logic (TAUDnTDL.TAUDnTDLm = 0)
- Slave channel 3:
  - Negative logic (TAUDnTDL.TAUDnTDLm = 1)

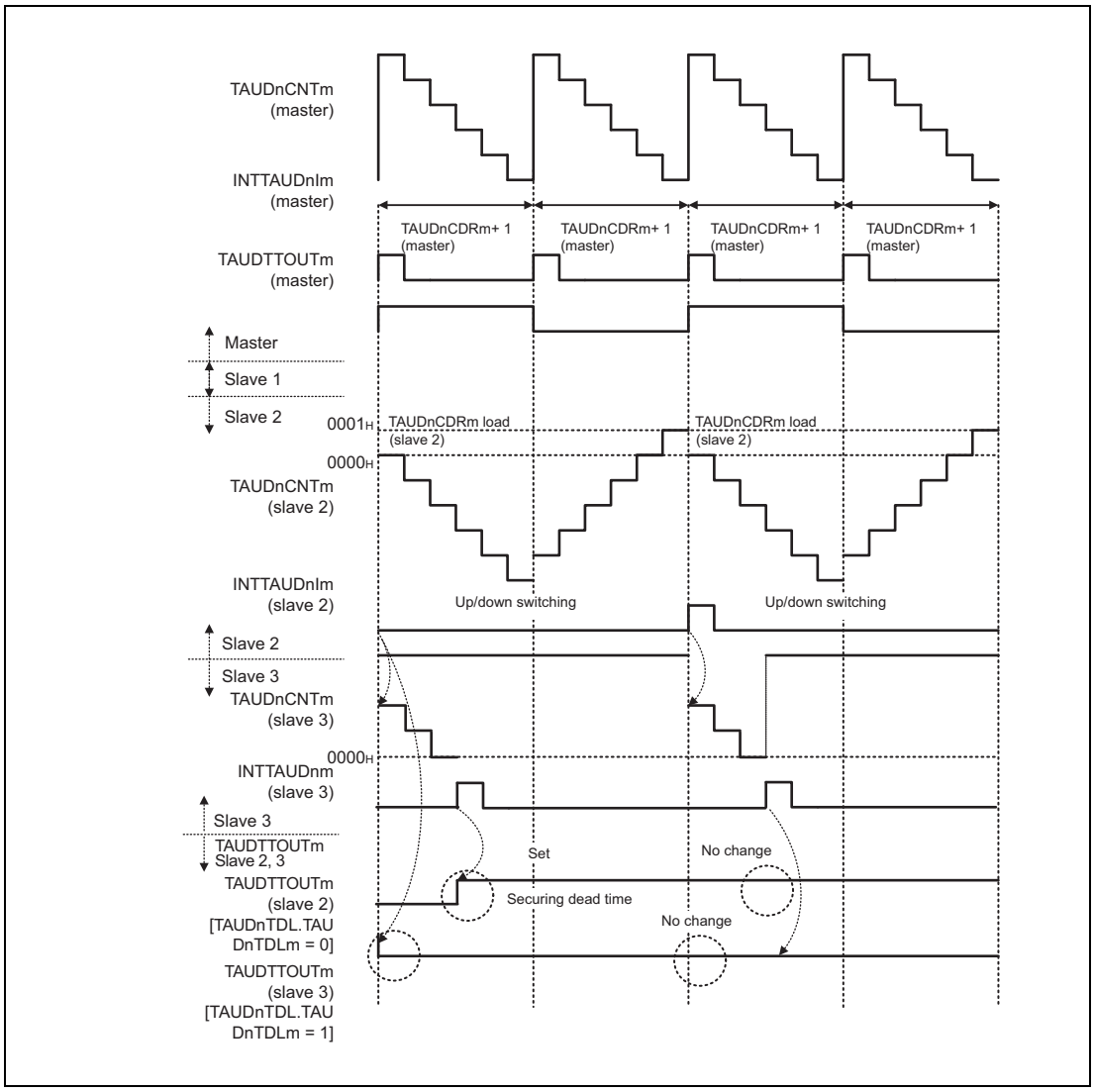


Figure 23.123 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm (slave 2) = 0000_H, the slave channel counter does not reach 0001_H while counting up. Therefore, no INTTAUDnlm 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.TDLm = 1 because no reset conditions are satisfied on that channel.

23.15.9 A/D Conversion Trigger Output Function Type 2

23.15.9.1 Overview

Summary

This function is identical to **Section 23.15.7, 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.

23.15.9.2 Block Diagram and General Timing Diagram

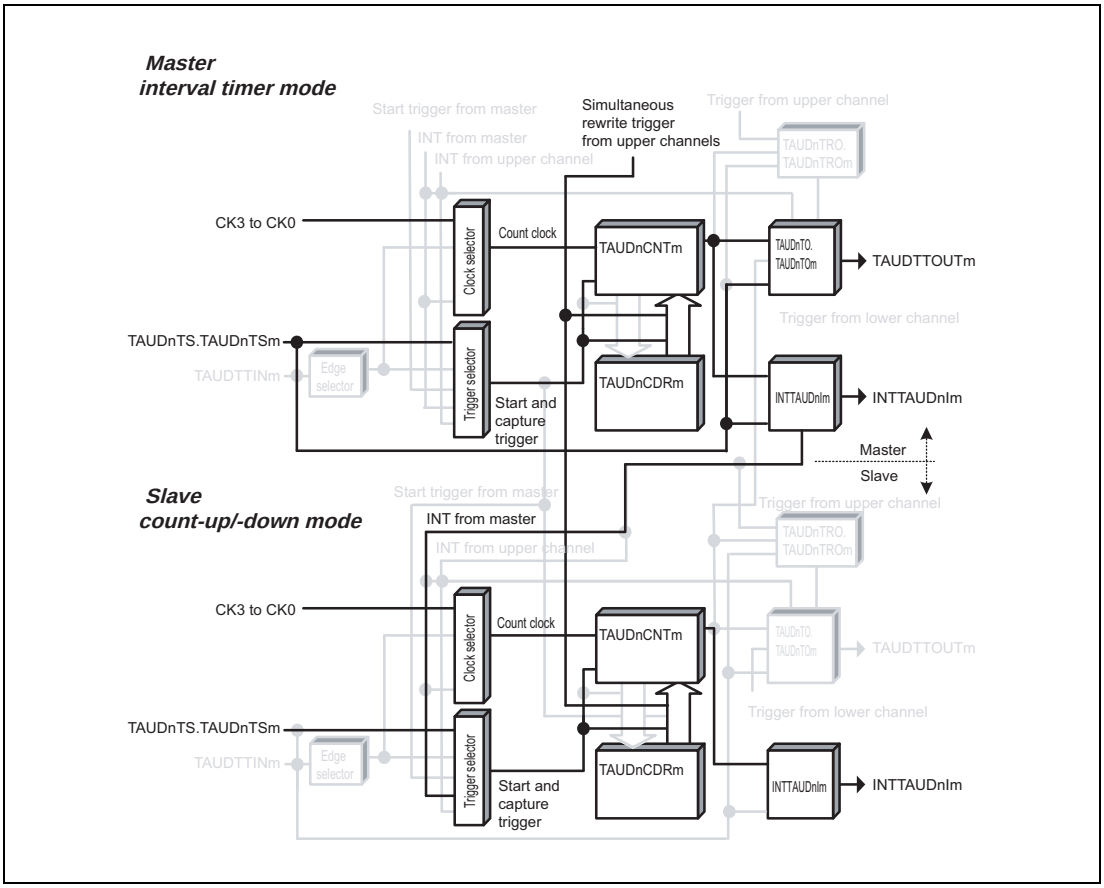


Figure 23.124 Block Diagram of A/D Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

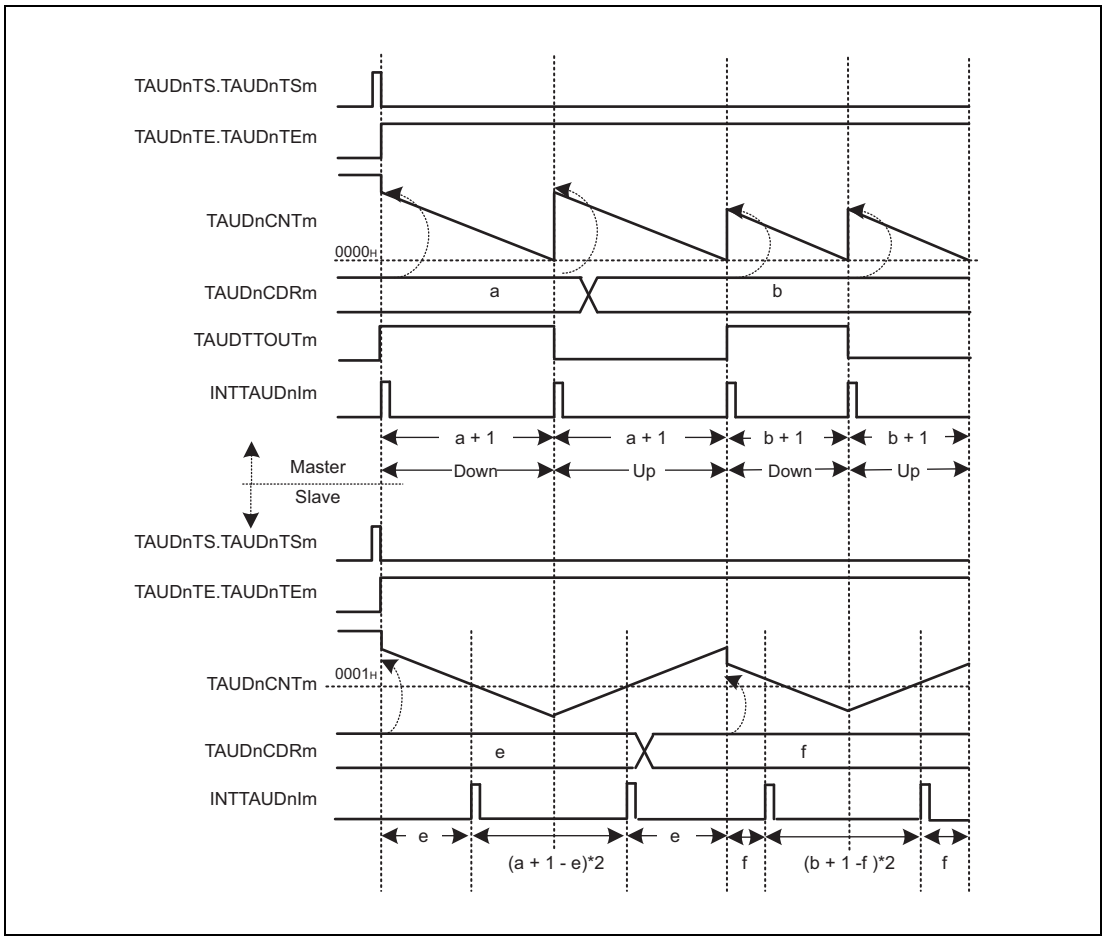


Figure 23.125 General Timing Diagram of A/D Conversion Trigger Output Function Type 2

23.15.10 Interrupt Request Signals Culling Function

23.15.10.1 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 23.15.1, PWM Output Function**)
- Triangle PWM Output Function (See **Section 23.15.7, Triangle PWM Output Function**
- Triangle PWM Output Function with Dead Time
  (See **Section 23.15.8, 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 23.211, 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 23.214, 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 23.6, Simultaneous Rewrite**.

23.15.10.2 Equations

Interrupt division operator = TAUDnCDRm (slave channel)

– One INTTAUDnlm is generated for the INTTAUDnlm count of the master channel defined by TAUDnCDRm (slave channel) + 1.

23.15.10.3 Block Diagram and General Timing Diagram

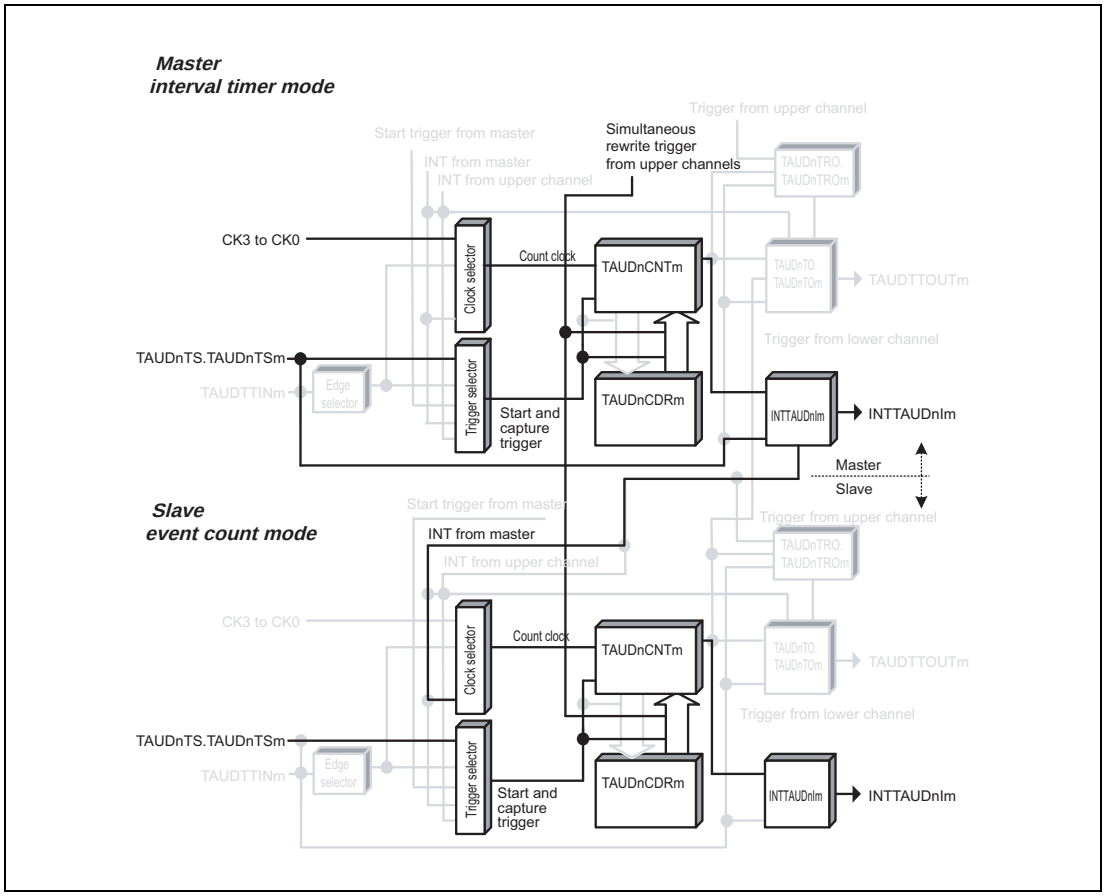


Figure 23.126 Block Diagram of Interrupt Request Signals Culling Function

The following settings apply to the general timing diagram.

Master channel:

- INTTAUDnIm is generated at the beginning of operation. (TAUDnCMORm.TAUDnMD0 = 1)

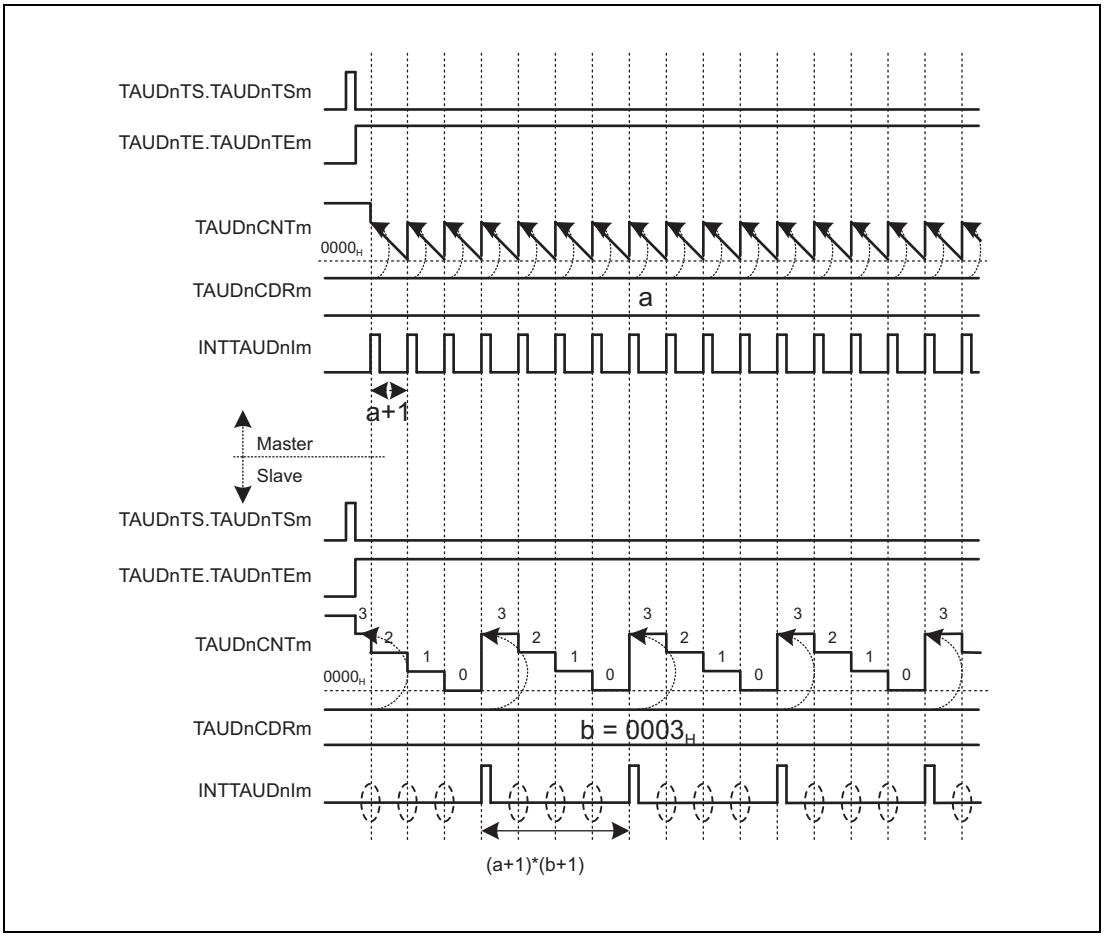


Figure 23.127 General Timing Diagram of Interrupt Request Signals Culling Function

23.15.10.4 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]		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 23.211 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 to this bit, 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 23.212 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 to these bits, 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 23.213 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.
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.

23.15.10.5 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 23.214 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 to this bit, 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 23.215 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 to these bits, 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 23.216 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.
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.

23.15.10.6 Operating Procedure for Interrupt Request Signals Culling Function

Table 23.217 Operating Procedure for Interrupt Request Signals Culling Function

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.10.4, Register Settings for the Master Channel**. Slave channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.15.10.5, Register Settings for the Slave Channel**. 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. • TAUDnCNTm (master) loads TAUDnCDRm value and continues count operation. • TAUDnCNTm of slave channels counts down each time INTTAUDnlm 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.

Restart Operation

23.15.10.7 Specific Timing Diagrams

(1) Interrupt count (master) = interrupt count (slave)

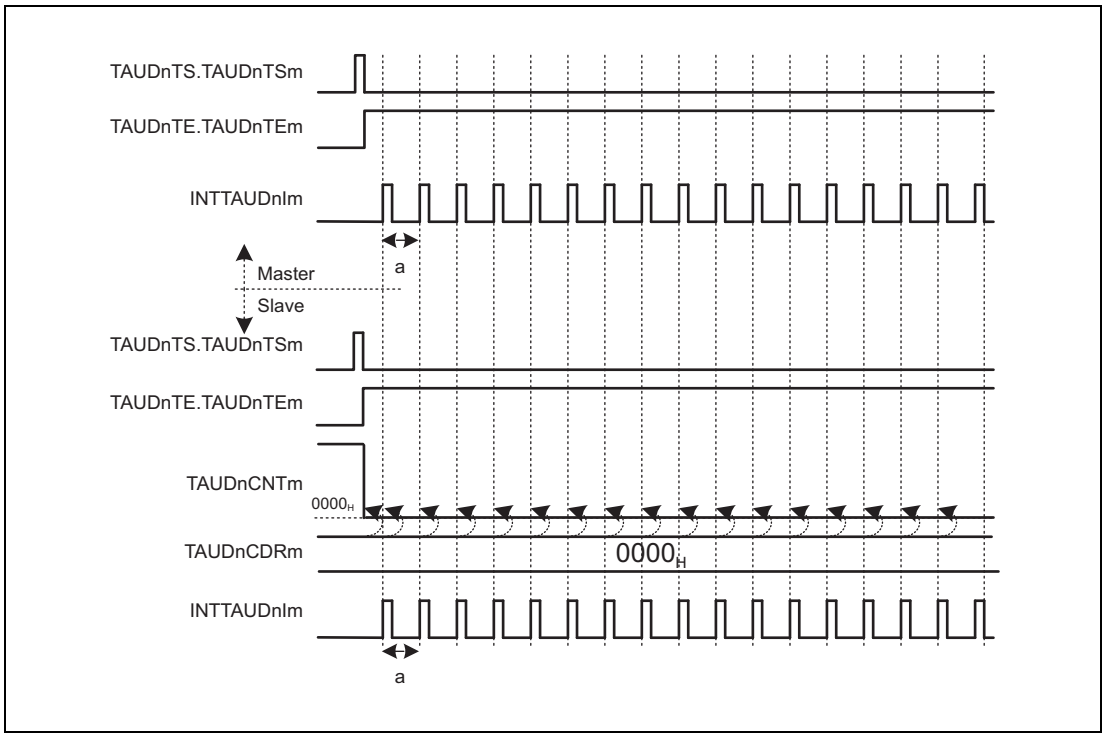


Figure 23.128 TAUDnCDRm (Slave) = 0000_H

- If TAUDnCDRm = 0000_H, the TAUDnCDRm value of the slave channel is loaded into TAUDnCNTm each time INTTAUDnlm 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.

23.16 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.

23.16.1 Non-Complementary Modulation Output Function Type 1

23.16.1.1 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 23.219, 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 23.222, Contents of the TAUDnCMORm Register for Slave Channel 1 of Non-Complementary Modulation Output Function Type 1**, and **Table 23.225, Contents of the TAUDnCMORm Register for Slave Channel 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 23.7, 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 23.7, 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, INTTAUDnlm 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 23.218, 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 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 23.218 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 23.6, 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 with 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.

23.16.1.2 Equations

Slave channels 2 to 7:

Pulse period = [TAUDnCDRm (master) + 1] × count clock cycle

Duty time = [TAUDnCDRm (slave)] × count clock cycle

23.16.1.3 Block Diagram and General Timing Diagram

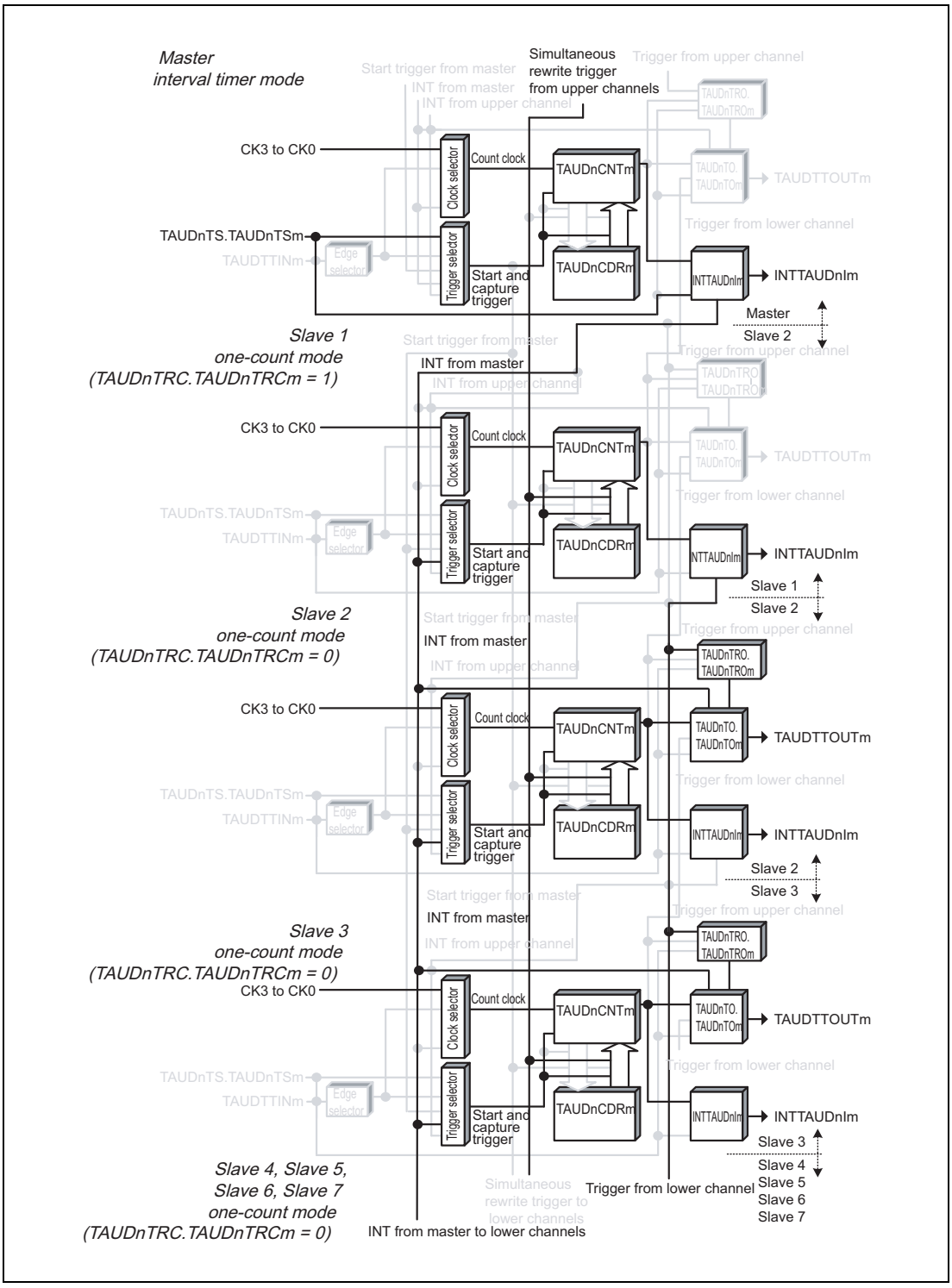


Figure 23.129 Block Diagram of Non-Complementary Modulation Output Function Type 1

The following settings apply to the general timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

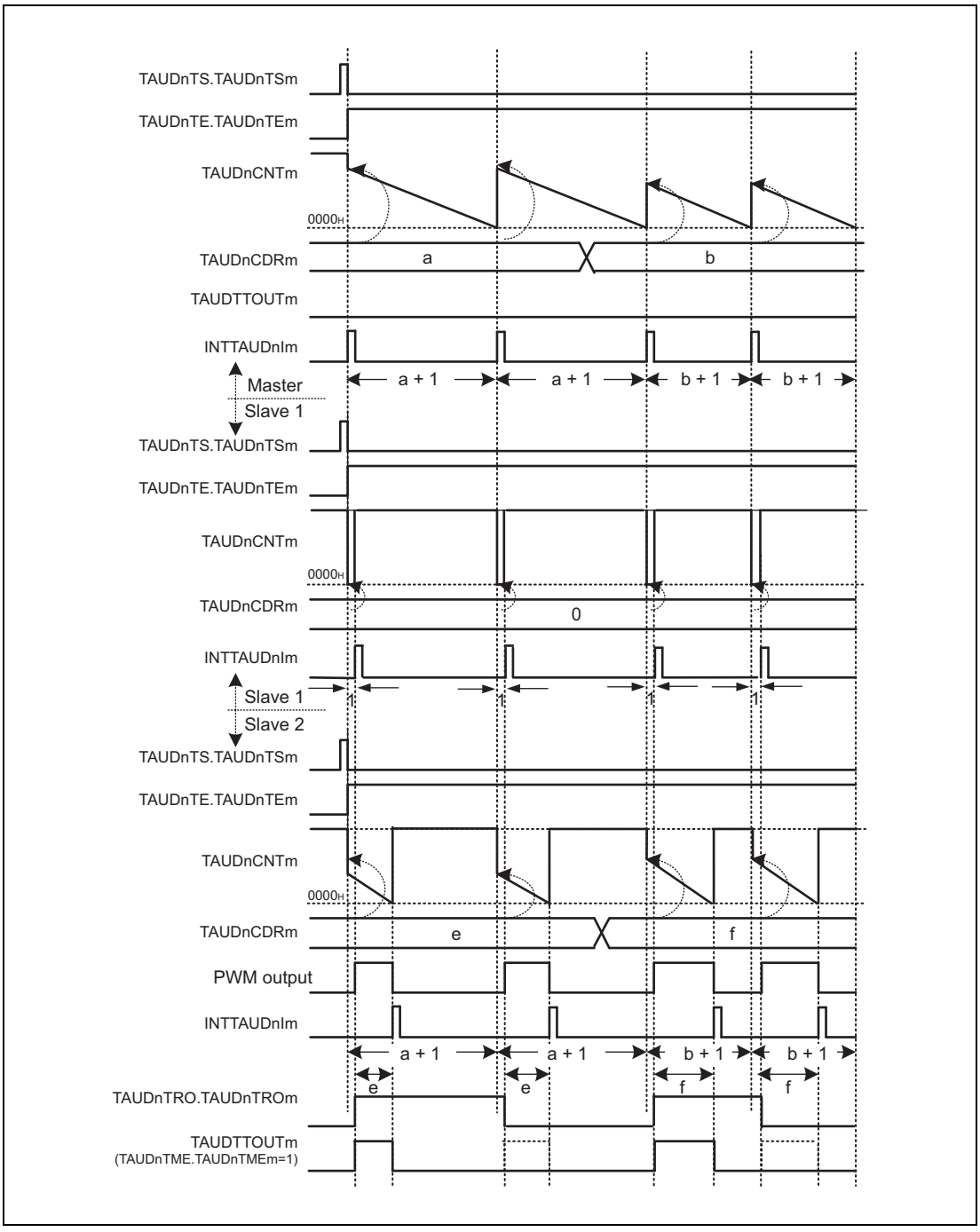


Figure 23.130 General Timing Diagram of Non-Complementary Modulation Output Function Type 1

NOTE

TAUDTTOUTm of slave channel 2 rises with a delay of one clock count after the rise of INTTAUDnIm of the master channel.

23.16.1.4 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]		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 23.219 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 to this bit, write the value after reset.
4 to 1	TAUDnMD[4:1]	0000: Interval timer mode
0	TAUDnMD0	1: INTTAUDnlm 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 23.220 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 to these bits, 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 23.221 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.

NOTE

Use with TAUDnRDS.TAUDnRDSm bit = 1 requires an upper channel higher than the master channel that operates with **Section 23.14.1, 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

23.16.1.5 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 23.222 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 to this bit, 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 23.223 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 to these bits, 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 23.224 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.

23.16.1.6 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]		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 23.225 Contents of the TAUDnCMORm Register for Slave Channel 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 to this bit, 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 23.226 Contents of the TAUDnCMURm Register for Slave Channel 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 to these bits, 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 23.227 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 23.228 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.

23.16.1.7 Operating Procedure for Non-Complementary Modulation Output Function Type 1

Table 23.229 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (1/2)

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.16.1.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.16.1.5, Register Settings for Slave Channel 1**. Slave channels 2 to 7: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.16.1.6, 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 23.229 Operating Procedure for Non-Complementary Modulation Output Function Type 1 (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.	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 is reloaded into TAUDnCNTm of master channel 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.

Restart Operation

23.16.1.8 Specific Timing Diagrams

The following settings apply to the specific timing diagram.

- Slave channels 2 to 7: Positive logic (TAUDnTOL.TAUDnTOLm = 0)

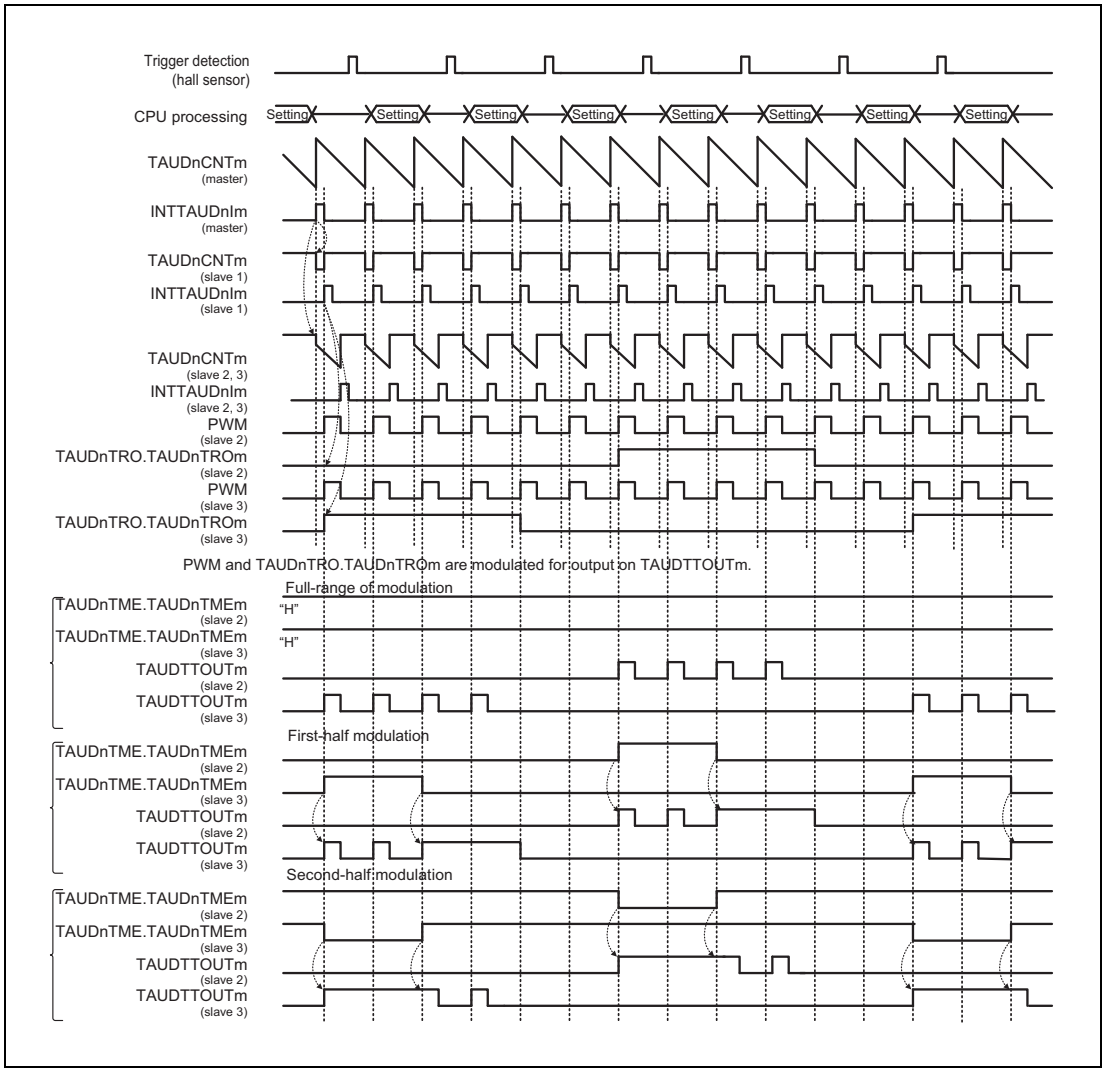


Figure 23.131 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 1

The above timing diagram shows how full modulation, fist-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.

23.16.2 Non-Complementary Modulation Output Function Type 2

23.16.2.1 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 23.231, 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 23.235, 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 23.238, Contents of the TAUDnCMORm Register for Slave Channel 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 23.7, Channel Output Modes**.)
- This function does not use TAUDTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 23.7, 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 23.7, 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 for the (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), 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 detecting an interrupt 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 23.230, 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 23.230 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 23.6, 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.

23.16.2.2 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

23.16.2.3 Block Diagram and General Timing Diagram

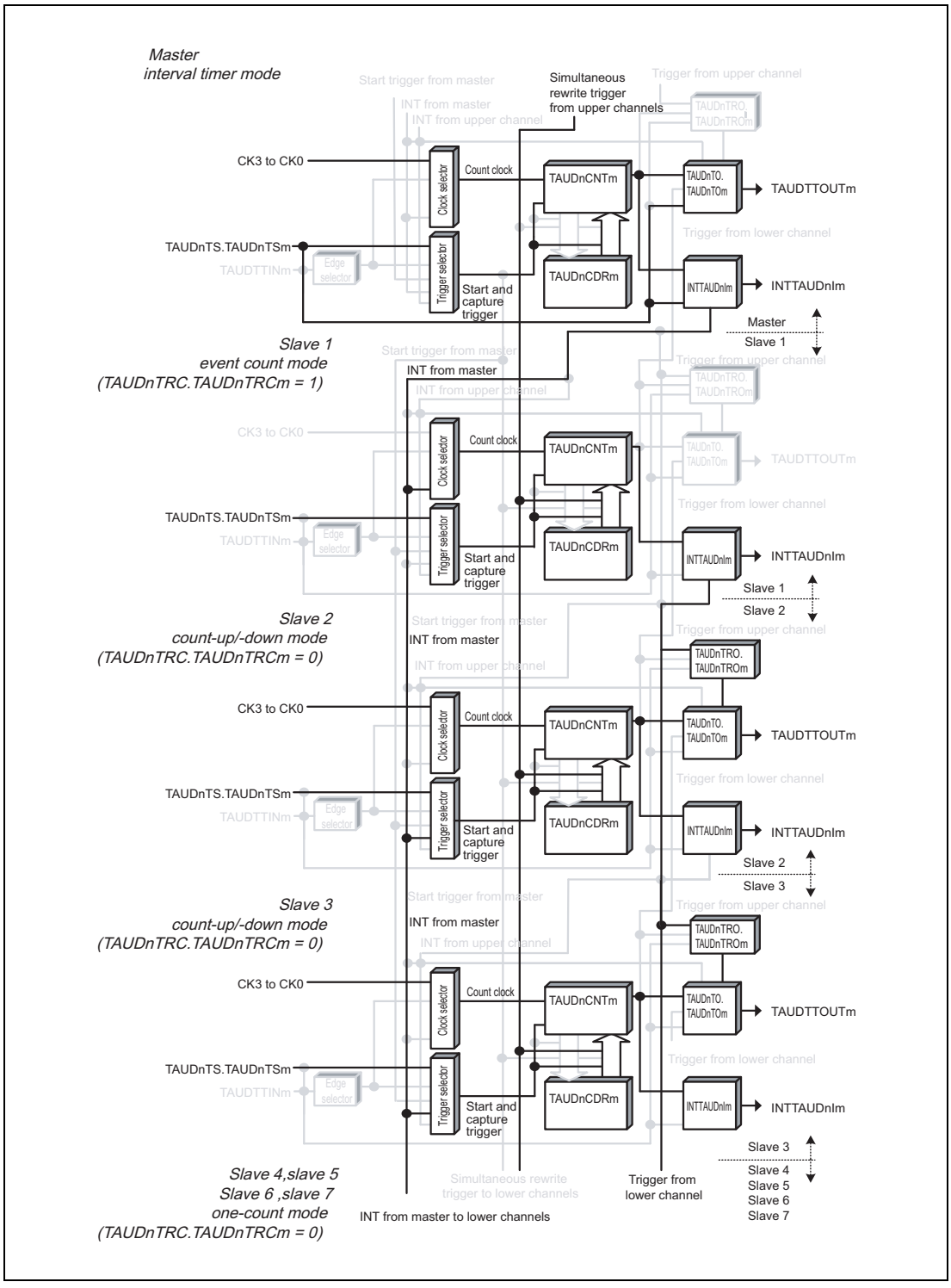


Figure 23.132 Block Diagram of Non-Complementary Modulation Output Function Type 2

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)

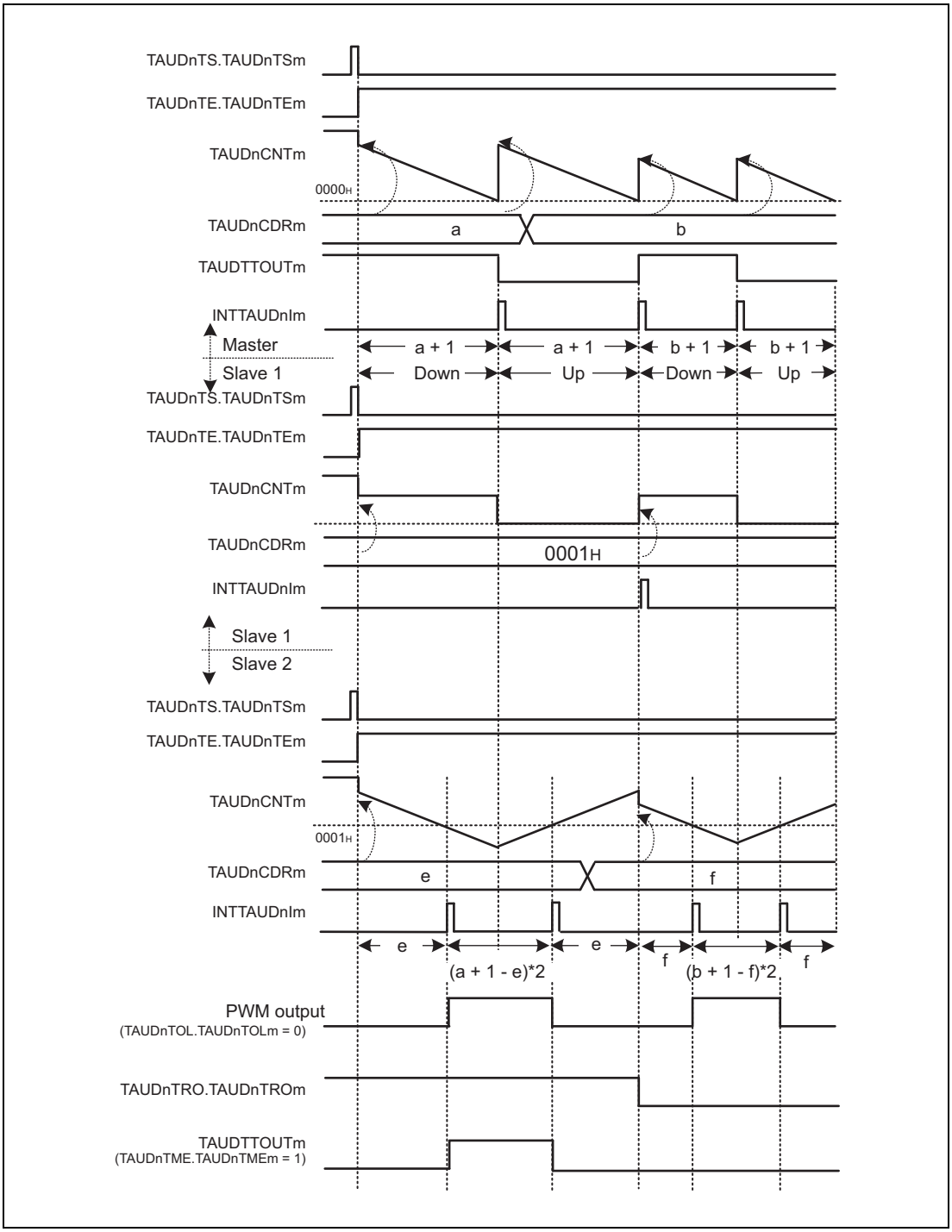


Figure 23.133 General Timing Diagram of Non-Complementary Modulation Output Function Type 2

23.16.2.4 Register Settings 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 23.231 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 to this bit, 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: INTTAUDnlm 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 23.232 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 to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the master channel

Table 23.233 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 23.234 Simultaneous Rewrite Settings for the Master Channel of Non-Complementary Modulation Output Function Type 2

Bit Name	Setting
TAUDnRDE.TAUDnRDEm	1: Disables 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.

23.16.2.5 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 23.235 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 to this bit, 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 23.236 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 to these bits, 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 23.237 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.

23.16.2.6 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]		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 23.238 Contents of the TAUDnCMORm Register for Slave Channel 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 to this bit, 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 23.239 Contents of the TAUDnCMURm Register for Slave Channel 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 to these bits, 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 23.240 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 23.241 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.

23.16.2.7 Operating Procedure for Non-Complementary Modulation Output Function Type 2

Table 23.242 Operating Procedure for Non-Complementary Modulation Output Function Type 2 (1/2)

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.16.2.4, Register Settings the Master Channel**. Slave channel 1: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.16.2.5, Register Settings for Slave Channel 1**. Slave channels 2 to 7: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.16.2.6, 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 23.242 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, but 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

23.16.2.8 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)

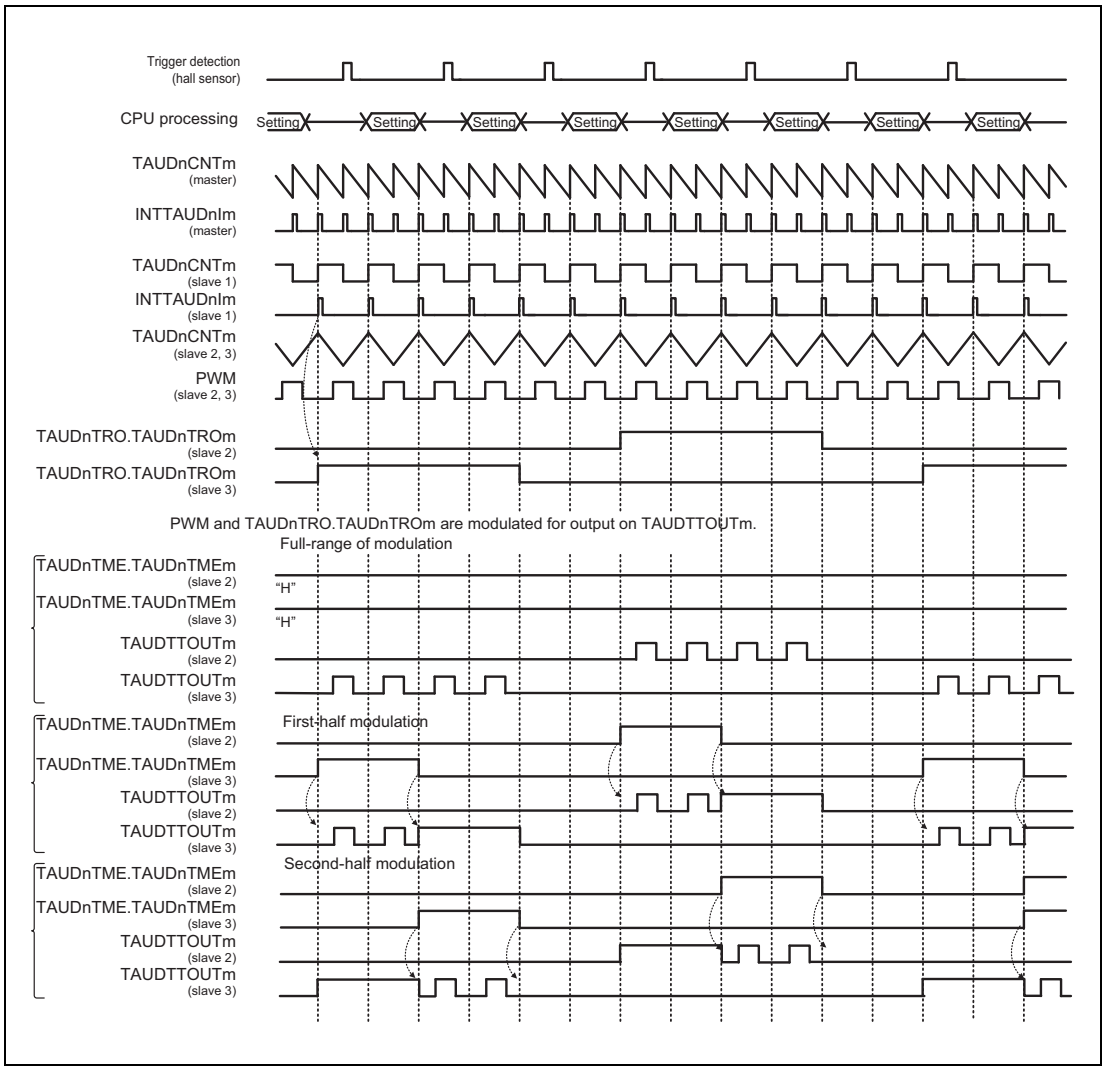


Figure 23.134 Specific Timing Diagram of Non-Complementary Modulation Output Function Type 2

The above timing diagram shows how full modulation, fist-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.

23.16.3 Complementary Modulation Output Function

23.16.3.1 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 23.244, Contents of the TAUDnCMORm Register for the Master Channel of the Non-Complementary Modulation Output Function**).
- The operating mode for slave channel 1 should be set to event count mode (See **Table 23.248, Contents of the TAUDnCMORm Register for Slave Channel 1 of the Non-Complementary Modulation Output Function**).
- The operating mode for slave channels 2, 4 and 6 should be set to count-up/-down mode (See **Table 23.251, Contents of the TAUDnCMORm Register for Slave Channel 2, 4, and 6 of the Non-Complementary Modulation Output Function**).
- The operating mode for slave channels 3, 5 and 7 should be set to one-count mode (See **Table 23.255, 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 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 **23.7, Channel Output Modes**).
- This function does not use TAUDTTOUTm of slave channel 1 but TAUDnTRC.TAUDnTRCm should be set to 1 (See **Section 23.7, 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 23.7, 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. 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 23.243, 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.TAUDnTROmis 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 23.243 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 23.6, Simultaneous Rewrite**.

23.16.3.2 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.

23.16.3.3 Block Diagram and General Timing Diagram

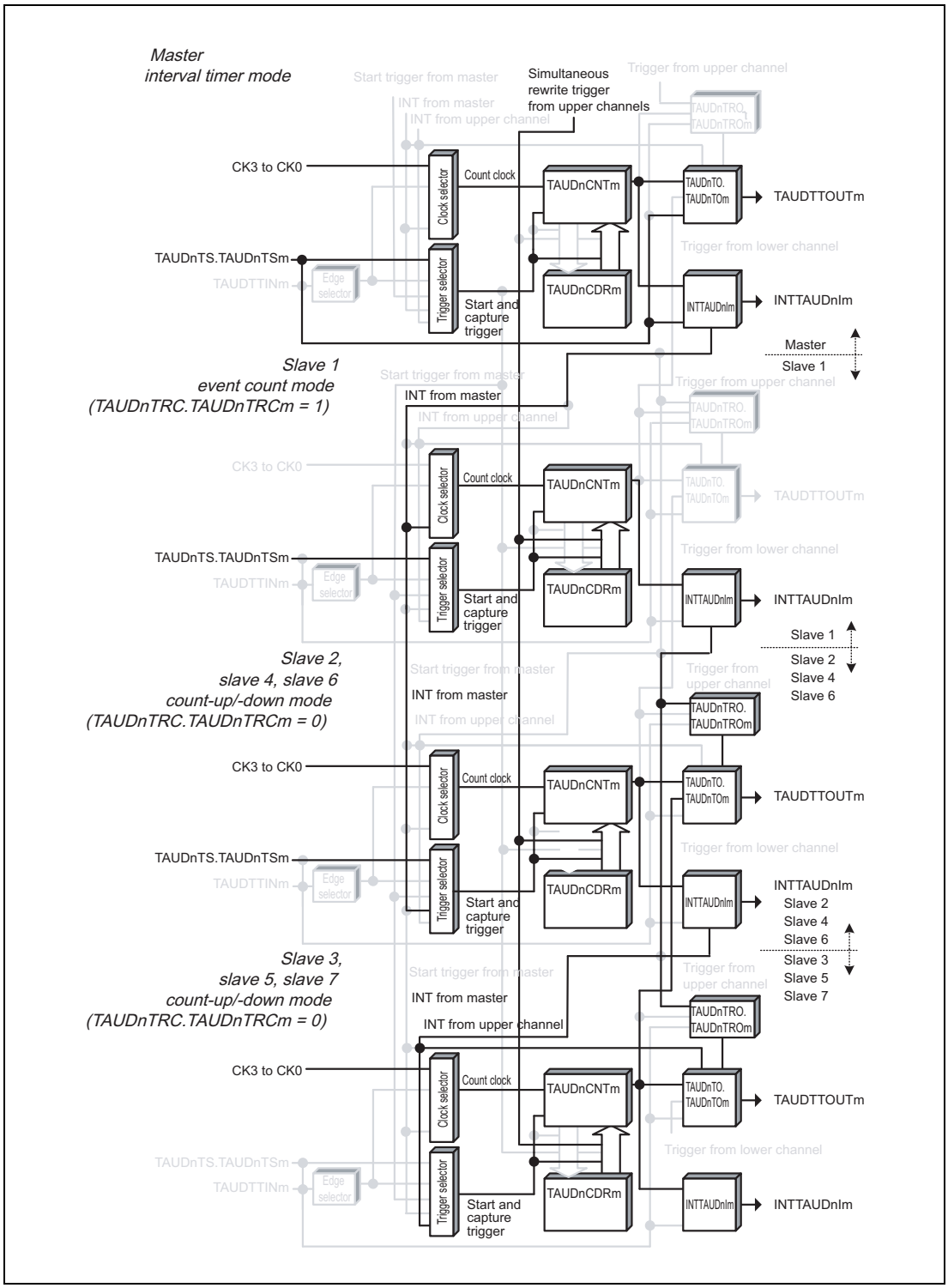


Figure 23.135 Block Diagram of Complementary Modulation Output Function

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)

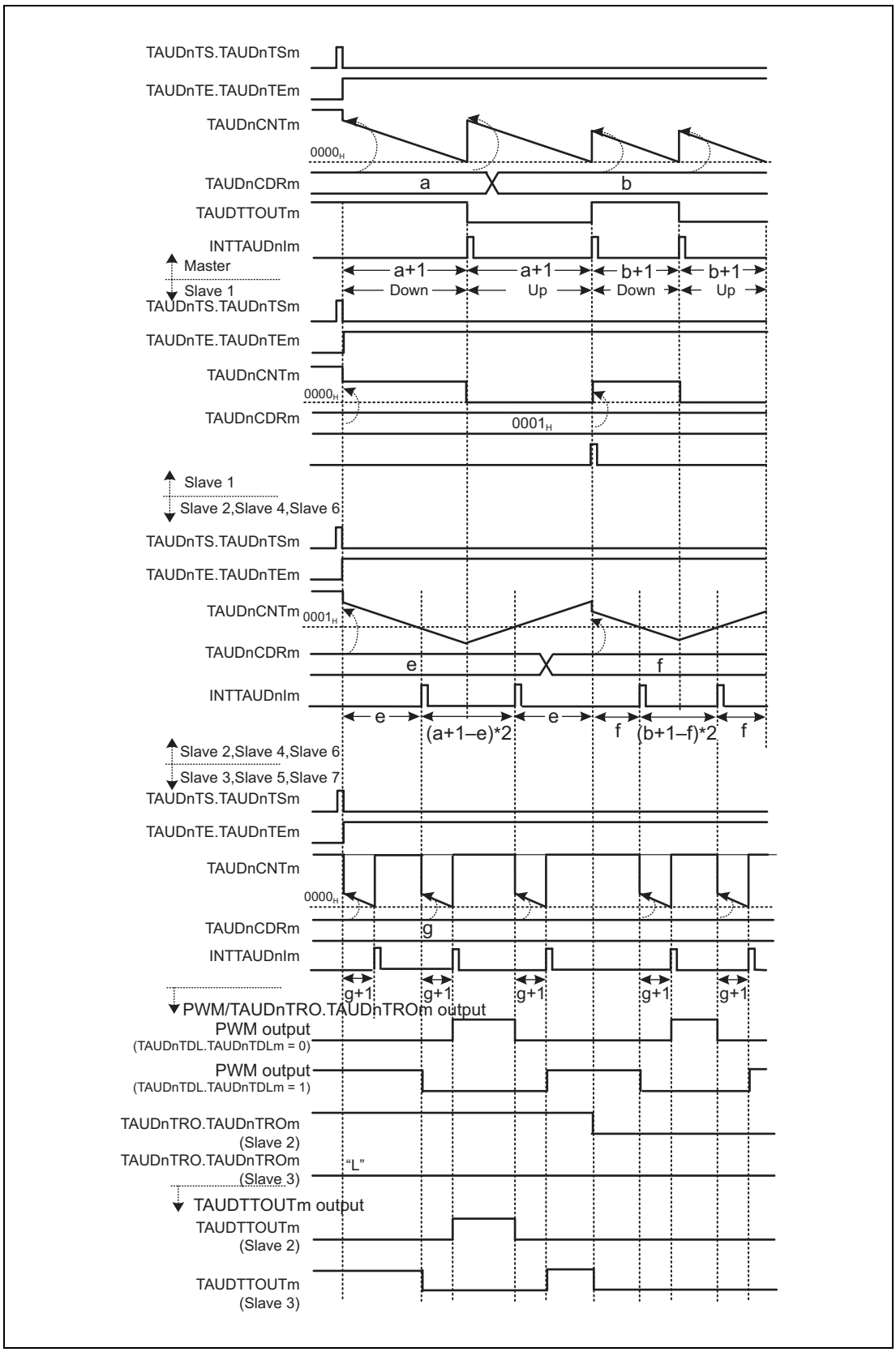


Figure 23.136 General Timing Diagram of Complementary Modulation Output Function

23.16.3.4 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 23.244 Contents of the TAUDnCMORm Register for the Master Channel of the Non-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 to this bit, 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 23.245 Contents of the TAUDnCMURm Register for the Master Channel of the Non-Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUDnTIS[1:0]	00: Unused. Set to 00.

(3) Channel output mode for the master channel

Table 23.246 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 23.247 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.

NOTE

If TAUDnRDS.TAUDnRDSm = 1, it is necessary for an upper channel higher than the master channel to generate a simultaneous rewrite trigger signal.

23.16.3.5 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]		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 23.248 Contents of the TAUDnCMORm Register for Slave Channel 1 of the Non-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 to this bit, 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 23.249 Contents of the TAUDnCMURm Register for Slave Channel 1 of the Non-Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 23.250 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.

23.16.3.6 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 23.251 Contents of the TAUDnCMORm Register for Slave Channel 2, 4, and 6 of the Non-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 to this bit, 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 23.252 Contents of the TAUDnCMURm Register for Slave Channel 2, 4, and 6 of the Non-Complementary Modulation Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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 23.253 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 23.254 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.

23.16.3.7 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 23.255 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 to this bit, 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 23.256 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 to these bits, 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 23.257 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 23.258 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.

23.16.3.8 Operating Procedure for Complementary Modulation Output Function

Table 23.259 Operating Procedure for Complementary Modulation Output Function (1/2)

	Operation	TAUDn Status
Initial Channel Setting	Master channel: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.16.3.4, Register Settings for the Master Channel**. Slave channel 1: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.16.3.5, Register Settings for Slave Channel 1**. Slave channels 2, 4, and 6: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.16.3.6, Register settings for slave channels 2, 4, and 6**. Slave channels 3, 5, and 7: Set TAUDnCMORm/ TAUDnCMURm register and the channel output mode as described in **Section 23.16.3.7, 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 23.259 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, but 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.

23.16.3.9 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)

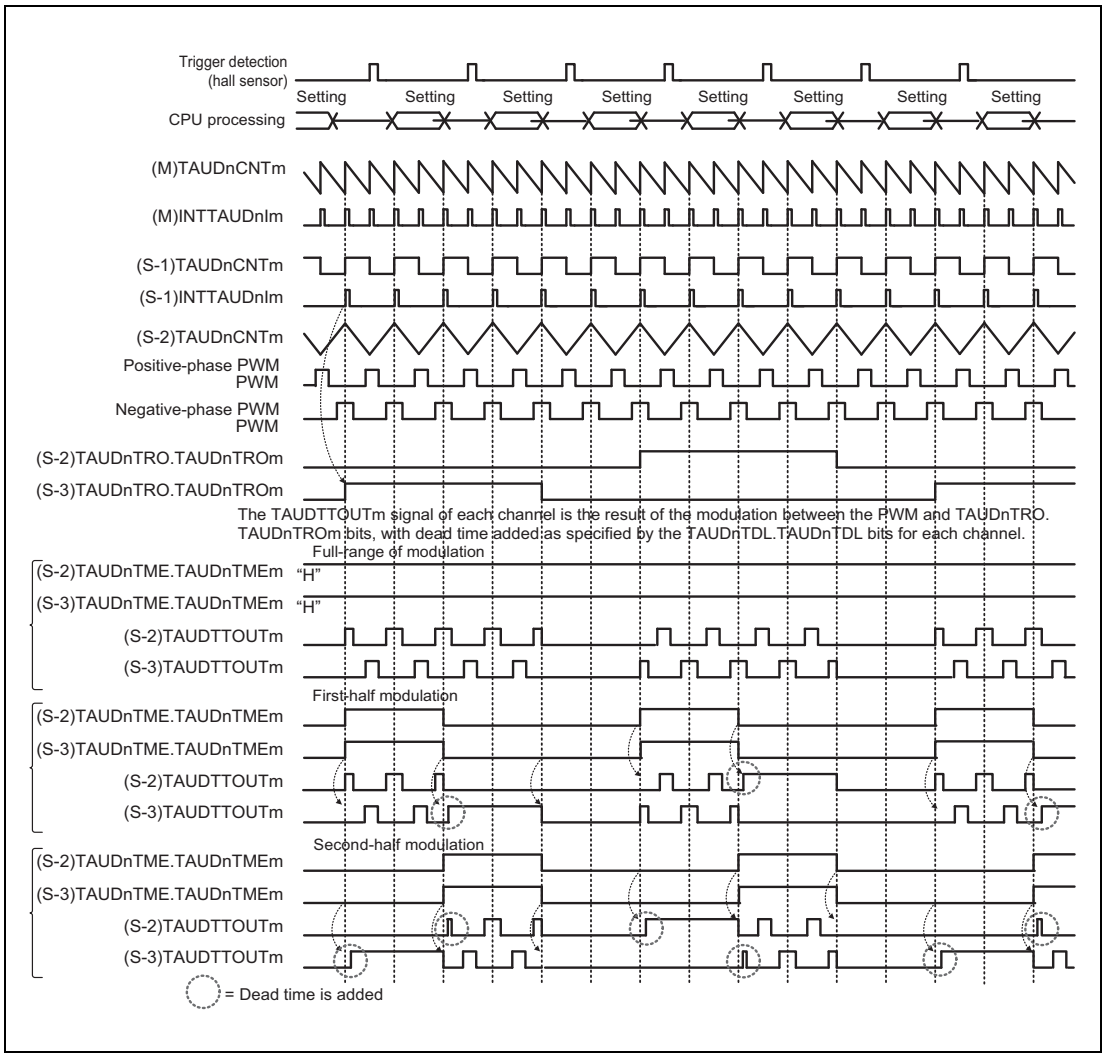


Figure 23.137 Specific Timing Diagram of Complementary Modulation Output Function

The above timing diagram shows how full modulation, fist-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 24 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 all RH850/F1L specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the TAUJ.

24.1 Features of RH850/F1L TAUJ

24.1.1 Number of Units

This microcontroller has the following number of TAUJ units.

Table 24.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	1	2	2	2
Name	TAUJn (n = 0)	TAUJn (n = 0)	TAUJn (n = 0)	TAUJn (n = 0, 1)	TAUJn (n = 0, 1)	TAUJn (n = 0, 1)

TAUJn has the following number of channels of timers.

Table 24.2 Unit Configurations and Channels

Unit Name (Channel Name) TAUJn	Number of Channels per Unit	RH850/F1L 48 pins (4 ch)	RH850/F1L 64 pins (4 ch)	RH850/F1L 80 pins (4 ch)	RH850/F1L 100 pins (8 ch)	RH850/F1L 144 pins (8 ch)	RH850/F1L 176 pins (8 ch)
TAUJ0	4	√	√	√	√	√	√
TAUJ1	4	—	—	—	√	√	√

Table 24.3 Index

Index	Meaning
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.

24.1.2 Register Base Address

TAUJn base addresses are listed in the following table.

TAUJn register addresses are given as offsets from the base addresses.

Table 24.4 Register Base Address

Base Address Name	Base Address
<TAUJ0_base>	FFE5 0000_H
<TAUJ1_base>	FFE5 1000_H

24.1.3 Clock Supply

The TAUJn clock supply is shown in the following table.

Table 24.5 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
TAUJ0	PCLK	CKSCLK_ATAUJ
TAUJ1	PCLK	CKSCLK_IPERI1

24.1.4 Interrupt Requests

TAUJn interrupt requests are listed in the following table.

Table 24.6 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
TAUJ0			
INTTAUJ0I0	Channel 0 interrupt	72	21 (Channels 0 to 7)
INTTAUJ0I1	Channel 1 interrupt	73	16 (Channels 8 to 15)
INTTAUJ0I2	Channel 2 interrupt	74	17 (Channels 8 to 15)
INTTAUJ0I3	Channel 3 interrupt	75	22 (Channels 0 to 7)
TAUJ1			
INTTAUJ1I0	Channel 0 interrupt	160	46 (Channels 0 to 7)
INTTAUJ1I1	Channel 1 interrupt	161	36 (Channels 8 to 15)
INTTAUJ1I2	Channel 2 interrupt	162	47 (Channels 0 to 7)
INTTAUJ1I3	Channel 3 interrupt	163	37 (Channels 8 to 15)

24.1.5 Reset Sources

TAUJn reset sources are listed in the following table. TAUJn is initialized by these reset sources.

Table 24.7 Reset Sources

Unit Name	Reset Source
TAUJ0	All reset sources except the transition to DEEPSTOP mode (AWORES)
TAUJ1	All reset sources (ISORES)

24.1.6 External Input/Output Signals

External input/output signals of TAUJn are listed below.

Table 24.8 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
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 24.1.8, TAUJ0 Input Selection**.

24.1.7 Internal Input/Output Signals

The internal input/output signals of TAUJn are listed below.

Table 24.9 Internal Input/Output Signals

Unit Signal Name	Outline	Connected to
TAUJnTSSTm	Simultaneous channel start trigger input	PIC

24.1.8 TAUJ0 Input Selection

The 1-Hz pulse output (RTCA0OUT) from RTCA0 can be input to TAUJTTIN2 and TAUJTTIN3 as shown in the following figure.

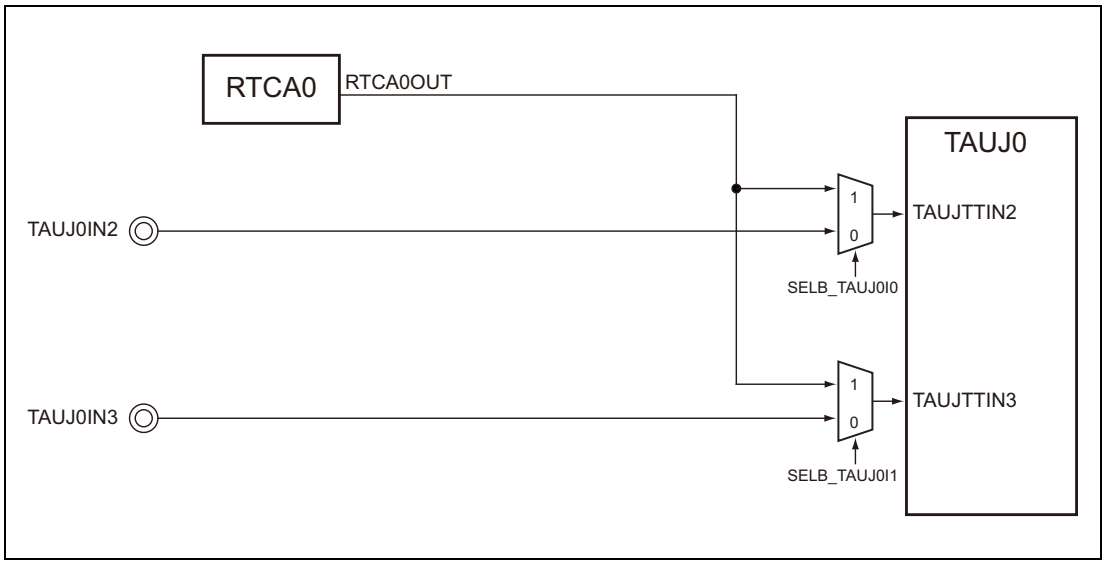


Figure 24.1 Selection of Signals Input to TAUJ0

The following table shows how to select signals input to the TAUJ.

Table 24.10 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

24.1.8.1 SELB_TAUJ0I — TAUJTTINm Input Signal Selection Register

This register selects the input signals to several TAUJ0 inputs.

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

Address: FFBC 0100_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 24.11 SELB_TAUJ0I Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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

24.2 Overview

24.2.1 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), i.e. multiple channels depend on each other with one master and one or more slave channels.

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.

24.2.2 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 use any channel independent 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

Depending 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.

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 24.12 Functional List of TAUJ Operations

Operation Function	Example
Independent Channel Operation Functions	**Section 24.12**
Interval Timer Function	**Section 24.12.1**
TAUJTTINm Input Interval Timer Function	**Section 24.12.2**
TAUJTTINm Input Pulse Interval Measurement Function	**Section 24.12.3**
TAUJTTINm Input Signal Width Measurement Function	**Section 24.12.4**
TAUJTTINm Input Position Detection Function	**Section 24.12.5**
TAUJTTINm Input Period Count Detection Function	**Section 24.12.6**
Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)	**Section 24.12.7**
Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)	**Section 24.12.8**
Synchronous Channel Operation Functions	**Section 24.13**
PWM Output Function	**Section 24.13.1**

24.2.4 TAUJ I/O and Interrupt Request Signals

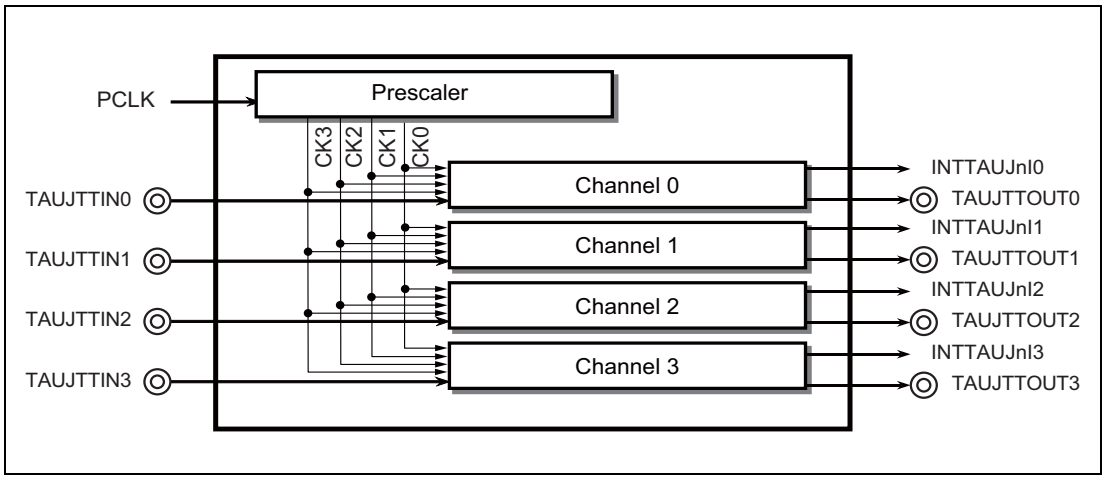


Figure 24.2 TAUJ I/O and Interrupt Request Signals

24.2.5 Block Diagram

The following figure shows the main components of the TAUJ.

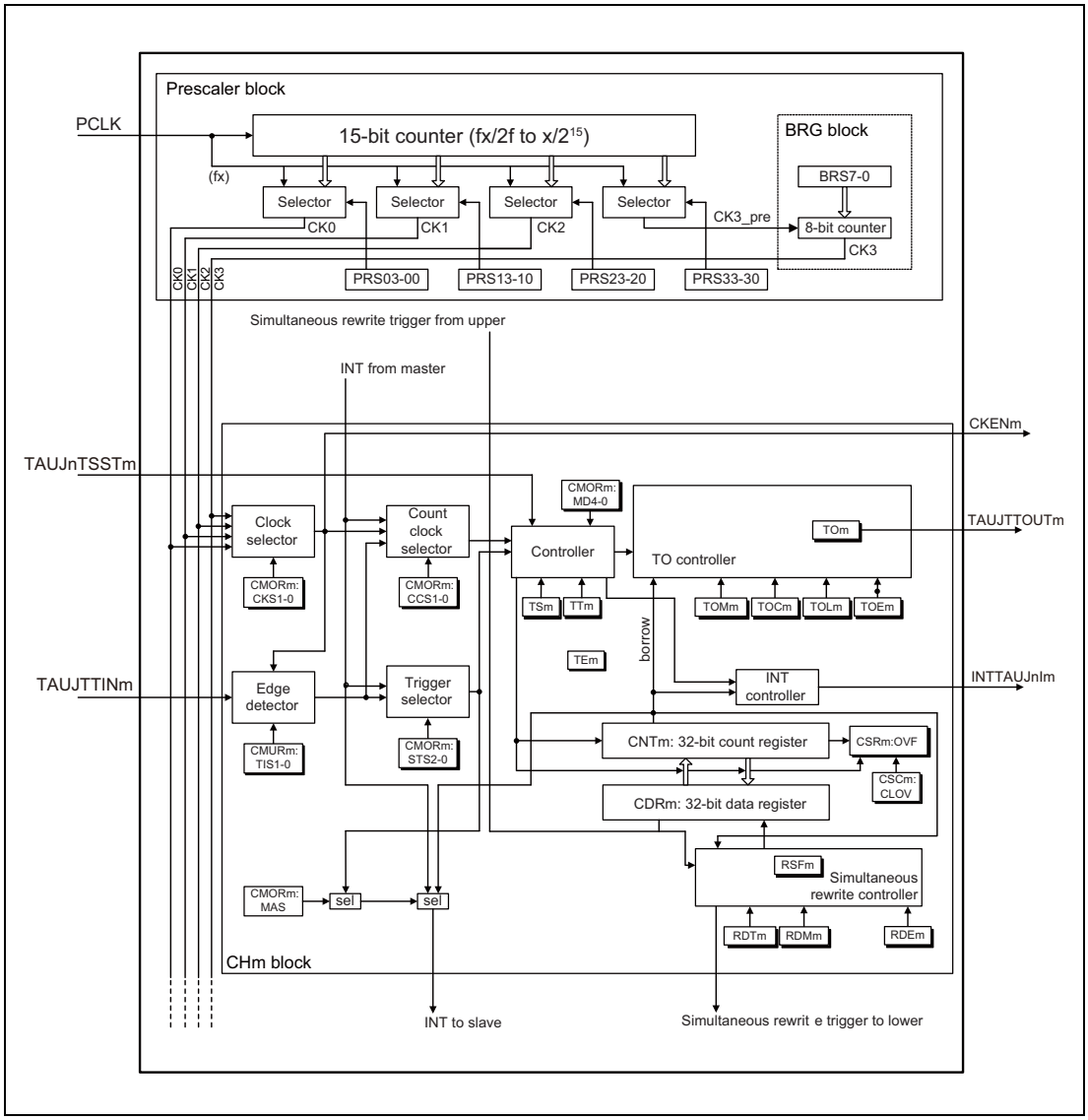


Figure 24.3 Block Diagram of the TAUJ

The prefix “TAUJn” has been omitted from the register names for the sake of clarity in the above figure.

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 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 an additional division factor that is not a power of 2.

Count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source.

- One of the prescaler outputs CK0 to CK3 (selected by the clock selector)
- Valid edge of the TAUJTTINm input signal

Controller

The controller controls the main operations of the counter.

- 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 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.

24.3 Registers

24.3.1 List of Registers

TAUJ registers are listed in the following table.

For details about <TAUJn_base>, see **Section 24.1.2, Register Base Address**.

Table 24.13 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> + $A8_H$

24.3.2 Details of TAUJn Prescaler Registers

24.3.2.1 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: Readable/writable 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 24.14 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 24.14 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 24.14 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 24.1.3, Clock Supply**.

24.3.2.2 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: Readable/writable 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 24.15 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

24.3.3 Details of TAUJn Control Registers

24.3.3.1 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: Readable/writable 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 24.16 TAUJnCDRm Register Contents

Bit Position	Bit Name	Function
31 to 0	TAUJnCDR [31:0]	Data register for capture/compare values

24.3.3.2 TAUJnCNTm — TAUJn Channel Counter Register

This is a channel m counter register.

Access: Only readable in 32-bit units.

Address: <TAUJn_base> + 10_H + m × 4_H

Value after reset: FFFF $FFFF_H$
The initial value after reset depends on the operating mode.

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 24.17 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 24.18 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 reset release

24.3.3.3 TAUJnCMORm — TAUJn Channel Mode OS Register

This register controls channel m operation.

Access: Readable/writable in 16-bit units.
Writable only 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 24.19 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	Valid edge of TAUJTTINm input signal
1	0	Setting prohibited
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 24.19 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>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 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 rowspan="3">Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</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 valid only when channel m is in capture 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 to this bit, write the value after reset.</td></tr>
</table>

Table 24.19 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.

24.3.3.4 TAUJnCMURm — TAUJn Channel Mode User Register

This register specifies a type of valid edge detection used for TAUJTTINm input.

Access: Readable/writable 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 24.20 TAUJnCMURm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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].

24.3.3.5 TAUJnCSRm — TAUJn Channel Status Register

This register indicates the overflow status of channel m.

Access: Only readable in 8-bit units.

Address: <TAUJn_base> + 30_H + m × 4_H

Value after reset: 00_H

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

Table 24.21 TAUJnCS Rm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset 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].

24.3.3.6 TAUJnCSCm — TAUJn Channel Status Clear Trigger Register

This register is a trigger register for clearing the overflow flag TAUJnCSRm.TAUJnOVF of channel m.

Access: Only writable in 8-bit units.

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 24.22 TAUJnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	TAUJnCLOV	0: No function 1: Clears the overflow flag TAUJnCSRm.TAUJnOVF

24.3.3.7 TAUJnTS — TAUJn Channel Start Trigger Register

This register enables the counter operation for each channel.

Access: Only writable in 8-bit units.

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 24.23 TAUJnTS Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing to these bits, 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.

24.3.3.8 TAUJnTE — TAUJn Channel Enable Status Register

This register indicates whether a counter operation is enabled.

Access: Only readable 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 24.24 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.

24.3.3.9 TAUJnTT — TAUJn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access: Only writable in 8-bit units.

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 24.25 TAUJnTT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing to these bits, 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.

24.3.4 Details of TAUJn Simultaneous Rewrite Register

24.3.4.1 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/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 24.26 TAUJnRDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, 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

24.3.4.2 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/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 24.27 TAUJnRDM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, 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.

24.3.4.3 TAUJnRDT — TAUJn Channel Reload Data Trigger Register

This register triggers the simultaneous rewrite enabling state.

Access: This register can only be written in 8-bit unit.

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 24.28 TAUJnRDT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing to these bits, 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

24.3.4.4 TAUJnRSF — TAUJn Channel Reload Status Register

This flag register indicates the simultaneous rewrite status.

Access: This register can only 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 24.29 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).

24.3.5 Details of TAUJn Output Registers

24.3.5.1 TAUJnTOE — TAUJn Channel Output Enable Register

This register enables and disables independent channel output mode controlled by software.

Access: This register can be read/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 24.30 TAUJnTOE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, 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 Only TAUJnTOm bits for which timer output of a channel is disabled (TAUJnTOEm = 0) can be written.

24.3.5.2 TAUJnTO — TAUJn Channel Output Register

This register specifies and reads the level of TAUJTTOUTm.

Access: This register can be read/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 24.31 TAUJnTO Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, 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.

24.3.5.3 TAUJnTOM — TAUJn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read/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 24.32 TAUJnTOM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, 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.

24.3.5.4 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/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 24.33 TAUJnTOC Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
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.

24.3.5.5 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/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 24.34 TAUJnTOL Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, 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.

24.3.5.6 TAUJnEMU — TAUJn Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read/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 24.35 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 to these bits, write the value after reset.

24.4 Operating Procedure

The following lists the general operation procedure for the TAUJn:

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 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 24.12, Independent Channel Operation Functions**.
2. The function can be changed while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

24.5 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 24.5.1, Rules of Synchronous Channel Operation Function**.

The synchronous channel operation function are detailed in the following section.

- **Section 24.13, Synchronous Channel Operation Functions**

24.5.1 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 several 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 slave channels synchronizing to the master channel. 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 24.4**.

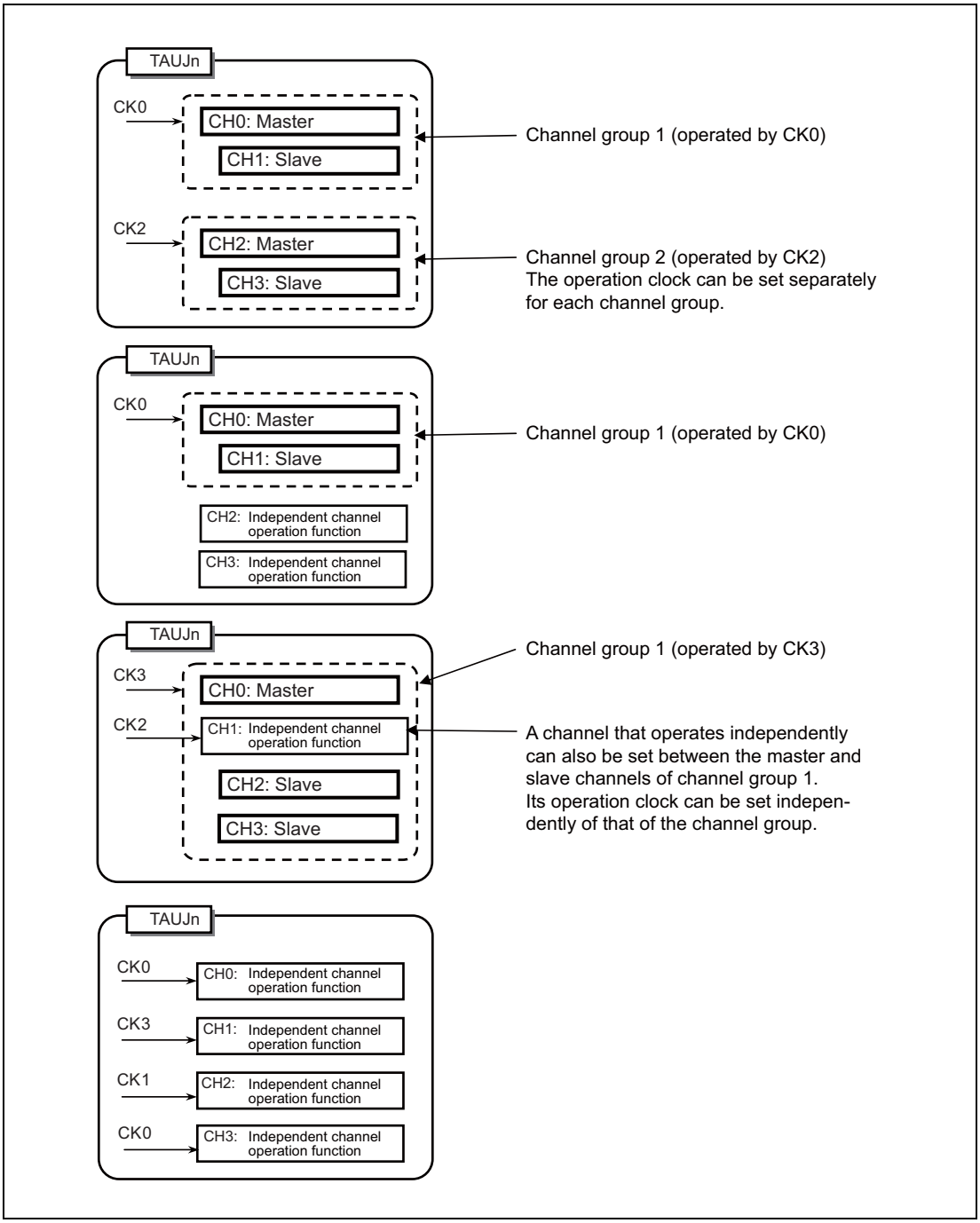


Figure 24.4 Grouping of Channels and Assignment of Operation Clocks

24.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously, both within a TAUJ unit and between TAUJ units.

24.5.2.1 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.

24.5.2.2 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.

24.6 Simultaneous Rewrite

24.6.1 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.

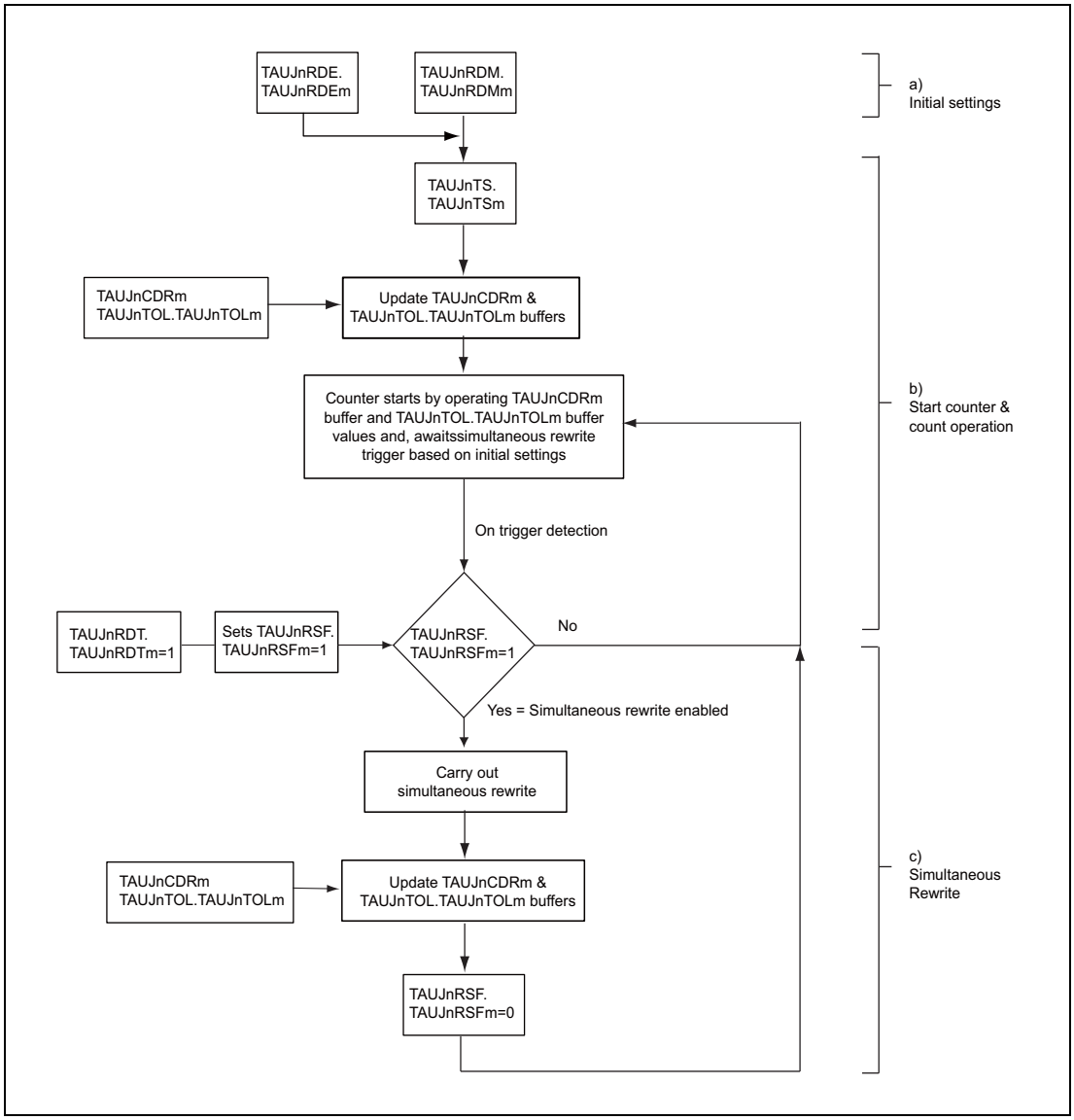


Figure 24.5 General Procedure for Simultaneous Rewrite

24.6.1.1 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.

24.6.1.2 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.

24.6.1.3 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.

24.6.2 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.

24.6.3 Simultaneous Rewrite Procedure

The simultaneous rewrite procedure with PWM output function is described in the following figure.

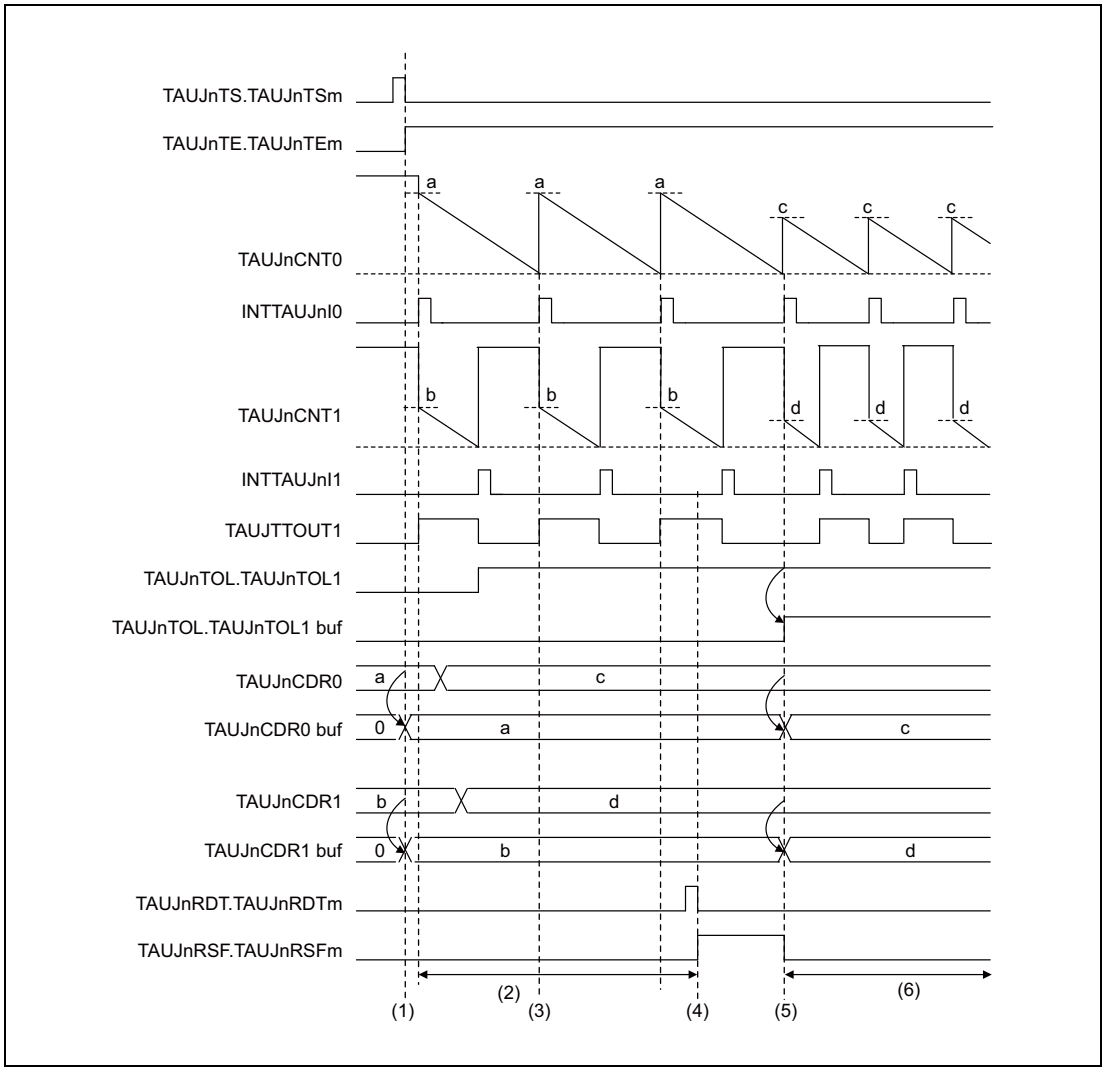


Figure 24.6 Simultaneous Rewrite with PWM Output Function

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) When the master channel starts counting, the CDRn value is loaded into CDRn buf, the TOLm value is loaded into TOLm buf, and RSFm is set to 0.

24.7 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 24.36, Channel Output Modes**.

The channel output modes are described in details below.

- **Section 24.7.2, Channel Output Modes Controlled Independently by TAUJn Signals**
- **Section 24.7.3, 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 invalid TAUJTTOUTm signal output.

See **Section 24.6, Simultaneous Rewrite**.

The various channel output modes and the channel output control bits are listed in **Table 24.36**.

Table 24.36 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):

- TAUJnTOE.TAUJnTOEm
- TAUJnTOM.TAUJnTOMm
- TAUJnTOC.TAUJnTOCm

24.7.1 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 24.36, Channel Output Modes**, and the output logic using the TAUJnTOL.TAUJnTOLm bit.

(3) Start the counter (TAUJnTS.TAUJnTSm = 1).

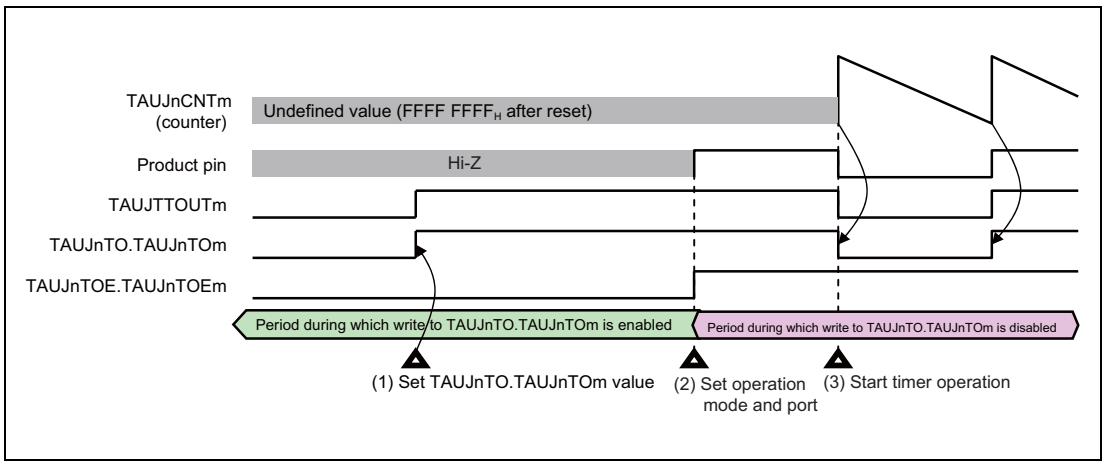


Figure 24.7 General Procedure for Specifying a TAUJTTOUTm Channel Output Mode

24.7.2 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 24.36, Channel Output Modes**.

24.7.2.1 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 24.36, Channel Output Modes**.

24.7.3 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 24.36, Channel Output Modes**.

24.7.3.1 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 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 master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in **Table 24.36, Channel Output Modes**.

24.8 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.

24.8.1 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.

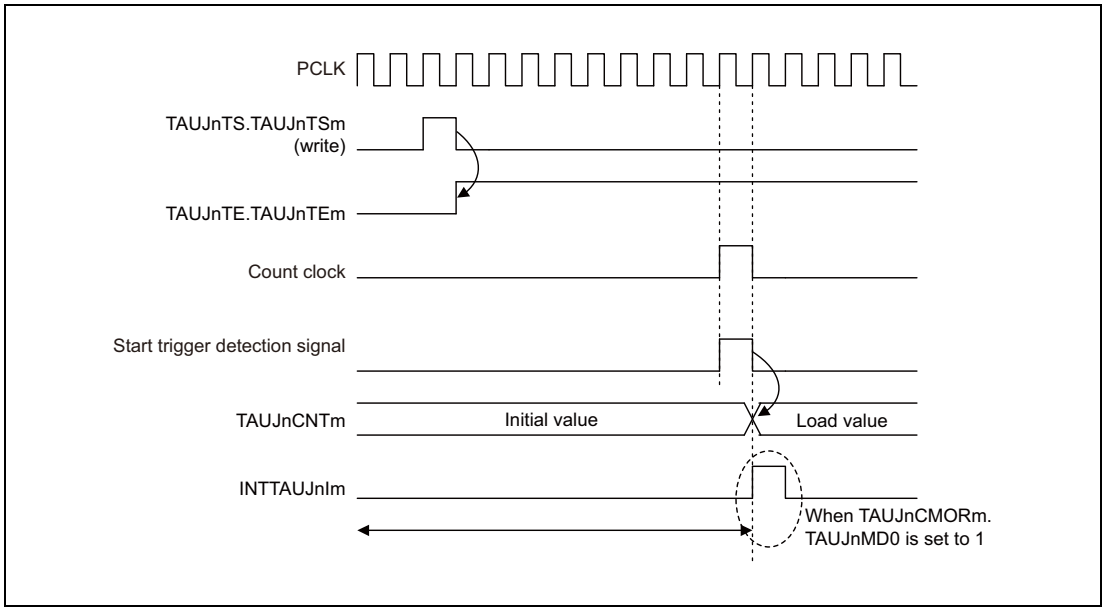


Figure 24.8 Start Timing in Interval Timer Mode, Capture Mode, and Count Capture Mode

24.8.2 Other Operating Modes

In other operating modes, 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.

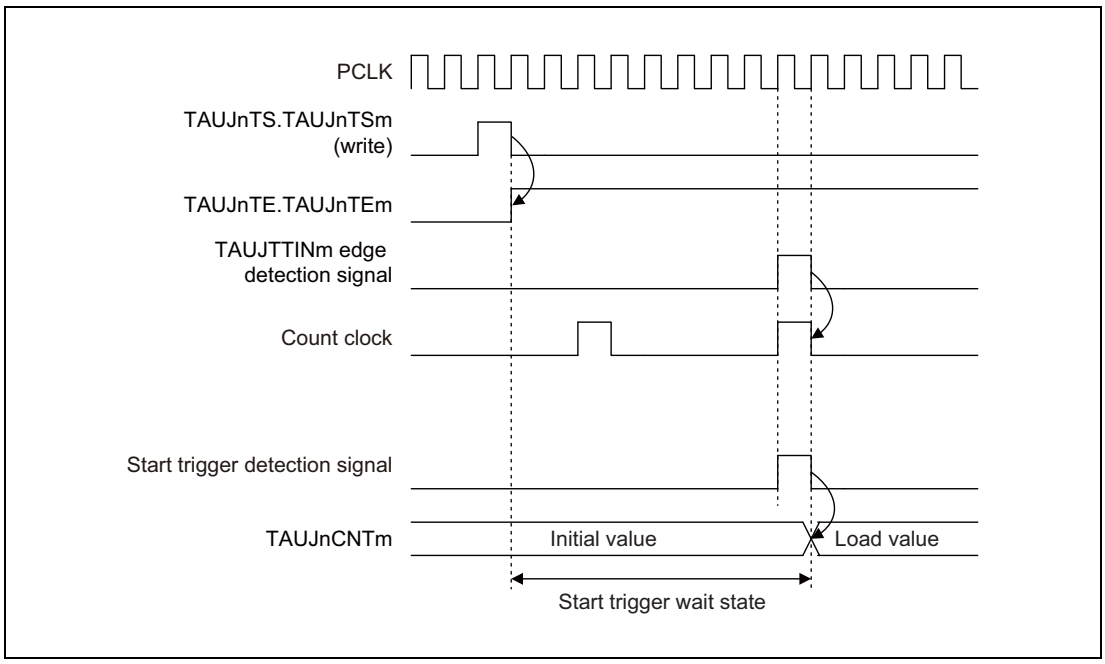


Figure 24.9 Start Timing in Other Operating Modes

24.9 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.

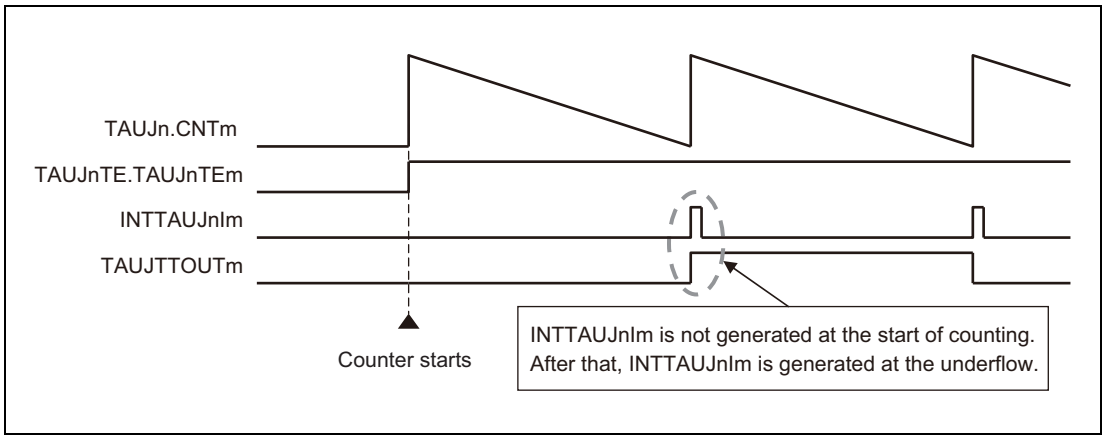


Figure 24.10 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJMD0 = 0)

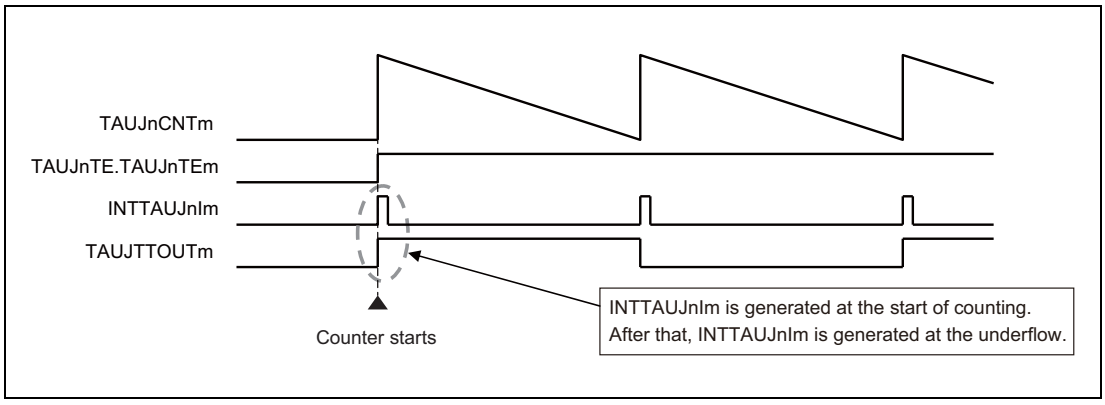


Figure 24.11 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJMD0 = 1)

24.10 Interrupt Generation upon Overflow

Certain independent functions that count up, overflow without generating an interrupt when they reach FFFF $FFFF_H$. This section describes how it is possible to generate an interrupt, by combining a channel operating in one of these modes with a channel in a different operation mode which 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.

24.10.1 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 TAUDJnCNTm of the TAUJTTINm input position detection function exceeds FFFF FFFF$_H$.

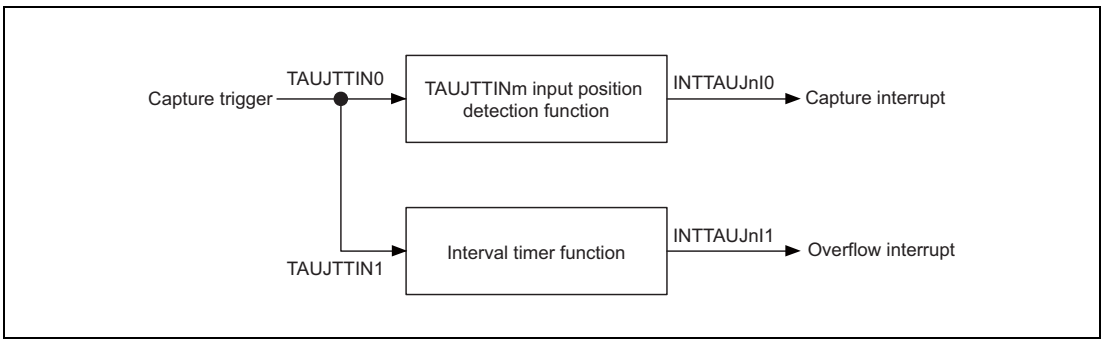


Figure 24.12 Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function

Timing diagram

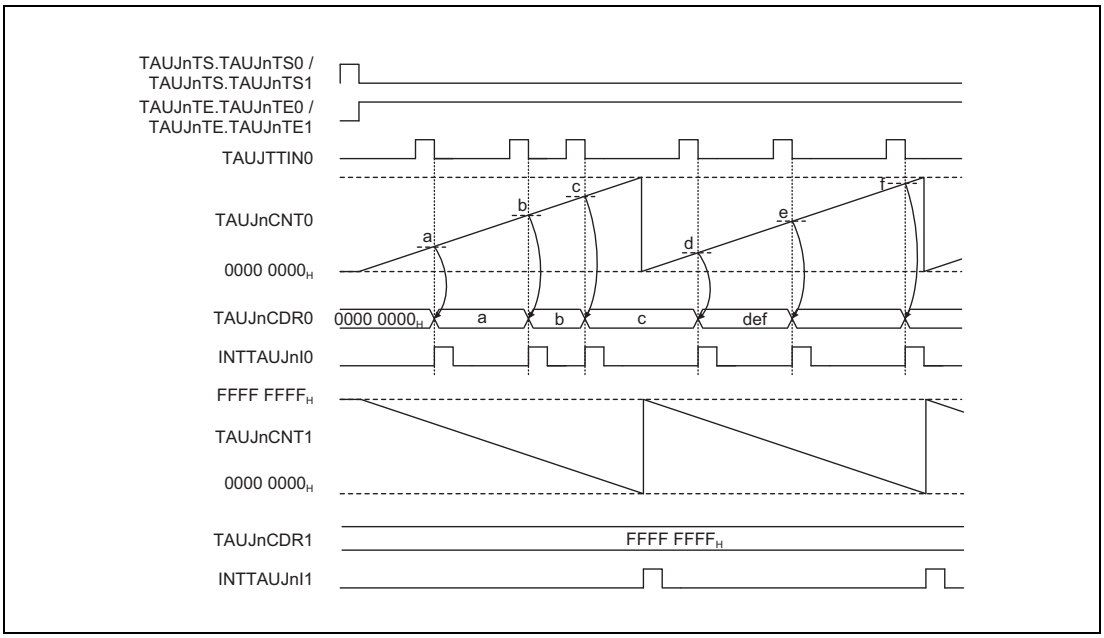


Figure 24.13 Interrupt Generation via Combination of the TAUJTTINm Input Position Detection Function and the Interval Timer Function

24.11 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.

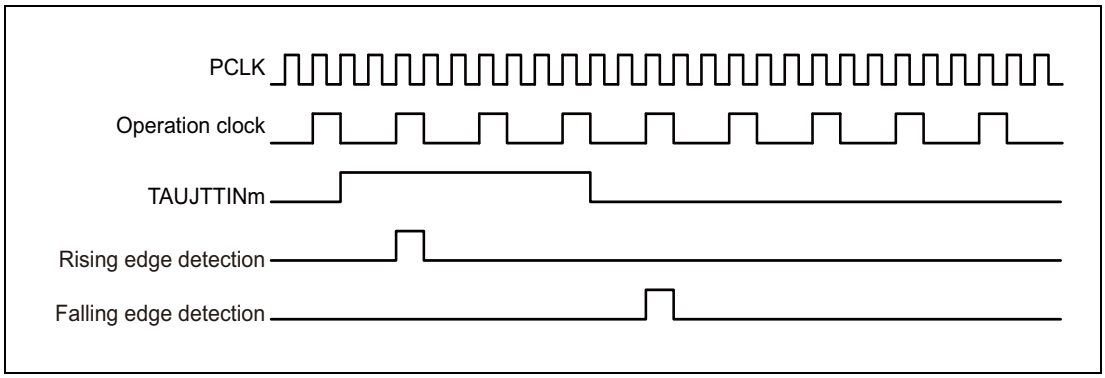


Figure 24.14 Basic Edge Detection Timing

Figure 24.14 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.

24.12 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 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 (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 TAUJnTTOUTm 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 negative TAUJTTOUTm signal compared to when TAUJnCMORm.TAUJnMD0 is set to 1.

24.12.1.2 Equations

INTTAUJnIm cycle = count clock cycle × (TAUJnCDRm + 1)

TAUJTTOUTm square wave cycle = count clock cycle × (TAUJnCDRm+1) × 2

24.12.1.3 Block Diagram and General Timing Diagram

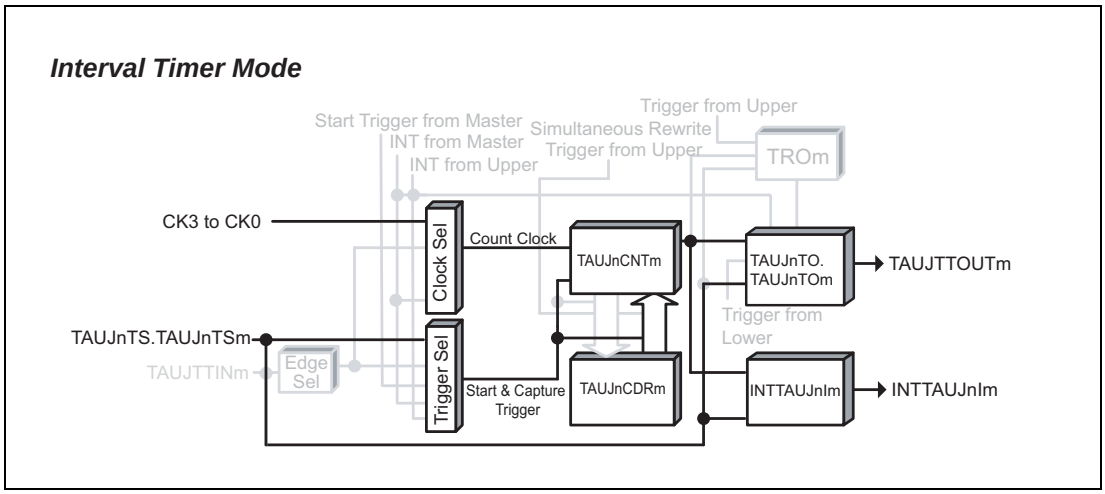


Figure 24.15 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at operation start (TAUJnCMORm.(TAUJnMD0 = 1)

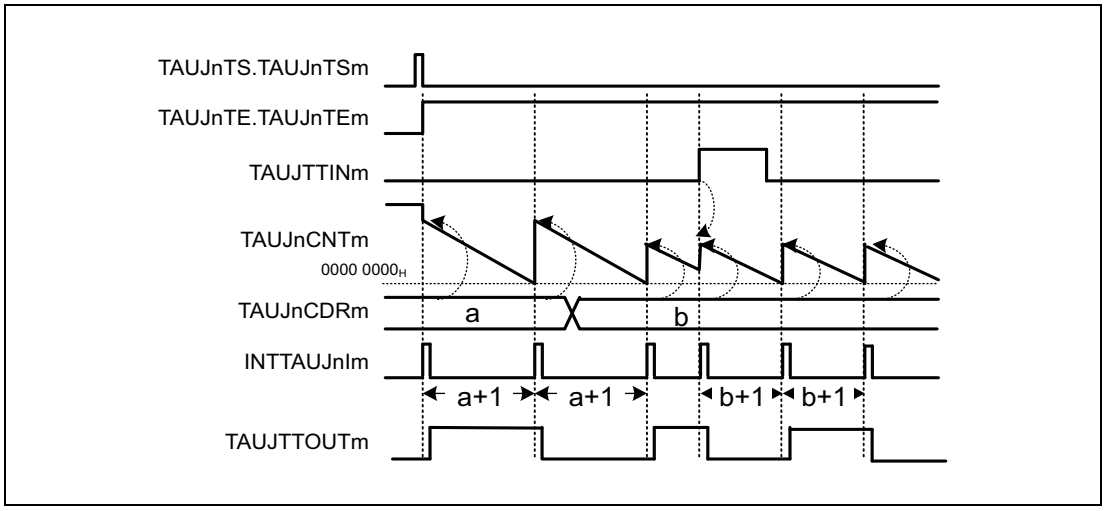


Figure 24.16 General Timing Diagram for Interval Timer Function

24.12.1.4 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 24.37 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 to this bit, 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 24.38 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 to these bits, write the value after reset.
0, 1	TAUJnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode

Table 24.39 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 24.11, TAUJTTINm Edge Detection**.

(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 24.40 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.

24.12.1.5 Operating Procedure for Interval Timer Function

Table 24.41 Operating Procedure for Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 24.37, Contents of the TAUJnCMORm register for Interval Timer Function** and **Table 24.38, 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 24.39, 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

24.12.1.6 Specific Timing Diagrams

(1) TAUJnCDRm = 0000 0000_H, count clock = PCLK/2

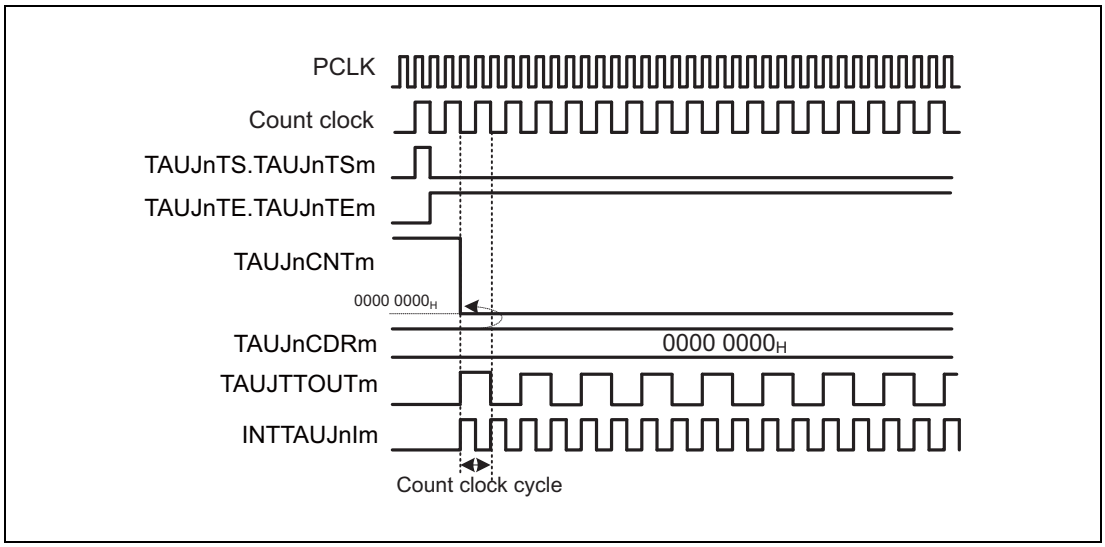


Figure 24.17 AUJnCDRm = 0000 0000_H, Count Clock = PCLK/2

- 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

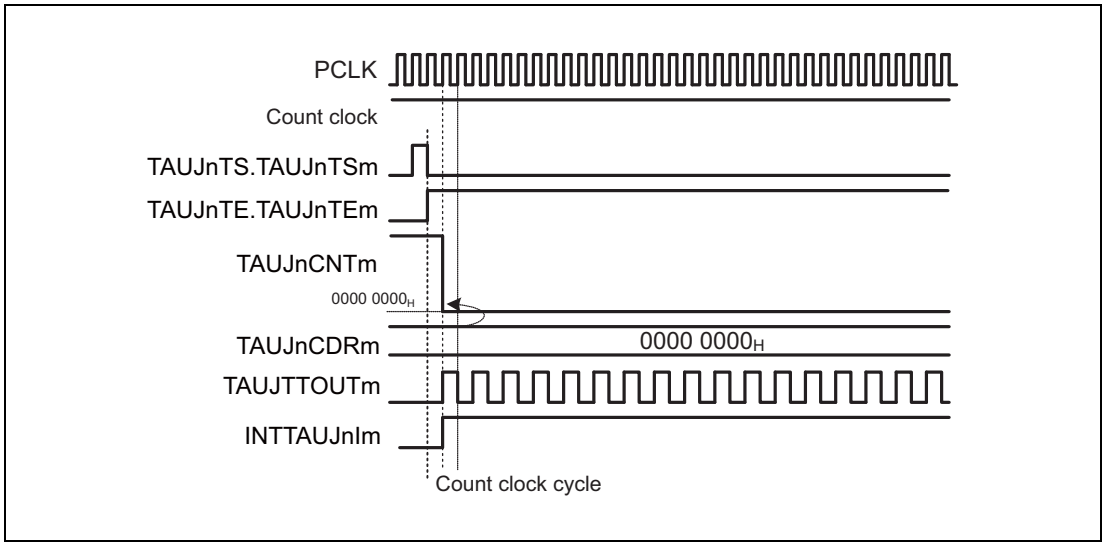


Figure 24.18 TAUJnCDRm = 0000 0000$_H$, Count Clock = PCLK

- 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$_H$.
- INTTAUDnIm is fixed at high level.
  Though the first interrupt is generated, subsequent interrupts are not generated. TAUJTTOUTm is toggled every PCLK clock.

(3) Operation stop and restart

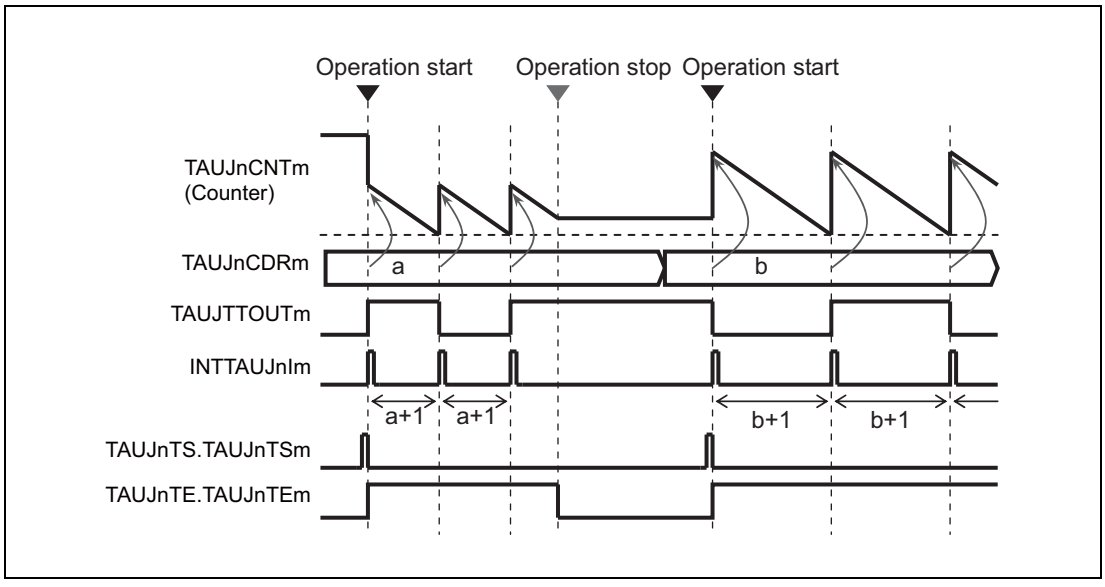


Figure 24.19 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 1)

- 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

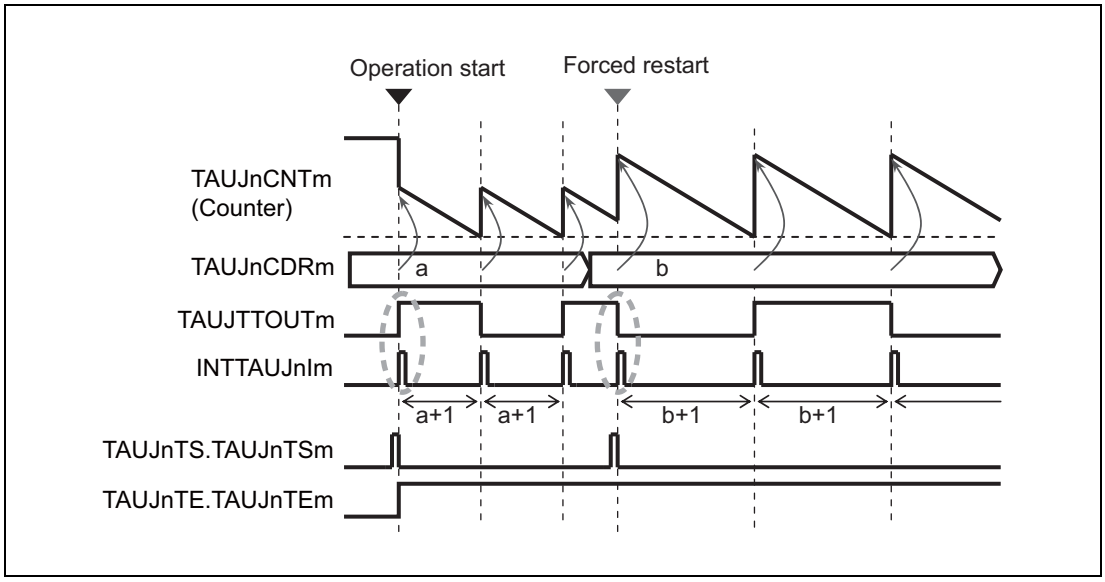


Figure 24.20 Forced Restart Operation (TAUJnCMORm.TAUJnMD0 = 1)

- 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.

24.12.2 TAUJTTINm Input Interval Timer Function

24.12.2.1 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.

Description

This function operates in an identical manner to the interval timer function (see **Section 24.12.1, 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.

24.12.2.2 Equations

INTTAUJnIm cycle = count clock cycle × (TAUJnCDRm + 1)

TAUJTTOUTm square wave cycle = count clock cycle × (TAUJnCDRm + 1) × 2

24.12.2.3 Block Diagram and General Timing Diagram

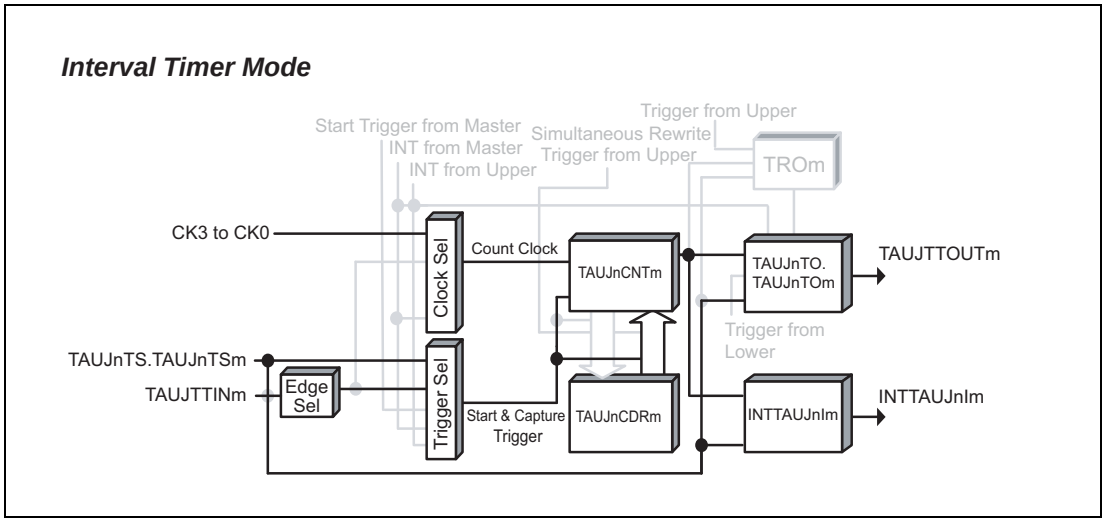


Figure 24.21 Block Diagram for TAUJTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 0)
- Rising edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 01_B)

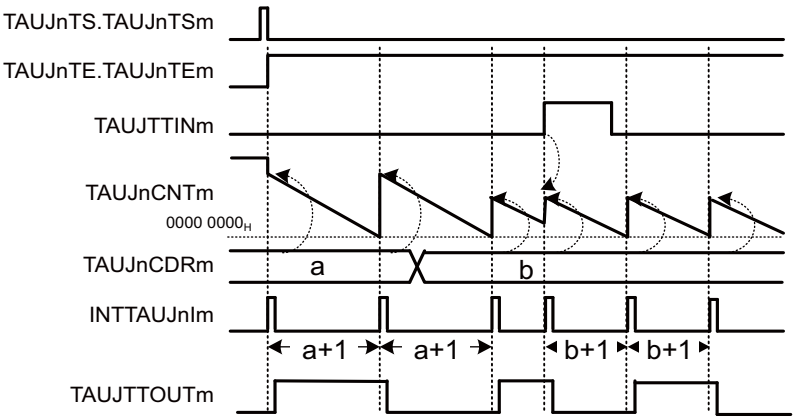


Figure 24.22 General Timing Diagram for TAUJTTINm Input Interval Timer Function

24.12.2.4 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 24.42 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 to this bit, 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 24.43 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 to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

(3) Channel output mode

Table 24.44 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 24.11, TAUJTTINm Edge Detection**.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm input interval timer function. Therefore, these registers must be set to 0.

Table 24.45 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.

24.12.2.5 Operating Procedure for TAUJTTINm Input Interval Timer Function

Table 24.46 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 24.42, Contents of the TAUJnCMORm register for TAUJTTINm Input Interval Timer Function** and **Table 24.43, 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 24.44, 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

24.12.2.6 Specific Timing Diagrams

The timing diagrams in **Section 24.12.1, Interval Timer Function** also apply, except for this function the counter can also be restarted by a valid TAUJTTINm input edge.

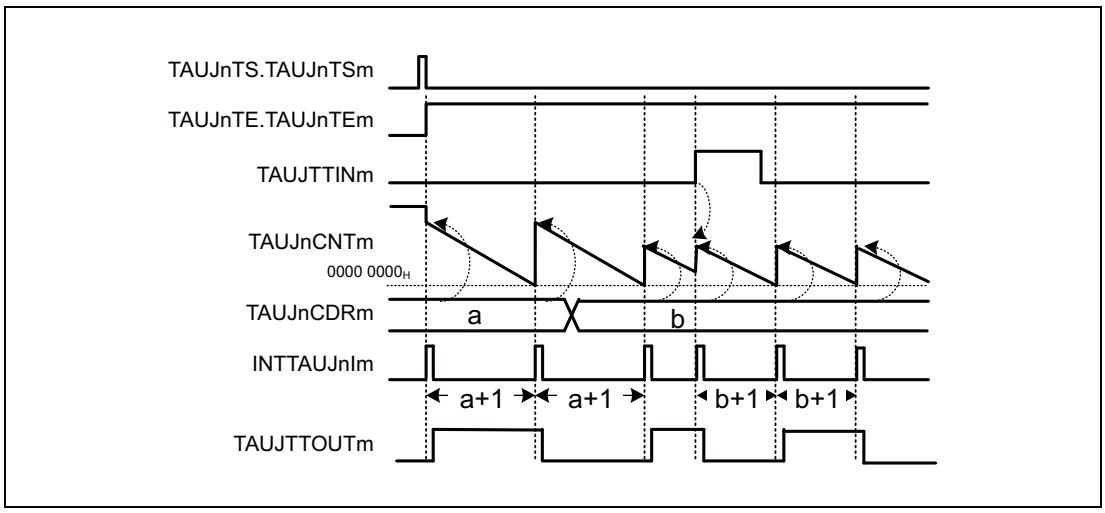


Figure 24.23 Counter Triggered By Rising TAUJTTINm input edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B), TAUJnCMORm.TAUJnMD0 = 1

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).

24.12.3 TAUJTTINm Input Pulse Interval Measurement Function

24.12.3.1 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 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 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 24.47 Effects of an Overflow

	When Overflow Occurs		When a Valid TAUJTTINm Input is then Detected	
TAUJnCMORm.COS[1:0]	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 **Table 24.37, Contents of the TAUJnCMORm register for Interval Timer Function**.

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.

24.12.3.2 Equations

TAUJTTINm input pulse interval = count clock cycle ×
[(TAUJnCSRm.TAUJnOVF × (FFFF $FFFF_H$ + 1)) + TAUJnCDRm capture value + 1]

24.12.3.3 Block Diagram and General Timing Diagram

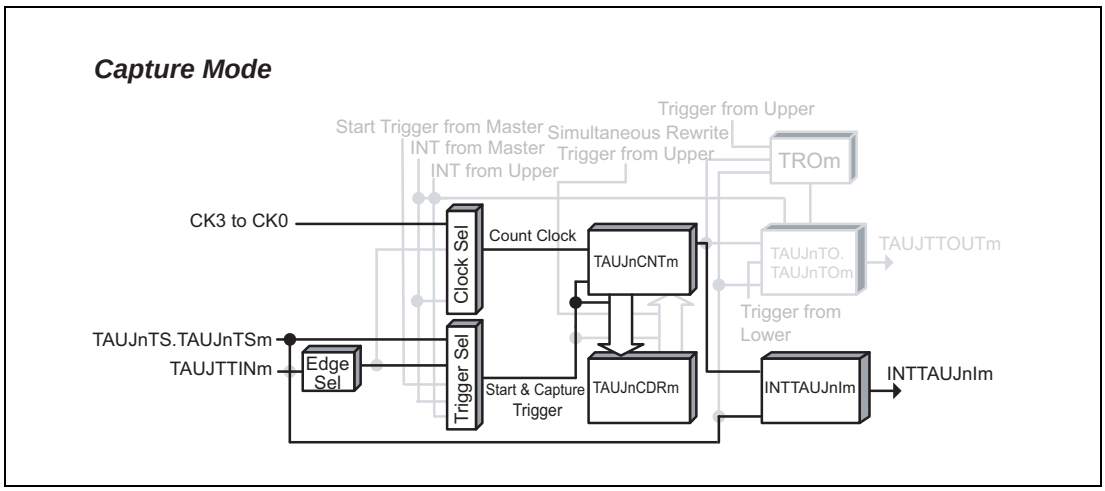


Figure 24.24 Block Diagram for TAUJTTINm Input Pulse Interval Measurement Function

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).

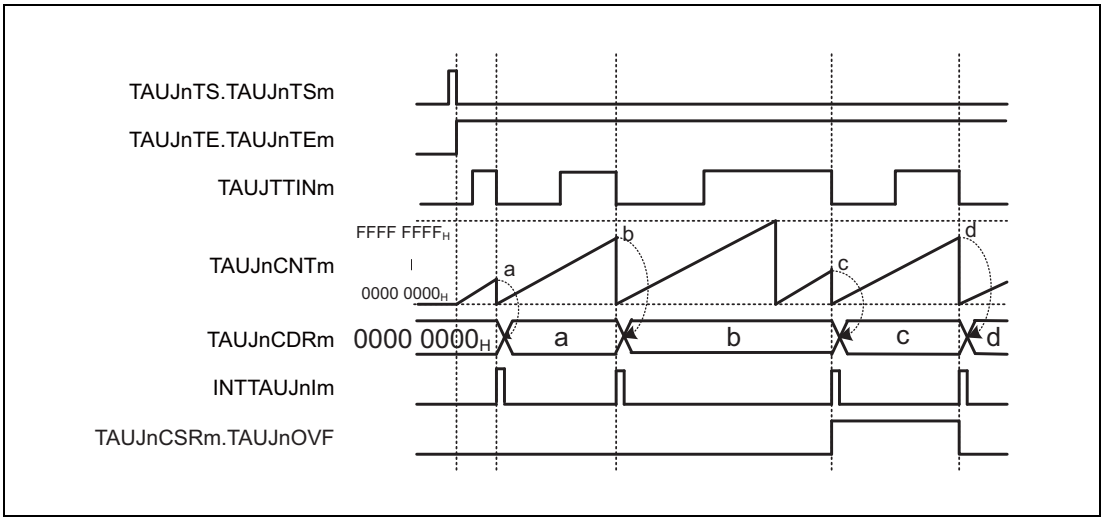


Figure 24.25 General Timing Diagram For TAUJTTINm Input Pulse Interval Measurement Function

24.12.3.4 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 24.48 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 24.47, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing to this bit, 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 24.49 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 to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

(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 24.50 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.

24.12.3.5 Operating Procedure for TAUJTTINm Input Pulse Interval Measurement Function

Table 24.51 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 24.48, Contents of the TAUJnCMORm Register for TAUJTTINm Input Pulse Interval Measurement Function** and **Table 24.49, 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

24.12.3.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

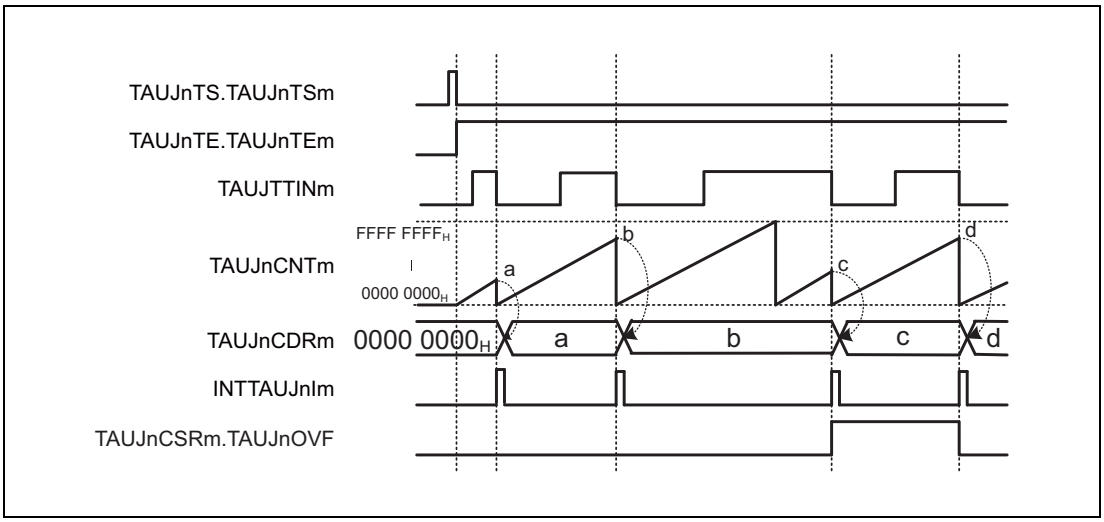


Figure 24.26 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- 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

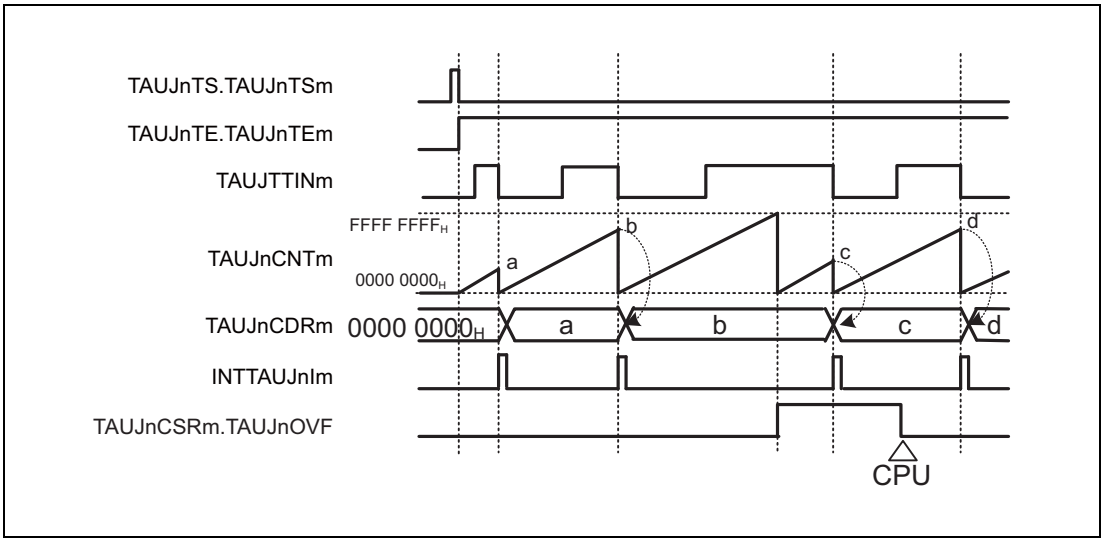


Figure 24.27 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- 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

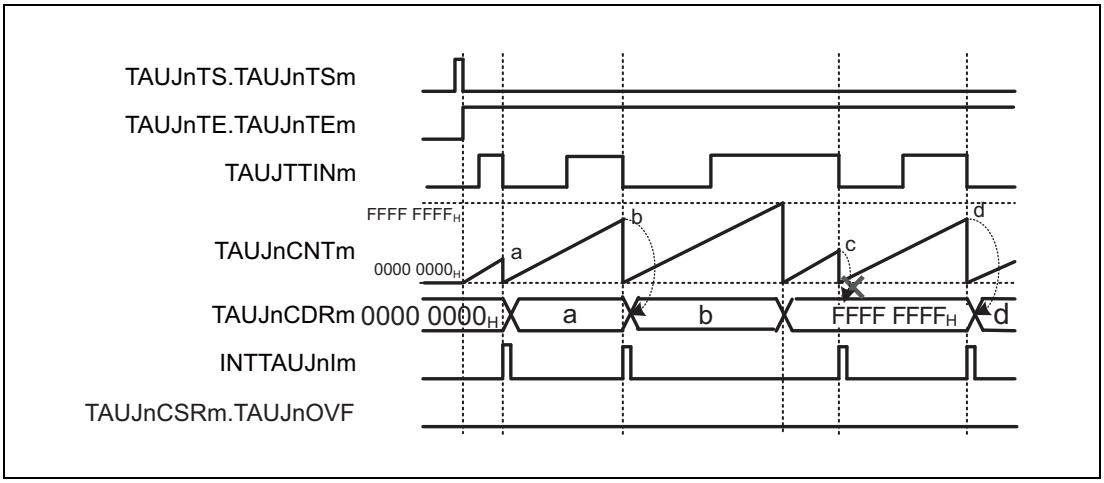


Figure 24.28 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- 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

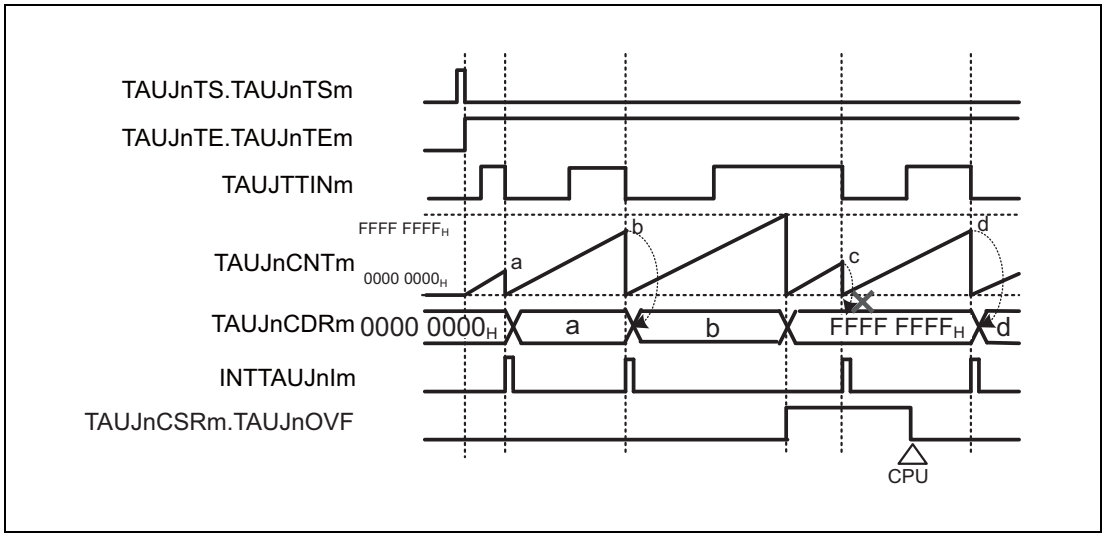


Figure 24.29 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- 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.

24.12.4 TAUJTTINm Input Signal Width Measurement Function

24.12.4.1 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 24.52 Effects of an Overflow

	When Overflow Occurs		When a Valid TAUJTTINm Input Stop Edge is Detected	
TAUJnCMORm.COS[1:0]	**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.OVF = 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.

24.12.4.2 Equations

TAUJTTINm input signal width = count clock cycle ×
[(TAUJnCSRm.TAUJnOVF × (FFFF FFFF$_H$ + 1)) + TAUJnCDRm capture value + 1]

24.12.4.3 Block Diagram and General Timing Diagram

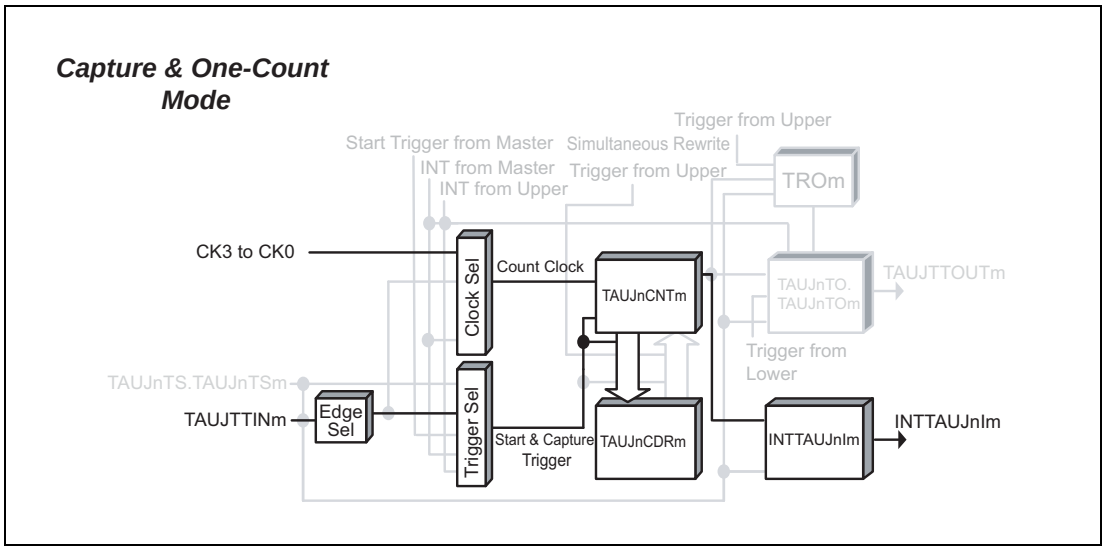


Figure 24.30 Block Diagram for TAUJTTINm Input Signal Width Measurement Function

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).

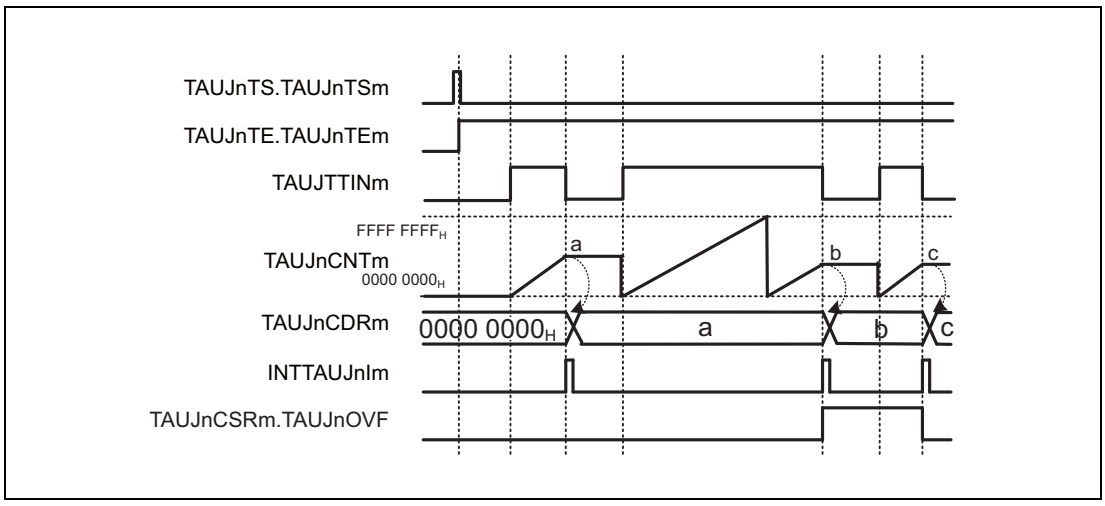


Figure 24.31 General Timing Diagram for TAUJTTINm Input Signal Width Measurement Function

24.12.4.4 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 24.53 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 24.52, Effects of an Overflow**.
5	Reserved	When read, the value after reset is returned. When writing to this bit, 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 24.54 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 to these bits, 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 24.55 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.

24.12.4.5 Operating Procedure for TAUJTTINm Input Signal Width Measurement Function

Table 24.56 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 24.53, Contents of the TAUJnCMORm Register for TAUJTTINm Input Signal Width Measurement Function** and **Table 24.54, 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 TAUJnCSC.CLOV 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. 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

24.12.4.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

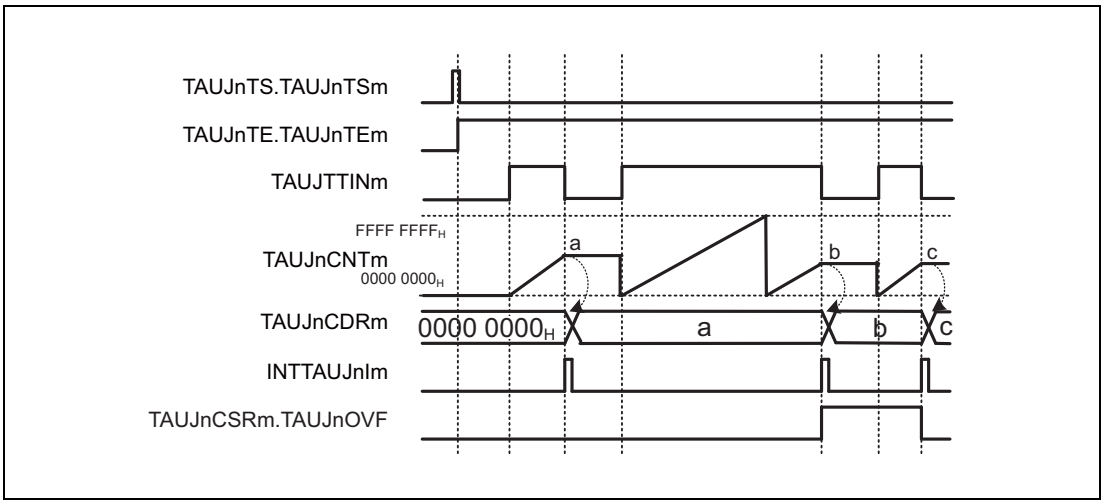


Figure 24.32 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- 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

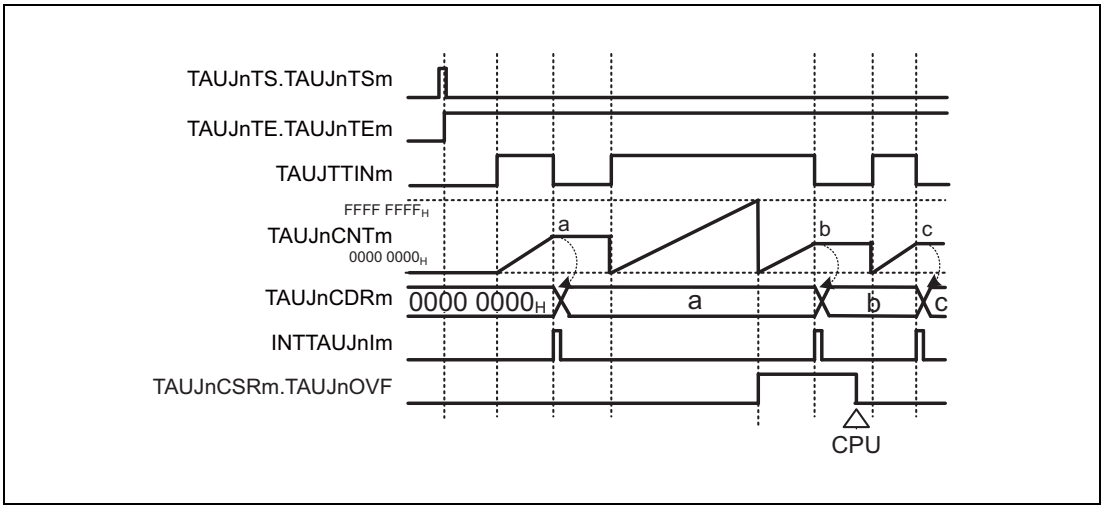


Figure 24.33 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- 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

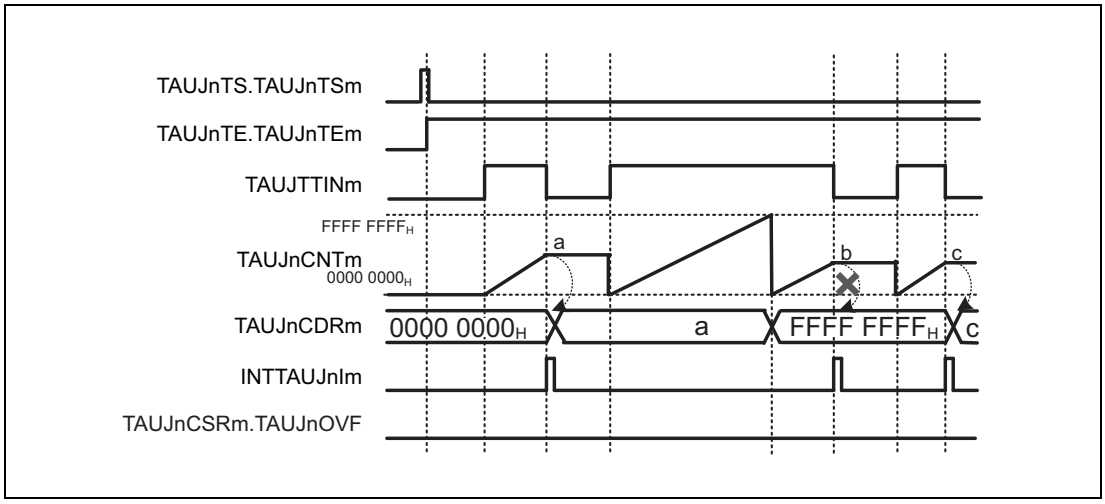


Figure 24.34 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- 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

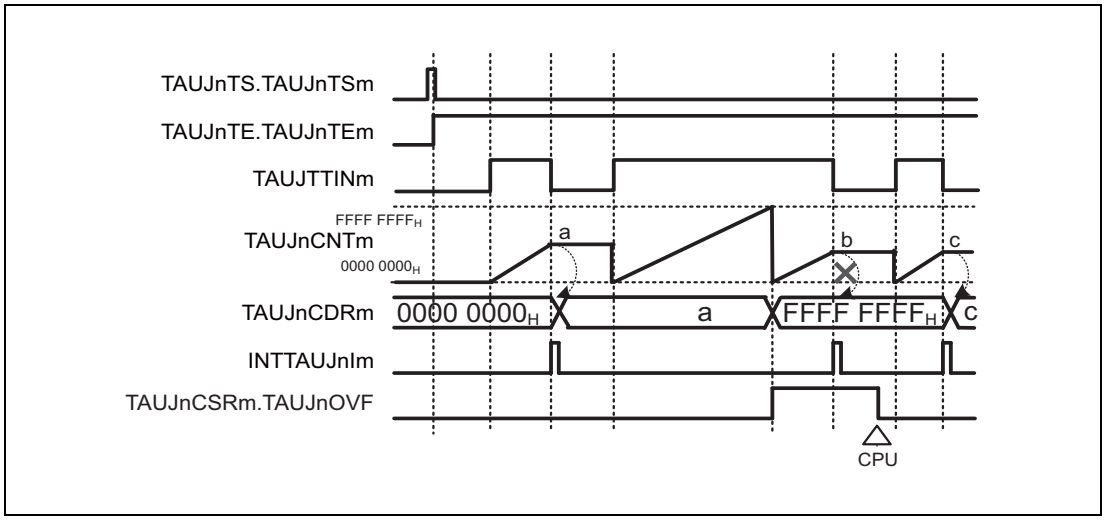


Figure 24.35 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- 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.

24.12.5 TAUJTTINm Input Position Detection Function

24.12.5.1 Overview

Summary

This function measures the interval of an input signal 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 stop 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.

Conditions

If the TAUJnCMORm. TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated.

24.12.5.2 Equations

Function duration at a TAUJTTINm input pulse =
count clock cycle × (TAUJnCDRm capture value + 1)

24.12.5.3 Block Diagram and General Timing Diagram

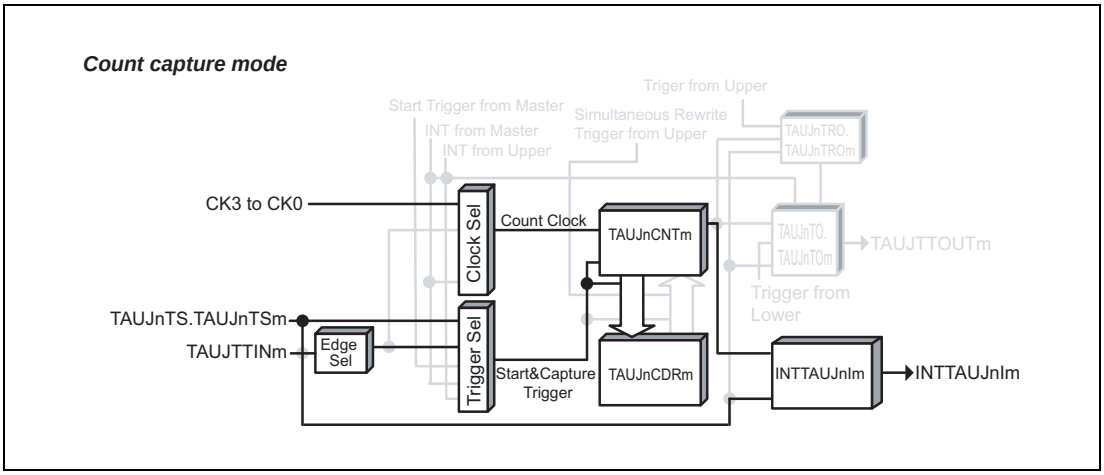


Figure 24.36 Block Diagram of TAUJTTINm Input Period Count Detection Function

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)

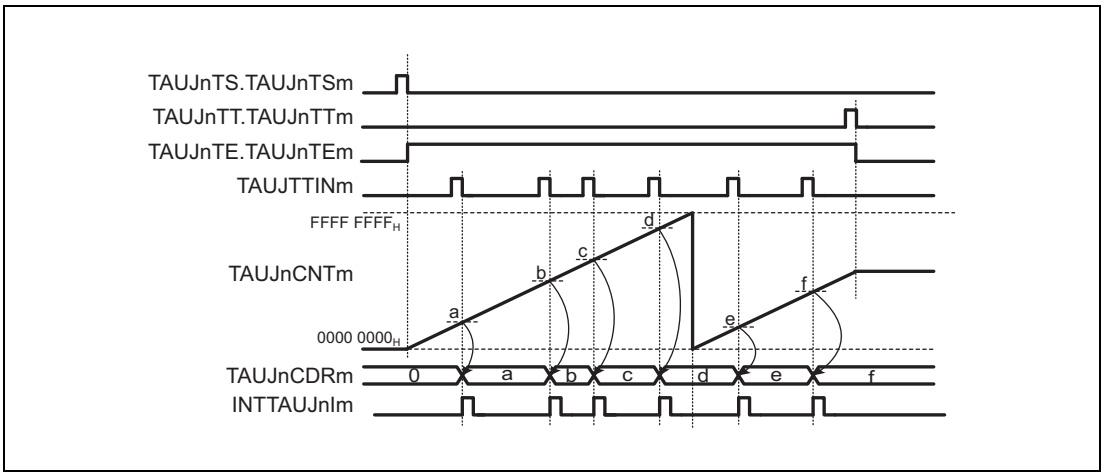


Figure 24.37 General Timing Diagram for TAUJTTINm Input Position Detection Function

24.12.5.4 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 24.57 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 to this bit, 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 24.58 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 to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

(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 24.59 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

24.12.5.5 Operating Procedure for TAUJTTINm Input Position Detection Function

Table 24.60 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 24.57, Contents of the TAUJnCMORm Register for TAUJTTINm Input Position Detection Function** and **Table 24.58, 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

24.12.5.6 Specific Timing Diagrams

(1) Operation stop and restart

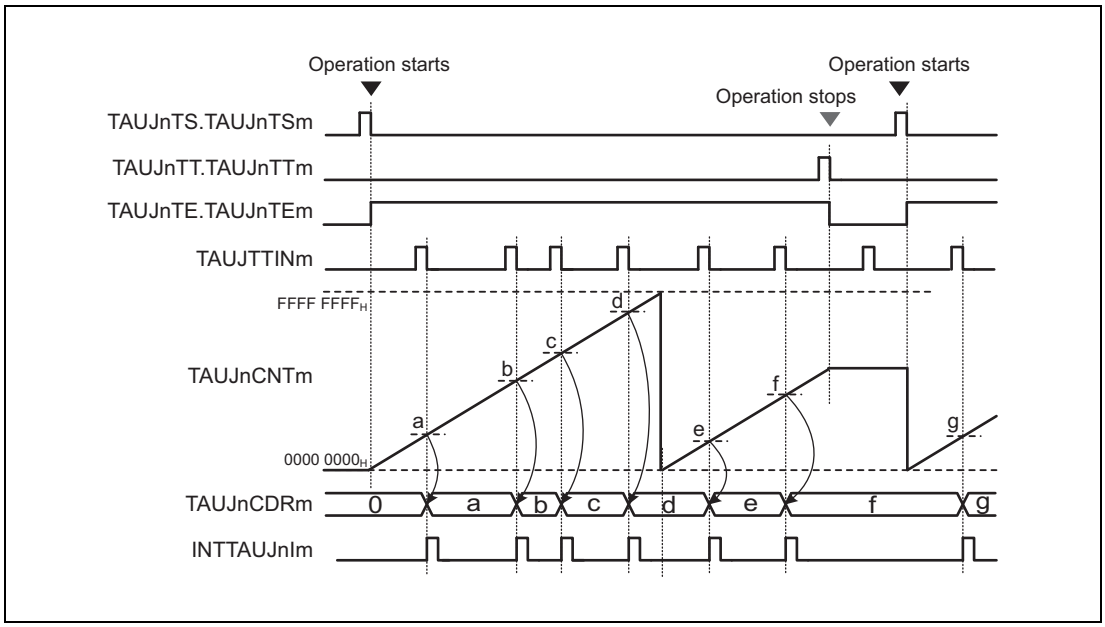


Figure 24.38 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

- 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.

24.12.6 TAUJTTINm Input Period Count Detection Function

24.12.6.1 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.

24.12.6.2 Equations

Cumulative TAUJTTINm input width =
count clock cycle × (TAUJnCDRm capture value + 1)

24.12.6.3 Block Diagram and General Timing Diagram

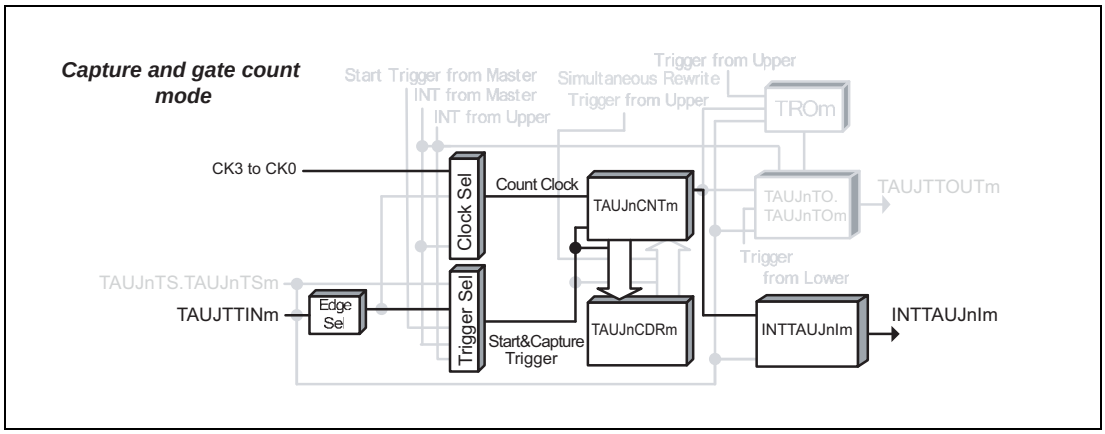


Figure 24.39 Block Diagram for TAUJTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

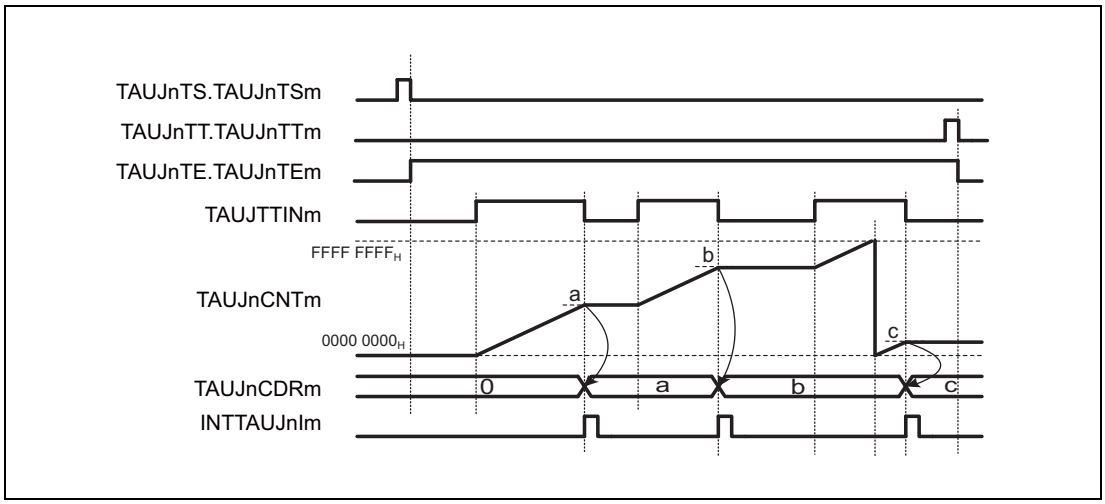


Figure 24.40 General Timing Diagram for TAUJTTINm Input Period Count Detection Function

24.12.6.4 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 24.61 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 to this bit, 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 24.62 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 to these bits, 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 24.63 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

24.12.6.5 Operating Procedure for TAUJTTINm Input Period Count Detection Function

Table 24.64 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 24.61, Contents of the TAUJnCMORm Register for TAUJTTINm Input Period Count Detection Function** and **Table 24.62, 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	TAUJnTTINm 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 start 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

24.12.6.6 Specific Timing Diagrams

(1) Operation Stop and Restart

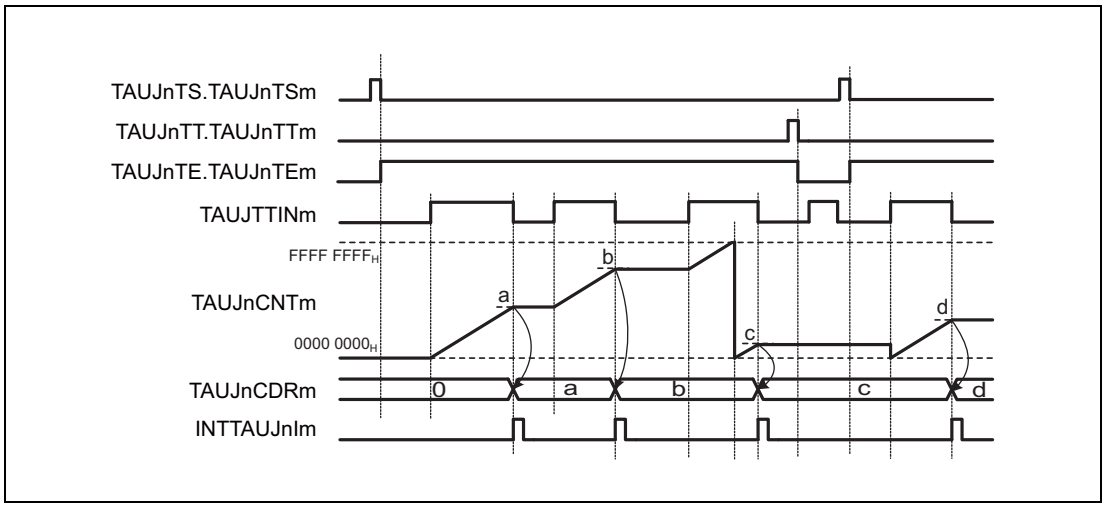


Figure 24.41 Operation Stop and Restart (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

- 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 $0000\ 0000_H$.

24.12.7 Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

24.12.7.1 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 FFF $FFFFF_H$.

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.

24.12.7.2 Block Diagram and General Timing Diagram

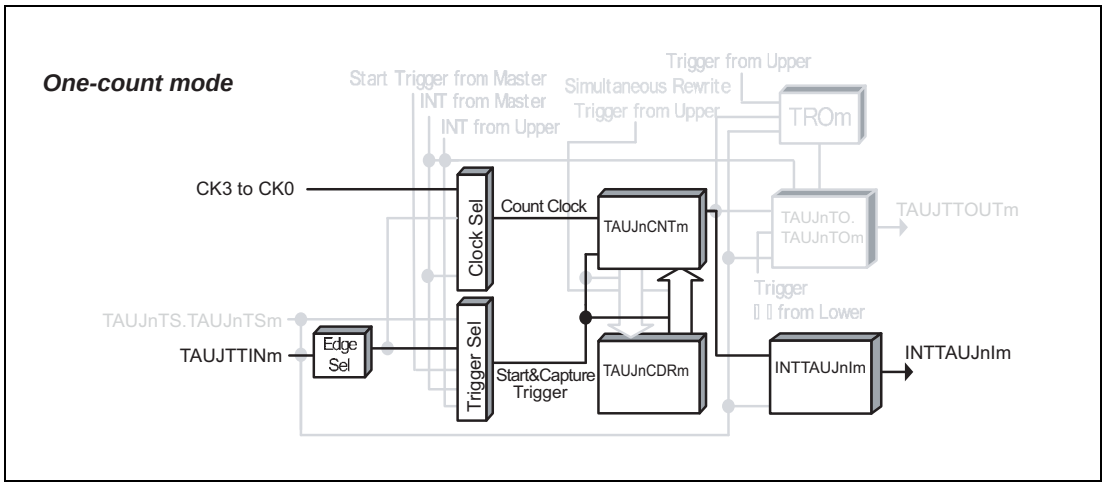


Figure 24.42 Block Diagram for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

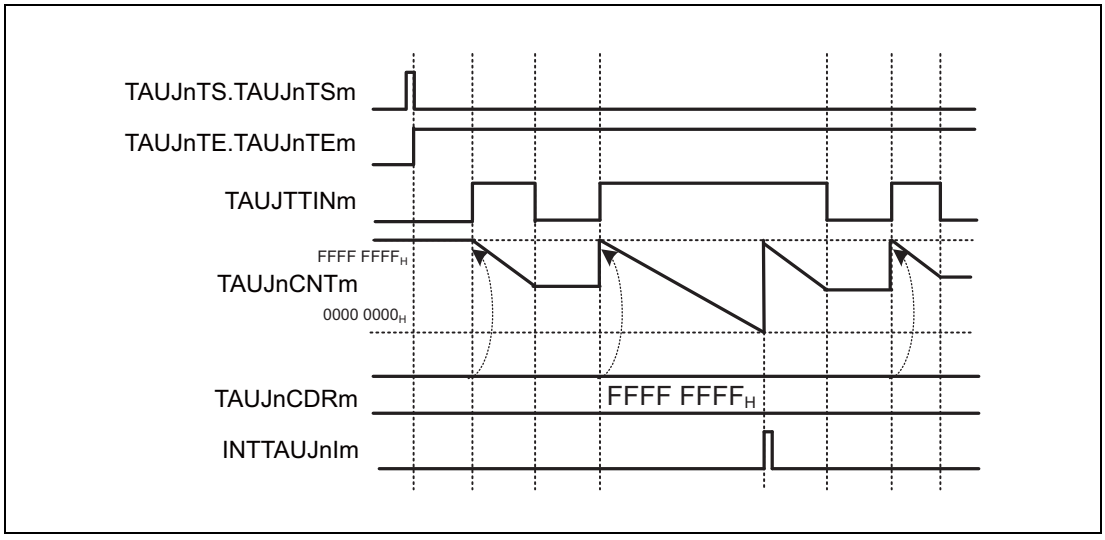


Figure 24.43 General Timing Diagram for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

24.12.7.3 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 24.65 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 to this bit, 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 24.66 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 to these bits, 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 24.67 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

24.12.7.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Table 24.68 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 24.65, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)** and **Table 24.66, 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

24.12.8 Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

24.12.8.1 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.

24.12.8.2 Block Diagram and General Timing Diagram

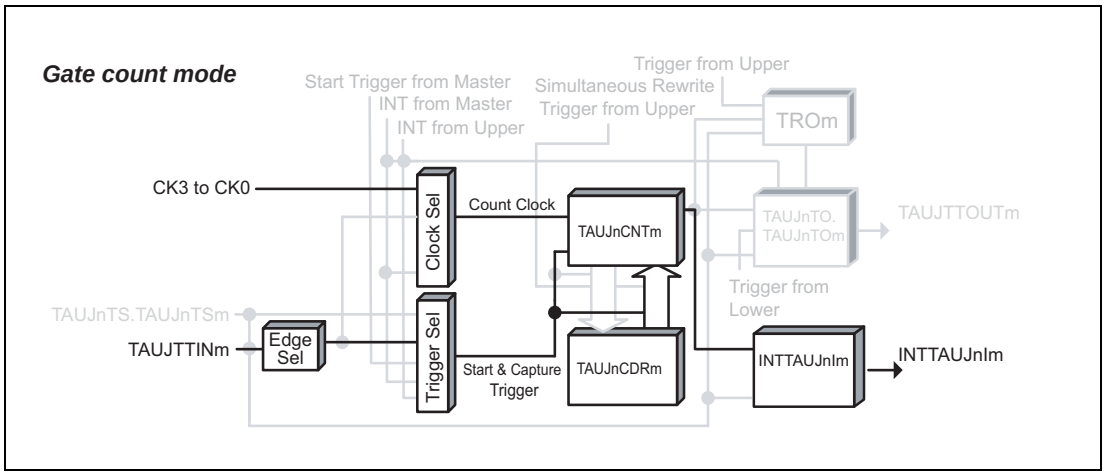


Figure 24.44 Block Diagram for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

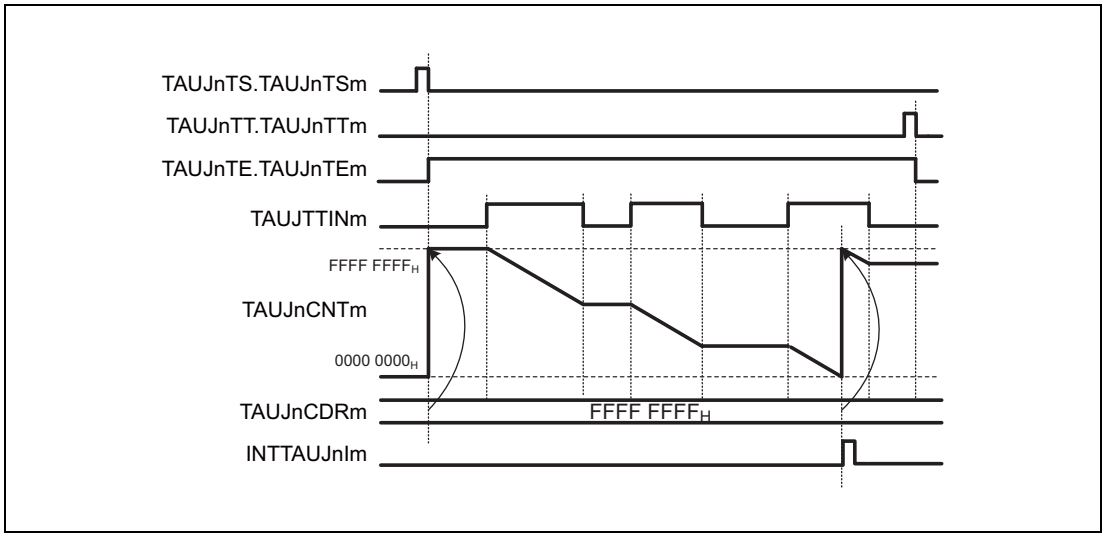


Figure 24.45 General Timing Diagram For Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

24.12.8.3 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 24.69 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 to this bit, 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 24.70 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 to these bits, 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 24.71 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.

24.12.8.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Table 24.72 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 24.69, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)** and **Table 24.70, 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, TAUJnCNTm the TAUJnCDRm value (FFFF FFFF$_H$).
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

24.13 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 24.2, Overview**.

24.13.1 PWM Output Function

24.13.1.1 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 24.73, 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 24.76, 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 **24.7, 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):

  The INTTAUJnIm of 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 24.6, Simultaneous Rewrite**.

24.13.1.2 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

24.13.1.3 Block Diagram and General Timing Diagram

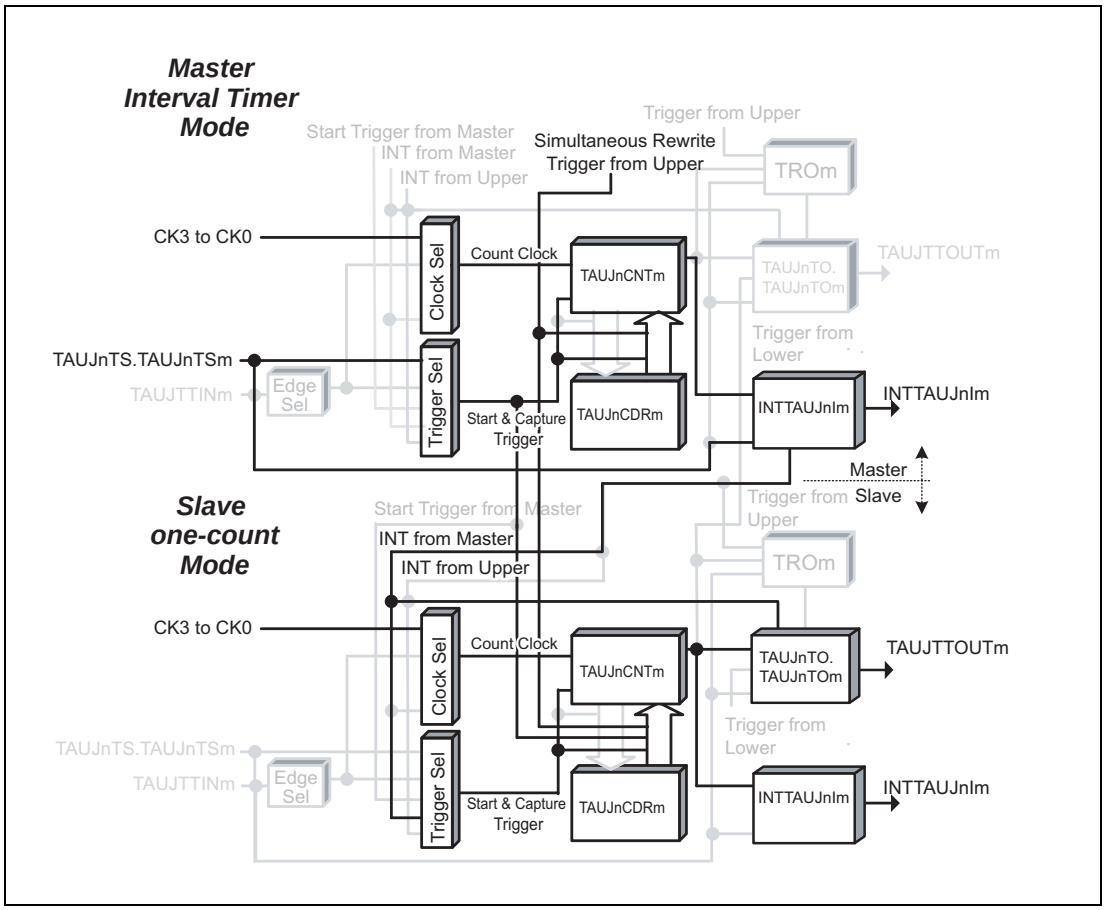


Figure 24.46 Block Diagram for PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUJnTOL.TAUJnTOLm = 0)

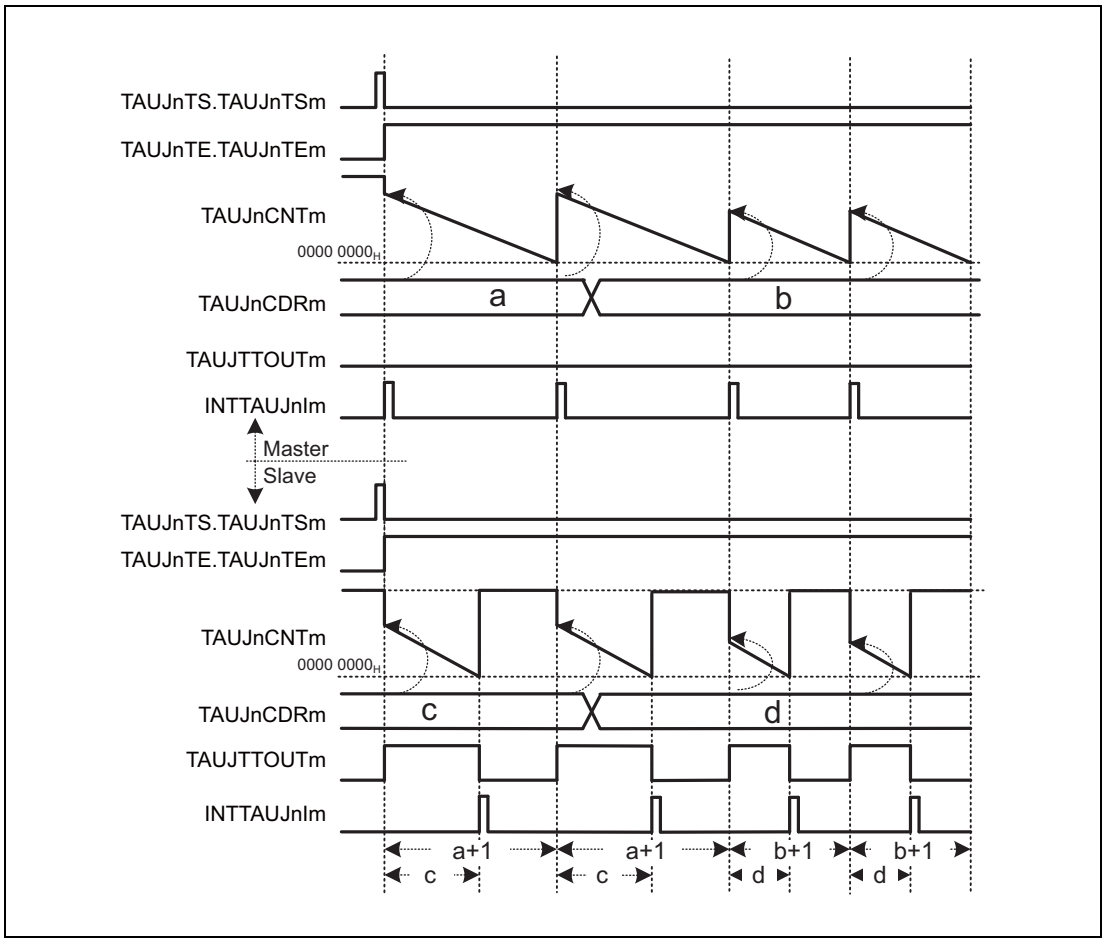


Figure 24.47 General Timing Diagram for PWM Output Function

NOTE

- The interval between the slave channel starting to count and an interrupt being generated is the value of corresponding TAUJnCDRm, whereas for the master channel the interval is the corresponding TAUJnCDRm + 1.
- The slave channel TAUJTTOUTm will rise with a delay of one count clock after the rising of the master channel INTTAUJnIm.

24.13.1.4 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 24.73 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 to this bit, 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 24.74 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 to these bits, 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 24.75 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.

24.13.1.5 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 24.76 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 to this bit, 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 24.77 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 to these bits, 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 24.78 Control Bit Settings for Independent 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 24.79 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.

24.13.1.6 Operating Procedure for PWM Output Function

Table 24.80 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 24.13.1.4, Register Settings for the Master Channel**. Slave channel: Set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in **Section 24.13.1.5, 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

24.13.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

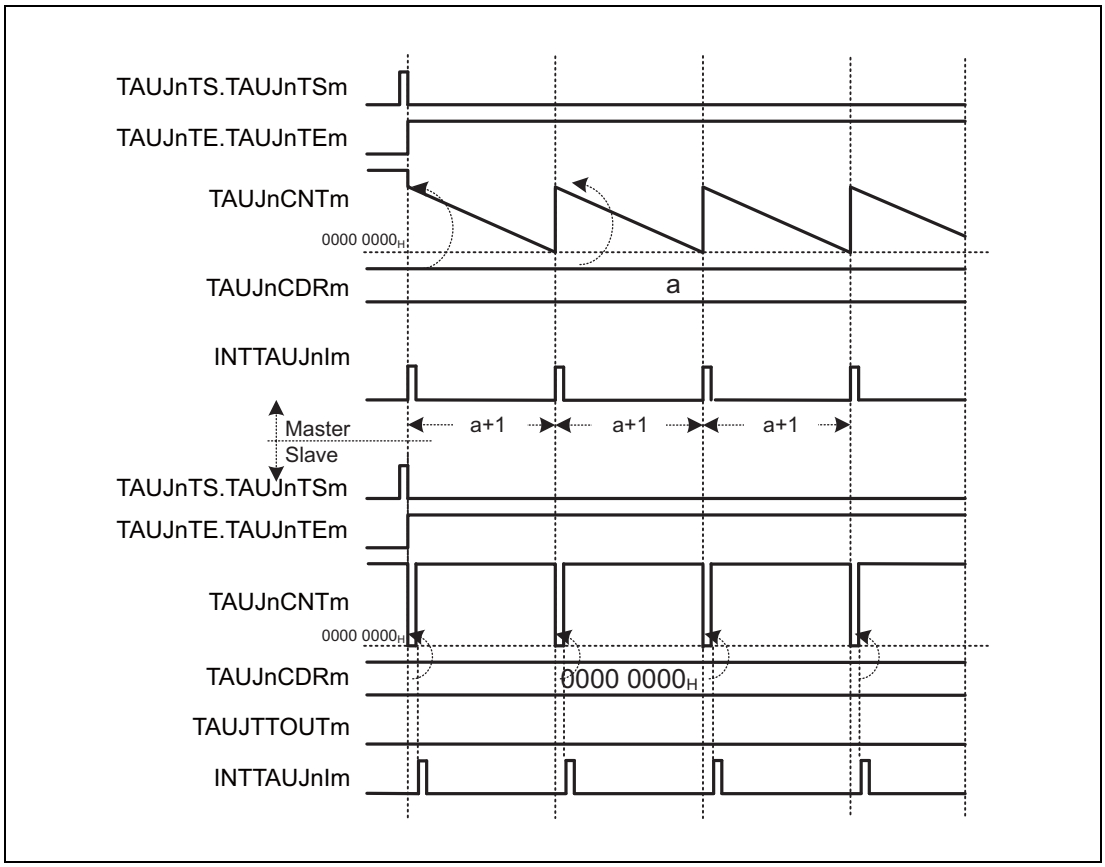


Figure 24.48 TAUJnCDRm (slave) = 0000 0000$_H$, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

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%

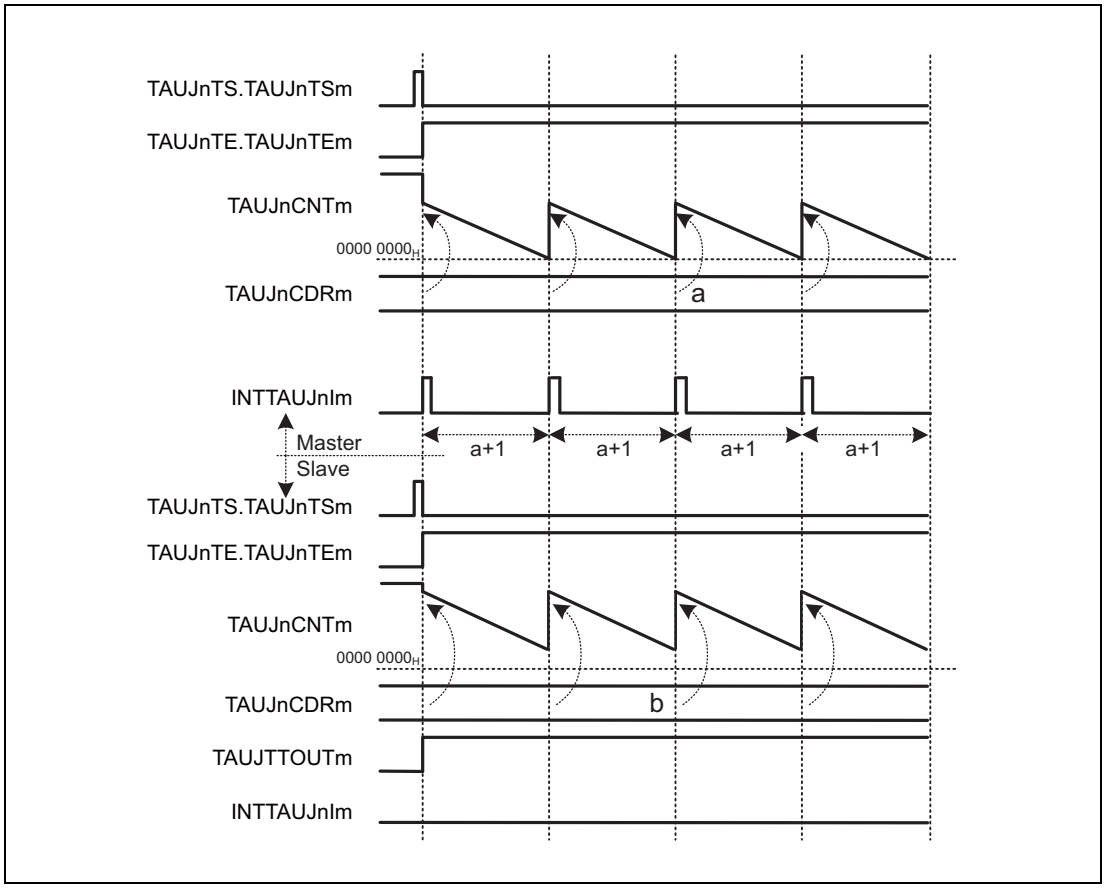


Figure 24.49 TAUJnCDRm (slave) ≥ TAUJnCDRm (master) + 1, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

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

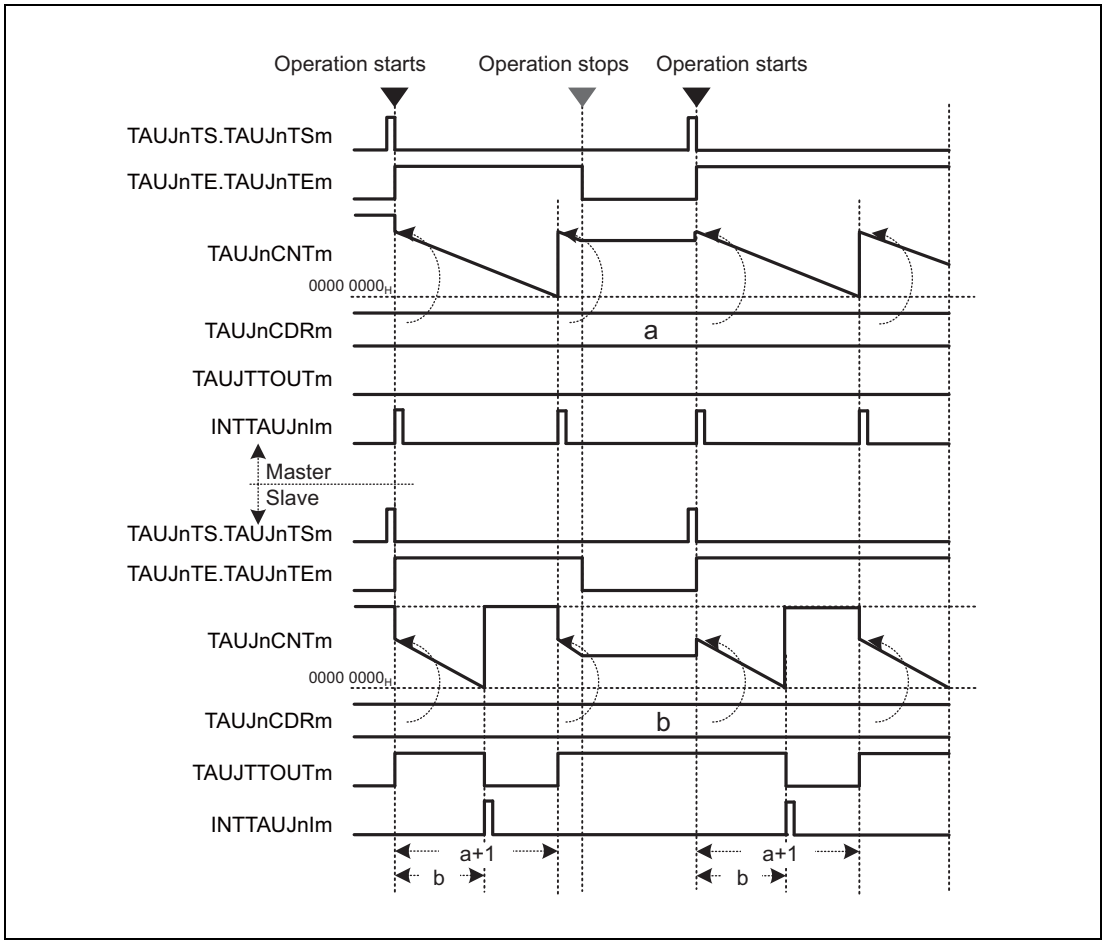


Figure 24.50 Stop and Restart Operation, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

- 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 25 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/F1L 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.

25.1 Features of RH850/F1L RTCA

25.1.1 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 25.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	—	—	—	—	1	1
Name	—	—	—	—	RTCAn (n = 0)	RTCAn (n = 0)

Table 25.2 Index

Index	Meaning
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.

25.1.2 Register Base Address

RTCAn base addresses are listed in the following table.

RTCAn register addresses are given as offsets from the base addresses in general.

Table 25.3 Register Base Address

Base Address Name	Base Address
<RTCA0_base>	FFE7 8000_H

25.1.3 Clock Supply

The RTCAn clock supply is shown in the following table.

Table 25.4 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
RTCA0	RTCATCKI	CKSCLK_ARTCA
	PCLK	CPUCLK2

25.1.4 Interrupt Requests

RTCAn interrupt requests are listed in the following table.

Table 25.5 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
RTCA0			
RTCATINT1S	1-second interval interrupt	201	—
RTCATINTAL	Alarm interrupt	202	—
RTCATINTR	Fixed interval interrupt	203	—

25.1.5 Reset Sources

RTCAn reset sources are listed in the following table. RTCAn is initialized by these reset sources.

Table 25.6 Reset Sources

Unit Name	Reset Source
RTCA0	Power-up reset (PURES)

25.1.6 External Input/Output Signals

External input/output signals of RTCAn are listed below.

Table 25.7 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
RTCA0		
RTCAT1HZ	1-Hz pulse output	RTCA0OUT*1

Note 1. RTCA0OUT is connected to TAUJ0. For details, see **Section 24, Timer Array Unit J (TAUJ)**.

25.2 Overview

25.2.1 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

25.2.2 Block Diagram

The block diagram shows the main components of the RTCA.

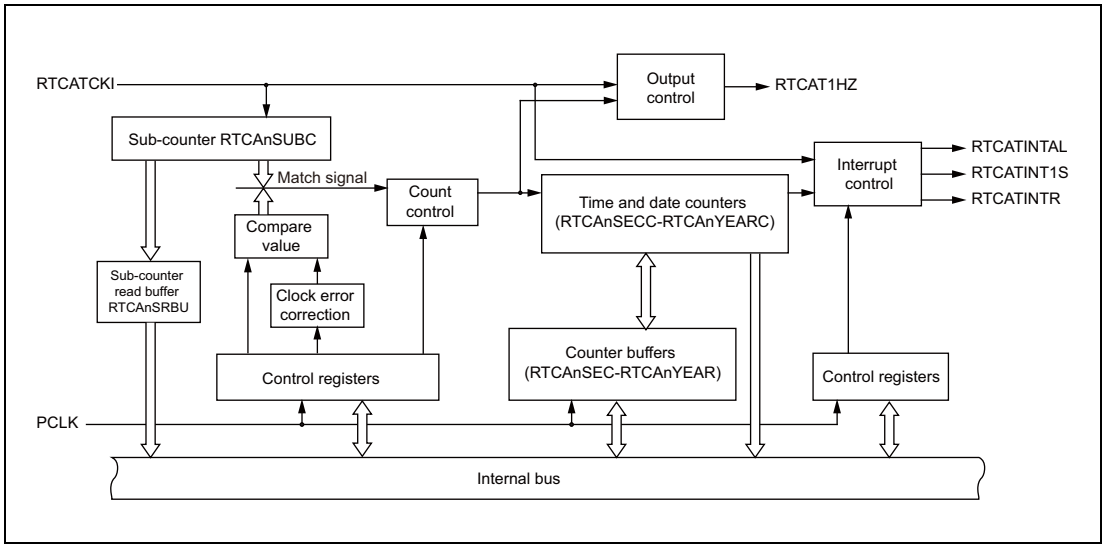


Figure 25.1 Block Diagram of the RTCA

25.2.3 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 zero 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 zero and triggers the seconds counter RTCAnSECC (and, if desired, the interrupt RTCATINT1S).

The sub-counter can generate a fixed interval interrupt every 0.25 seconds, 0.5seconds, 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 format.

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 25.5.3, 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 25.5.3, Reading Clock Counters** and **Section 25.5.2, Updating Clock Counters**.

The RTCAnTIMEC and RTCAnCALC registers and their corresponding buffer registers can be used to access 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.

25.3 Registers

25.3.1 List of Registers

RTCA registers are listed in the following table.

<RTCAn_base> is defined in **Section 25.1.2, Register Base Address**.

Table 25.8 Registers

Module	Register	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	Clock error correction register	RTCAnSUBU	<RTCAn_base> + 38_H
RTCAn	Sub-count register read buffer	RTCAnSRBU	<RTCAn_base> + 10_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 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

25.3.2 Details of RTCA Control Registers

25.3.2.1 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/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 25.9 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 25.5.1, 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 25.21, 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 25.4, 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 25.5.1, Initial Setting of the RTCA**.
3 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

25.3.2.2 RTCAnCTL1 — RTCA Control Register 1

This register controls the interrupt request generation and the 1-Hz pulse output.

Access: This register can be read/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 25.10 RTCAnCTL1 Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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 0 minutes 0 seconds
110	Every day	Every 1 day 0 hours 0 minutes 0 seconds (i.e., every midnight)
111	Every month	Every 1 month first day 0 hours 0 minutes 0 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.

25.3.2.3 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/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 25.11 RTCAnCTL2 Register Contents (1/2)

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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 25.5.5, 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 25.5.6, 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 25.5.4, 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 RTCA0SRBU when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1). The value of RTCA0SUBC is synchronized with RTCATCKI and loaded to RTCA0SRBU. For details, see **Section 25.5.4, 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 25.11 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.

25.3.3 Details of RTCA Sub-Counter Registers

25.3.3.1 RTCAnSUBC — RTCA Sub-Count Register

This counter counts the 1-second reference time. It operates using the count clock RTCATCKI.

Access: This register can only 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 25.12 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 25.5.4, Reading RTCAnSRBU**.
2. The count-operation of this sub-counter depends on the selected operation mode. See **Section 25.4, Operation**, for details.

25.3.3.2 RTCAnSRBU — RTCA Sub-Count Register Read Buffer

This register is the read buffer for the sub-counter RTCAnSUBC.

Access: This register can only 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 25.13 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 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 25.5.4, Reading RTCAnSRBU**.

25.3.3.3 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 25.4.4, Clock Error Correction**.

Access: This register can be read/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 25.14 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 25.5.1, Initial Setting of the RTCA**, and
   - **Section 25.5.5, Writing to RTCAnSUBU**.

25.3.3.4 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/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 25.15 RTCAnSCMP Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing to these bits, 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 25.5.1, Initial Setting of the RTCA** and **Section 25.5.6, Writing to RTCAnSCMP**.

25.3.4 Details of RTCA Clock Counter and Buffer Registers

25.3.4.1 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 can only 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 25.16 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 25.5.3, 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 25.5.1, Initial Setting of the RTCA**, and
   - **Section 25.5.2, Updating Clock Counters**

25.3.4.2 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/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 25.17 RTCAnSEC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, 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 25.5.1, Initial Setting of the RTCA,**
   - **Section 25.5.2, Updating Clock Counters**, and
   - **Section 25.5.3, Reading Clock Counters.**

25.3.4.3 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 can only 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 25.18 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 25.5.3, 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 25.5.1, Initial Setting of the RTCA**, and
   - **Section 25.5.2, Updating Clock Counters.**

25.3.4.4 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/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 25.19 RTCAnMIN Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, 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 25.5.1, Initial Setting of the RTCA,**
   - **Section 25.5.2, Updating Clock Counters**, and
   - **Section 25.5.3, Reading Clock Counters.**

25.3.4.5 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 25.21, 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 can only 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 25.20 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 25.21, 12- and 24-Hour Format**, for details.

NOTES

1. Perform RTCAnHOURC read according to the flow described in **Section 25.5.3, 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 25.5.1, Initial Setting of the RTCA**, and
   - **Section 25.5.2, 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 25.21 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

25.3.4.6 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/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 25.22 RTCAnHOUR Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5 to 0	RTCAnHOUR [5:0]	Hours in BCD See **Table 25.21, 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 25.5.1, Initial Setting of the RTCA,**
   - **Section 25.5.2, Updating Clock Counters**, and
   - **Section 25.5.3, Reading Clock Counters.**

25.3.4.7 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 can only 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 25.23 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 25.5.3, 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 calender count buffer register RTCAnCAL. See
   - **Section 25.5.1, Initial Setting of the RTCA**, and
   - **Section 25.5.2, Updating Clock Counters.**

25.3.4.8 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/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 25.24 RTCAnWEEK Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, 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 25.5.1, Initial Setting of the RTCA,**
   - **Section 25.5.2, Updating Clock Counters**, and
   - **Section 25.5.3, Reading Clock Counters.**

25.3.4.9 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 of the day 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 can only 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 25.25 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 25.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the day count buffer register RTCAnDAY or to the calender count buffer register RTCAnCAL. See
   - **Section 25.5.1, Initial Setting of the RTCA**, and
   - **Section 25.5.2, Updating Clock Counters.**

25.3.4.10 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/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 25.26 RTCAnDAY Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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 25.5.1, Initial Setting of the RTCA,**
   - **Section 25.5.2, Updating Clock Counters**, and
   - **Section 25.5.3, Reading Clock Counters.**

25.3.4.11 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 can only 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 25.27 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 25.5.3, 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 calender count buffer register RTCAnCAL. See
   - **Section 25.5.1, Initial Setting of the RTCA**, and
   - **Section 25.5.2, Updating Clock Counters.**

25.3.4.12 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/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 25.28 RTCAnMONTH Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing to these bits, 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 25.5.1, Initial Setting of the RTCA,**
   - **Section 25.5.2, Updating Clock Counters**, and
   - **Section 25.5.3, Reading Clock Counters.**

25.3.4.13 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 can only 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 25.29 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 25.5.3, 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 calender count buffer register RTCAnCAL. See
   - **Section 25.5.1, Initial Setting of the RTCA**, and
   - **Section 25.5.2, Updating Clock Counters.**

25.3.4.14 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/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 25.30 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 25.5.1, Initial Setting of the RTCA,**
   - **Section 25.5.2, Updating Clock Counters**, and
   - **Section 25.5.3, Reading Clock Counters.**

25.3.5 Details of RTCA Special Counter and Buffer Registers

25.3.5.1 RTCAnTIMEC — RTCA Time Count Register

This register enables the RTCAnHOURC, RTCAnMINC, and RTCAnSECC counters to be read simultaneously.

Access: This register can only 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 25.31 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 25.21, 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 25.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the time count buffer register RTCAnTIME. See
   - **Section 25.5.1, Initial Setting of the RTCA**, and
   - **Section 25.5.2, Updating Clock Counters.**

25.3.5.2 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/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 25.32 RTCAnTIME Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
21 to 16	RTCAnHOUR [5:0]	Hours in BCD See **Table 25.21, 12- and 24-Hour Format**, for details.
15	Reserved	When read, the value after reset is returned. When writing to this bit, 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 to this bit, write the value after reset.
6 to 0	RTCAnSEC [6:0]	Seconds in BCD

NOTE

Perform RTCAnTIME read/write as described in

- **Section 25.5.1, Initial Setting of the RTCA,**
- **Section 25.5.2, Updating Clock Counters**, and
- **Section 25.5.3, Reading Clock Counters.**

25.3.5.3 RTCAnCALC — RTCA Calendar Count Register

This register enables the RTCAnYEARC, RTCAnMONC, RTCAnDAYC, and RTCAnWEEKC counters to be read simultaneously.

Access: This register can only 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 25.33 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 25.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the clock time setting register RTCAnCAL. See
   - **Section 25.5.1, Initial Setting of the RTCA**, and
   - **Section 25.5.2, Updating Clock Counters.**

25.3.5.4 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/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 25.34 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 to these bits, 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 to these bits, 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 to these bits, 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 25.5.1, Initial Setting of the RTCA.**
- **Section 25.5.2, Updating Clock Counters**, and
- **Section 25.5.3, Reading Clock Counters.**

25.3.6 Details of RTCA Alarm Setting Registers

25.3.6.1 RTCAnALM — RTCA Alarm Minute Setting Register

This register specifies the minute of the alarm interrupt.

For details and example settings, see **Section 25.4.3, Alarm Interrupt Function**.

Access: This register can be read/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 25.35 RTCAnALM Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, 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.

25.3.6.2 RTCAnALH — RTCA Alarm Hour Setting Register

This register specifies the hour of the alarm interrupt.

For details and example settings, see **Section 25.4.3, Alarm Interrupt Function**.

Access: This register can be read/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 25.36 RTCAnALH Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, 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.

25.3.6.3 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/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 25.37 RTCAnALW Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, 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.
- To set the alarm for Monday and Wednesday, set RTCAnALW = 0000 1010.
- To set the alarm for Tuesday, Thursday, and Saturday, set RTCAnALW = 0101 0100.

For more examples, see **Section 25.4.3, Alarm Interrupt Function**.

25.3.7 RTCA Emulation Register

25.3.7.1 RTCAnEMU — RTCA Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read/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 25.38 RTCAnEMU Register Contents

Bit Position	Bit Name	Function
7	RTCAnSVSDIS	When the EPC.SVSTOP bit is set to 0: Count clock is supplied 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 (as at a breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

25.4 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 25.39 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 4.194304MHz	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 25.5.1, 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).**

25.4.1 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 25.40 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

25.4.2 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, or 1 (full) month.

The fixed interval interrupt function is controlled by bits RTCAnCTL1.RTCAnCT[2:0].

25.4.3 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 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 BCD0.

Examples

The following tables show some examplary 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 25.41 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 25.42 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

25.4.4 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 25.2, 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.

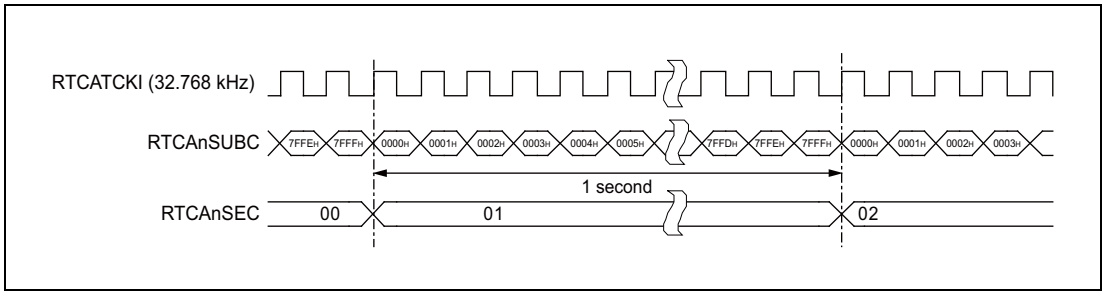


Figure 25.2 RTCATCKI = 32.768 kHz, No Clock Error Correction Required

RTCATCKI = 32.769 kHz

Figure 25.3, 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)

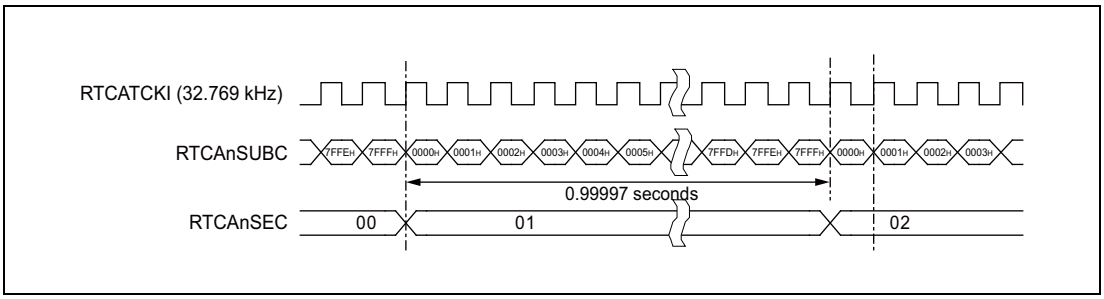


Figure 25.3 RTCATCKI = 32.769 kHz, No Clock Error Correction Enabled

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.

25.4.4.1 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 25.43 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 25.44 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

25.4.4.2 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 25.45 Setting of Bit RTCAnSUBU.RTCAnDEV

RTCAn DEV	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

25.4.4.3 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 25.46 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 25.47 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

25.5 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.

25.5.1 Initial Setting of the RTCA

The RTCA must be stopped before setting the initial setting value of each counter.

25.5.1.1 RTCA Stop Procedure

Stop the RTCA according to the following flow.

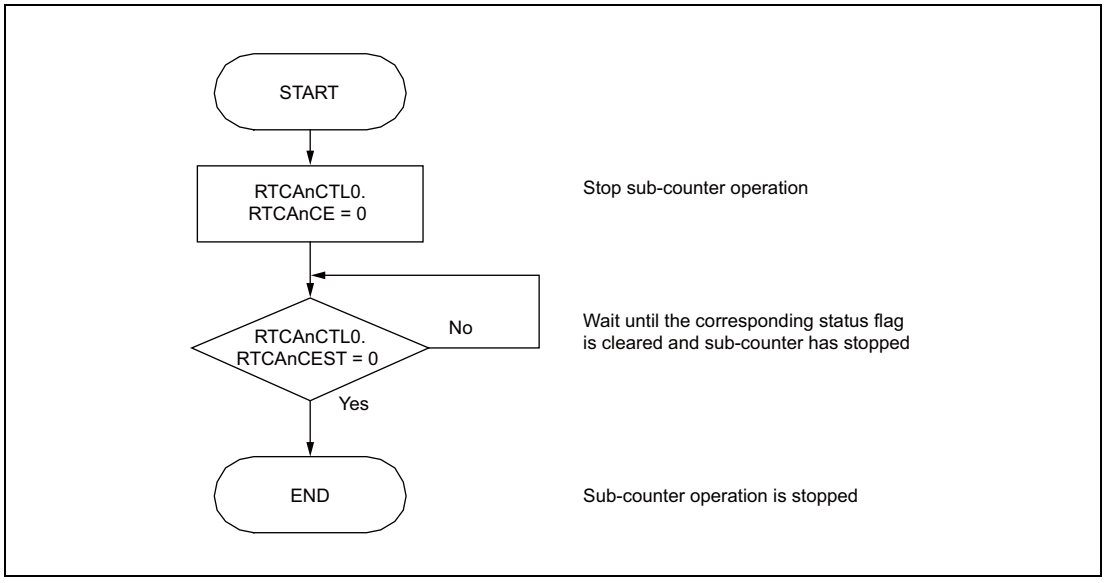


Figure 25.4 RTCA Stop Procedure

25.5.1.2 RTCA Initialization Procedure

Perform the initial setting of the RTCA according to the following flow:

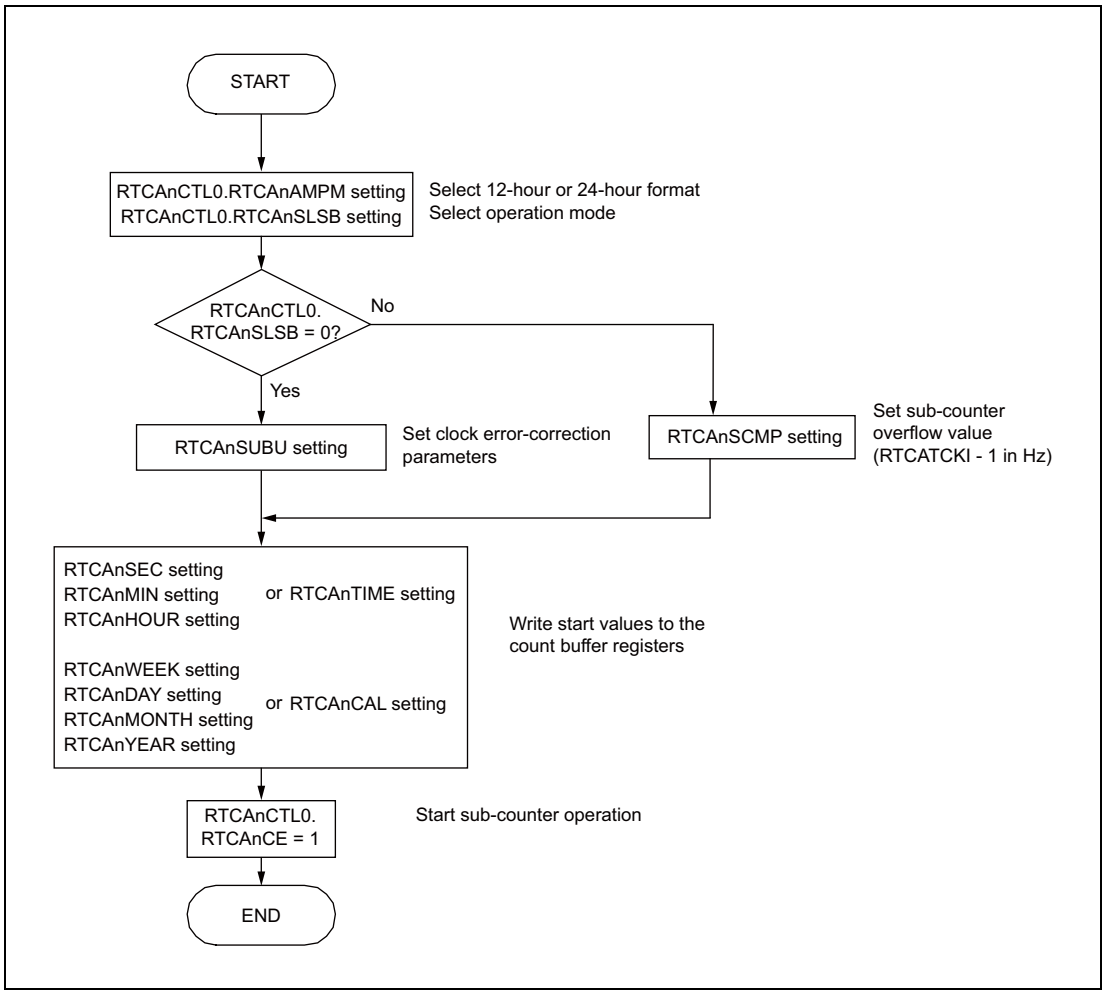


Figure 25.5 RTCA Initial Setup Procedure

CAUTION

The internal clock counter is synchronized with RTCATCKI.

In addition, two RTCATCKI periods are required until the completion of the above initial setting.

Therefore, PCLK must be continuously supplied until the completion of the initial setting.

Check that RTCA0CEST = 1, when the supply of PCLK is stopped after setting the initial setting value of RTCA.

25.5.2 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.

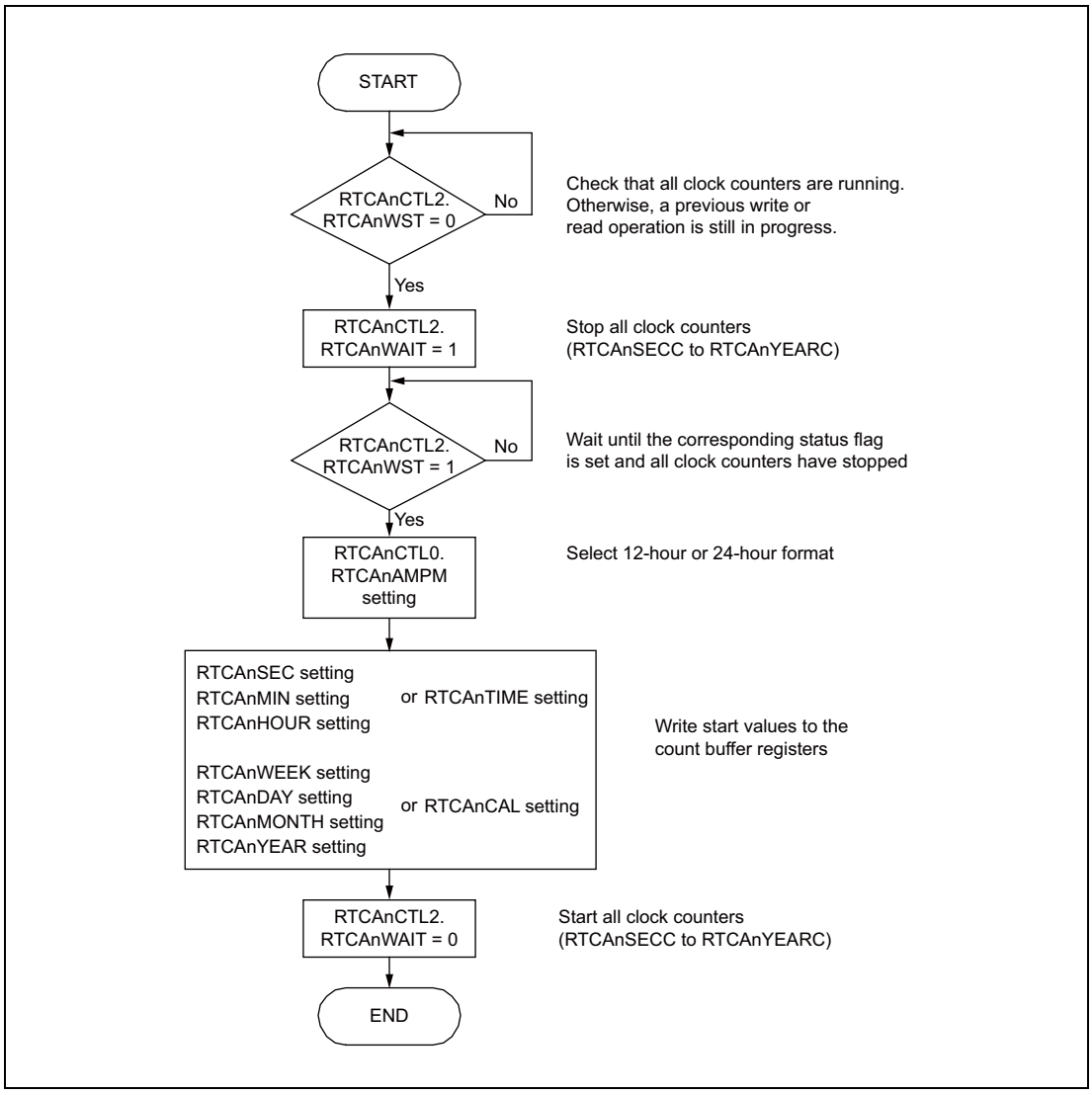


Figure 25.6 Updating Clock Counter Values

CAUTIONS

1. **The internal clock counter is synchronized with RTCATCKI.**
   In addition, two RTCATCKI periods are required until the completion of the clock counter updating.
   Therefore, PCLK must be continuously supplied until the completion of the clock counter updating.
   Check that RTCA0WST = 0, when the supply of PCLK is stopped after the completion of the 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.**

25.5.3 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 25.48 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.

25.5.3.1 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 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 = 80 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.

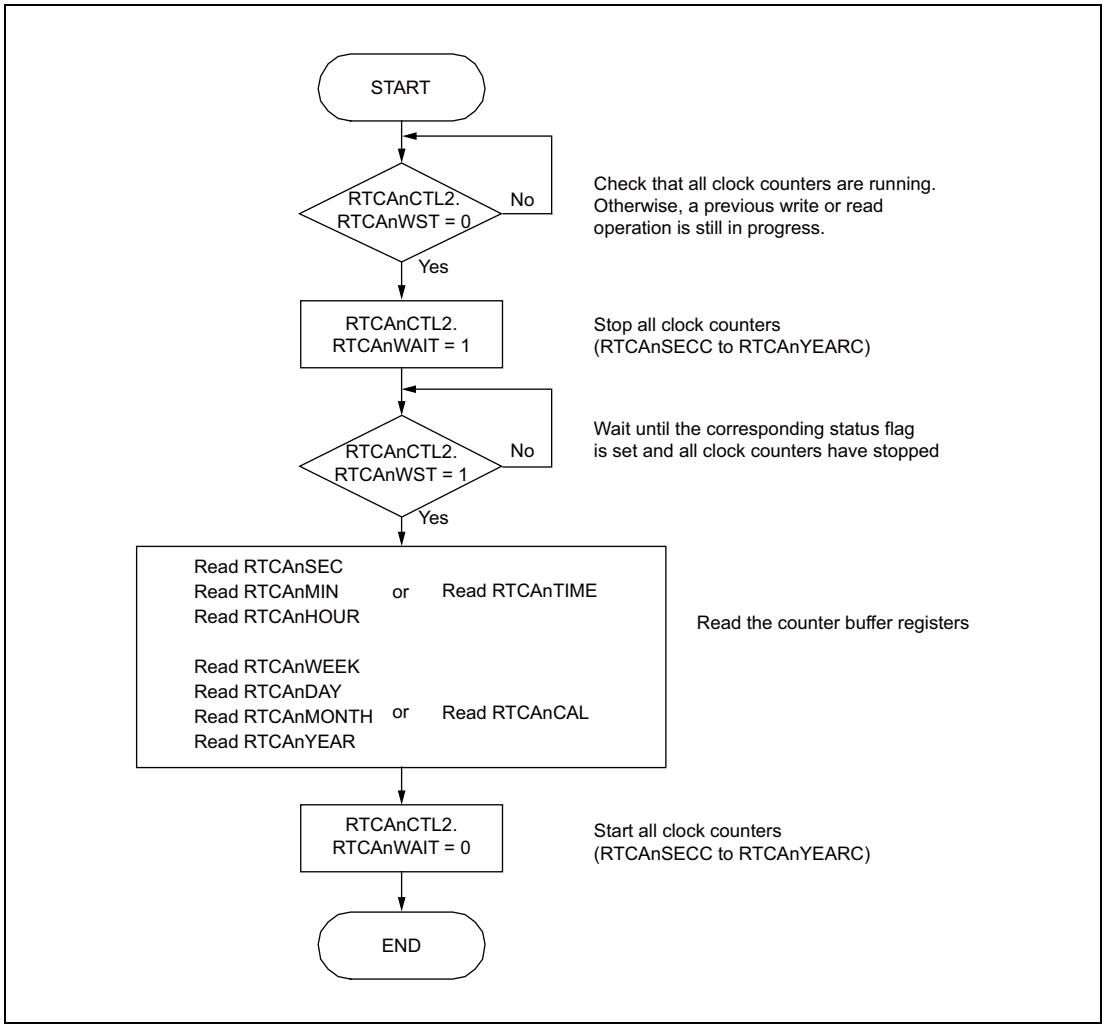


Figure 25.7 Reading Clock Count Buffer Registers

CAUTIONS

1. **The internal clock counter is synchronized with RTCATCKI.**
   In addition, two RTCATCKI periods are required until the completion of the above count buffer register reading.
   Therefore, PCLK must be continuously supplied until the completion of the count buffer register reading.
   Check that RTCA0CEST = 1 first to stop the supply of PCLK after the 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 and increment the seconds counter after restarting the clock counters.**
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.**

25.5.3.2 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.

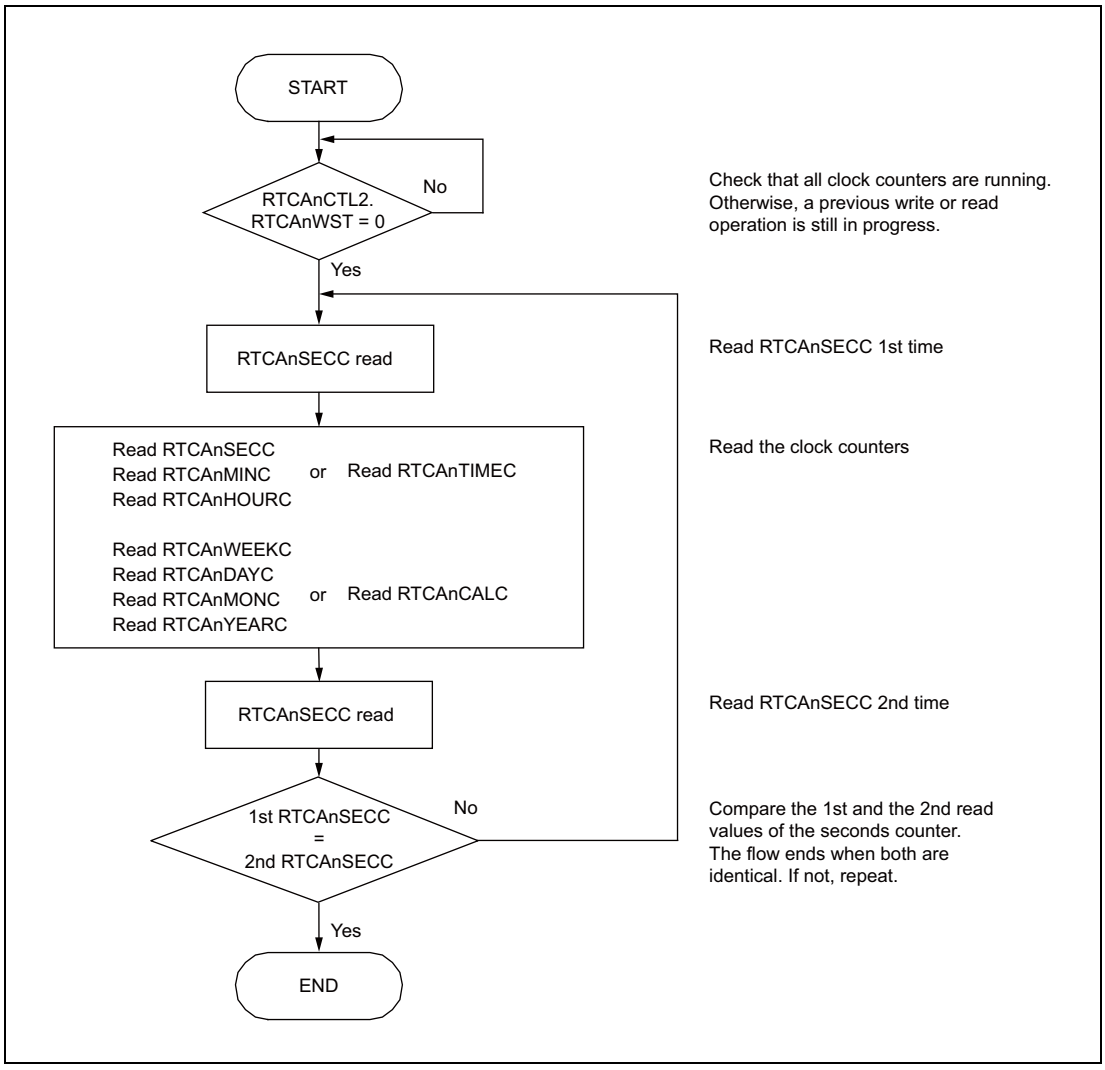


Figure 25.8 Reading Clock Counter Registers

NOTE

The procedure must be completed within one second.

25.5.4 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.

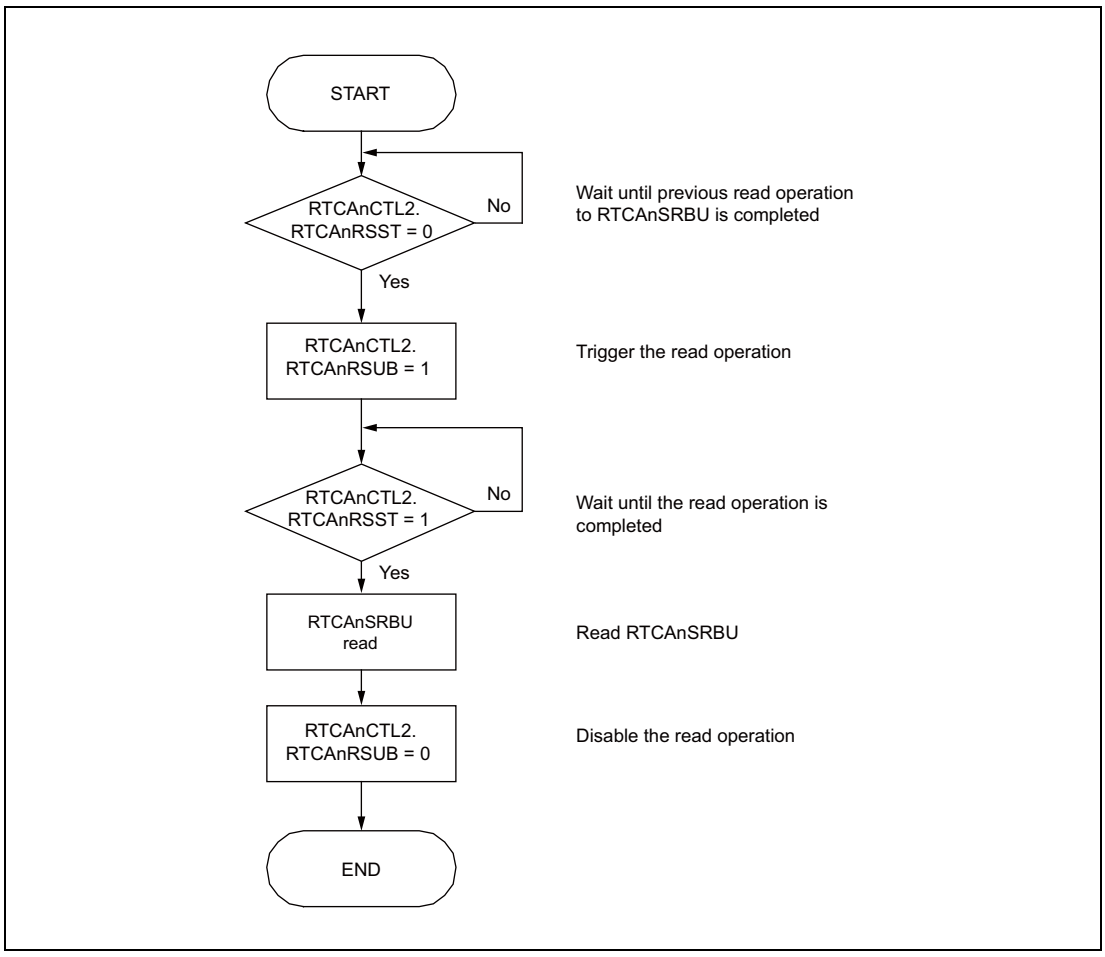


Figure 25.9 Reading the RTCAnSRBU Register

25.5.5 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.

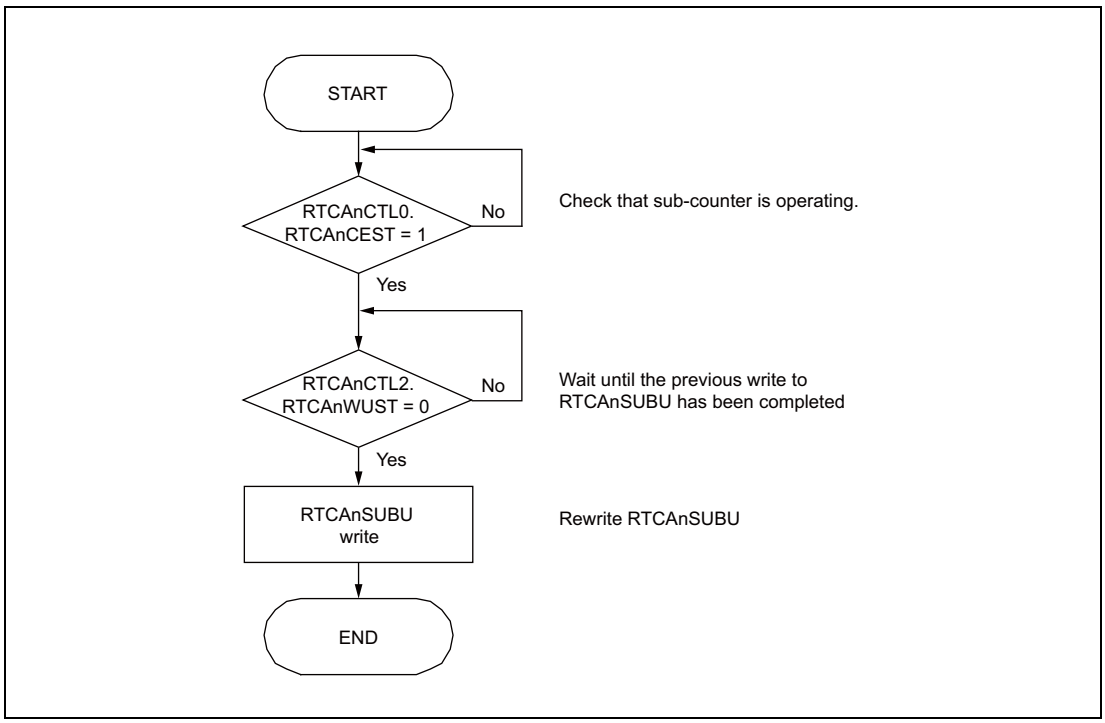


Figure 25.10 Writing to the RTCAnSUBU Register

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).

25.5.6 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.

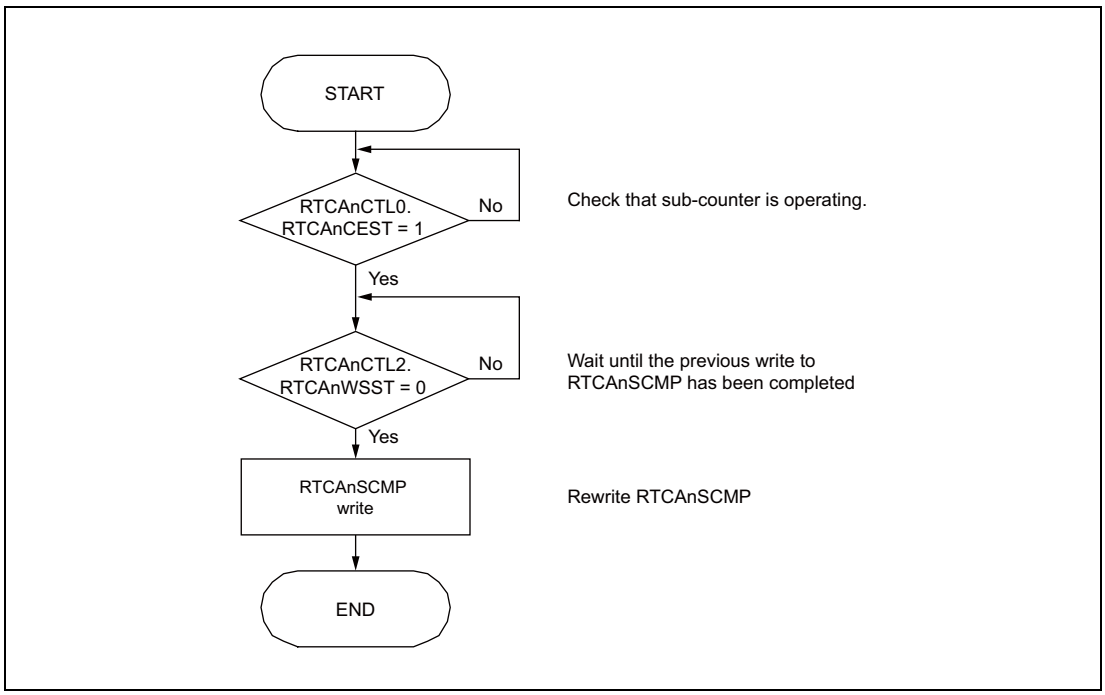


Figure 25.11 Writing the RTCAnSCMP Register

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).

25.6 Timing Diagrams

25.6.1 Timing of Counter Start

The following diagram illustrates the counter start after setting the time in the buffer registers.

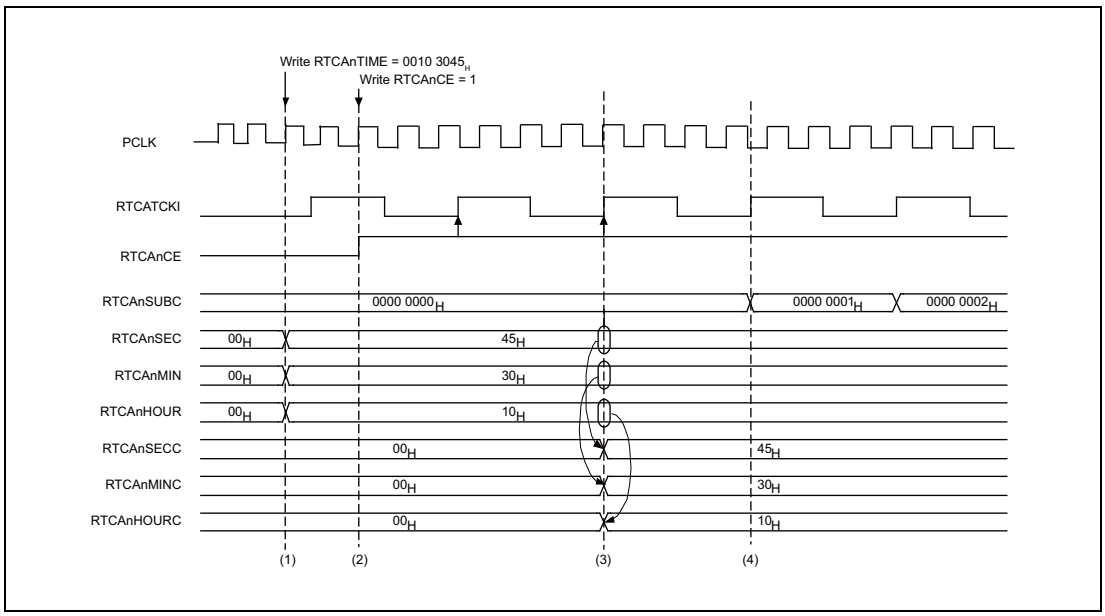


Figure 25.12 Counter Start Timing

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.

25.6.2 Timing of Clock Counter Update while Counter Is Enabled

The following diagram illustrates the counter restart after setting the time in the buffer registers.

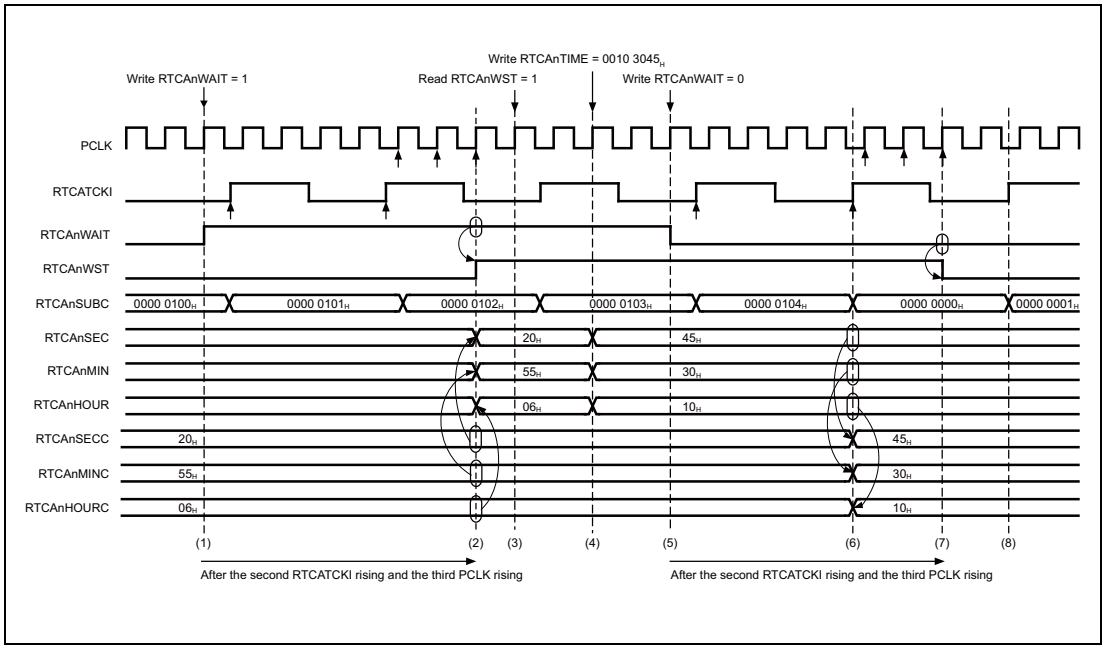


Figure 25.13 Clock Counter Update Timing

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.

25.6.3 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.

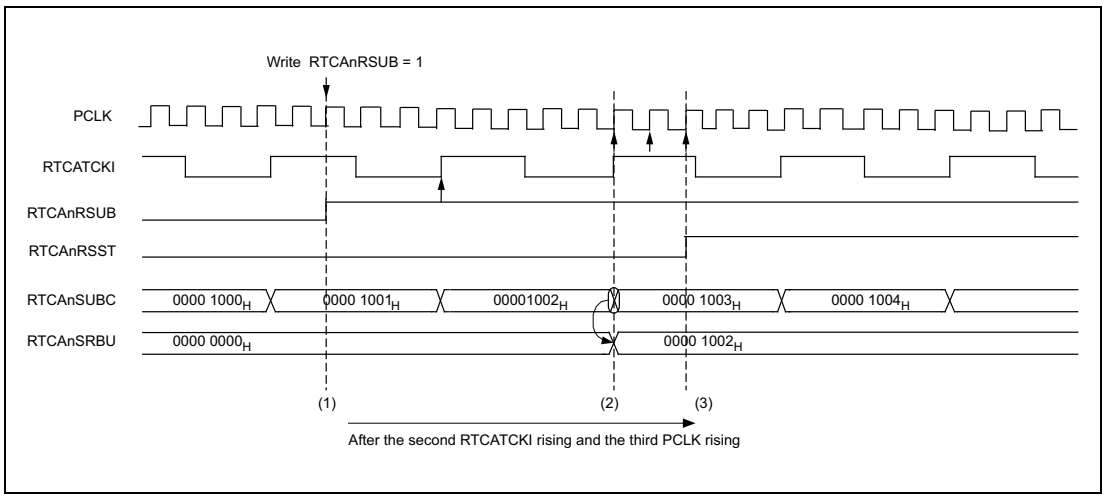


Figure 25.14 Timing when Reading the Sub-Counter Read Buffer Register Value

The timing diagram above shows the following:

(1) Setting RTCAnRSUB = 1 triggers loading of the sub-counter value to RTCA0SRBU.

(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 now be read.

Section 26 Encoder Timer (ENCA)

This section contains a generic description of the Encoder Timer (ENCA).

The first part in this section describes all RH850/F1L 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.

26.1 Features of RH850/F1L ENCA

26.1.1 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 26.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1					
Name	ENCAn (n = 0)					

Table 26.2 Index

Index	Meaning
n	Throughout this section, the individual ENCA units are identified by the index “n” (n = 0); for example, ENCAnCTL is the ENCAn control register.

26.1.2 Register Base Address

ENCAn base addresses are listed in the following table.

ENCAn register addresses are given as offsets from the base addresses in general.

Table 26.3 Register Base Address

Base Address Name	Base Address
<ENCA0_base>	FFE8 0000_H

26.1.3 Clock Supply

The ENCAn clock supply is shown in the following table.

Table 26.4 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
ENCA0	PCLK	CKSCLK_IPERI1

26.1.4 Interrupt Requests

ENCAn interrupt requests are listed in the following table.

Table 26.5 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
ENCA0			
ENCATIOV	Overflow interrupt	77	—
ENCATIUD	Underflow interrupt	78	—
ENCATINT0	Capture/compare match interrupt 0	79	—
ENCATINT1	Capture/compare match interrupt 1	80	—
ENCATIEC	Encoder clear interrupt	81	—

26.1.5 Reset Sources

ENCAn reset sources are listed in the following table. ENCAn is initialized by these reset sources.

Table 26.6 Reset Sources

Unit Name	Reset Source
ENCA0	All reset sources (ISORES)

26.1.6 External Input/Output Signals

External input/output signals of ENCAn are listed below.

Table 26.7 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
ENCA0		
ENCATTIN0	ENCAn capture trigger input 0	ENCA0TIN0
ENCATTIN1	ENCAn capture trigger input 1	ENCA0TIN1
ENCAnE0	ENCAn encoder input 0	ENCA0E0
ENCAnE1	ENCAn encoder input 1	ENCA0E1
ENCAnEC	ENCAn encoder clear input	ENCA0EC

26.1.7 Internal Input/Output Signals

Input/output signals to be connected between ENCA and PIC are listed below.

Table 26.8 Internal Input/Output Signals

Unit Signal Name	Outline	Connected to
ENCATSST	Simultaneous start trigger	PIC
ENCATTIN1	ENCAn capture trigger input 1	PIC

26.2 Overview

26.2.1 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 applied to the timer counter clear condition
- Edge or level for clearing the encoder input signal of the timer counter clear condition can be selected
- 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.

26.2.2 Block Diagram

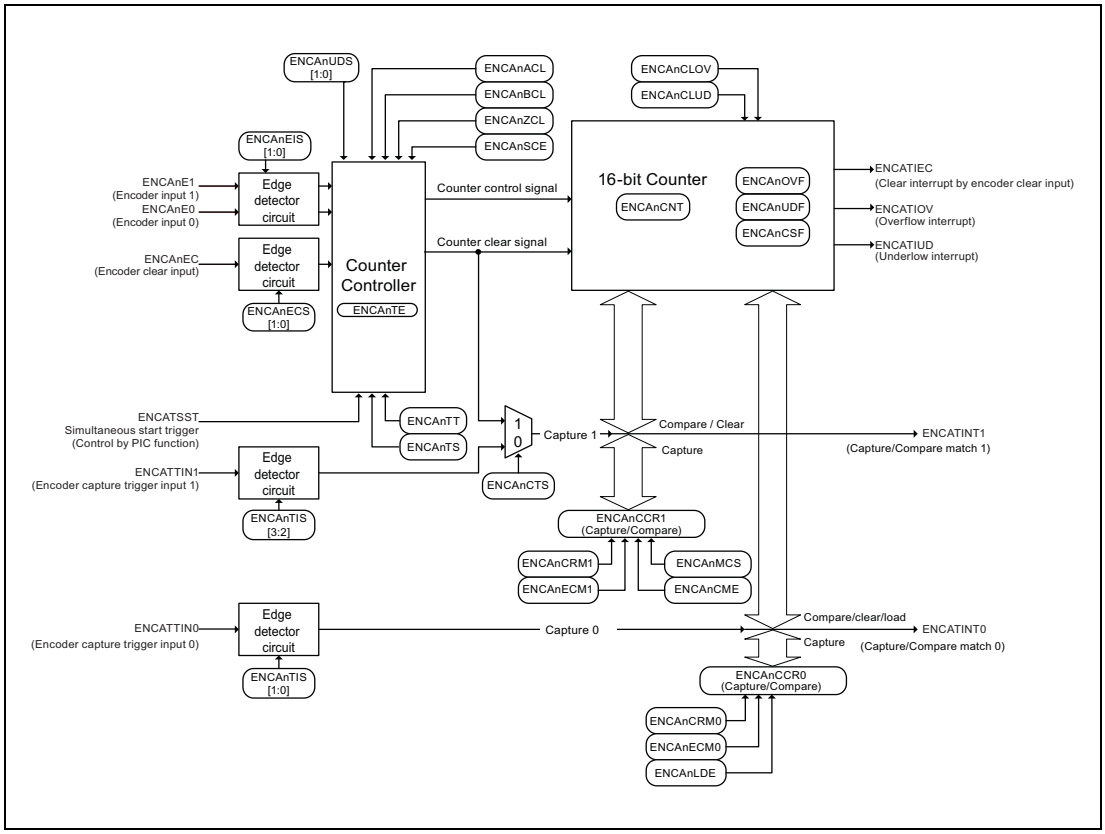


Figure 26.1 ENCA Block Diagram

26.3 Registers

26.3.1 List of Registers

ENCA registers are listed in the following table.

<ENCAn_base> is defined in **Section 26.1.2, Register Base Address**.

Table 26.9 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

26.3.2 ENCAnCTL — ENCAn Control Register

This register is used to configure various operation settings of the Encoder Timer.

Access: This register can be read/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 26.10 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 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 to these bits, 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 register 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: Uses ENCAnEC of the encoder clear input signal as the capture trigger for the ENCAnCCR1 register.
6, 5	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

Table 26.10 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]	UPDOWN Count Selection 1 and 0 This is the counter up/down control bit 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.

26.3.3 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/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 26.11 ENCAnIOC0 Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, 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

26.3.4 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/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 26.12 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 26.12 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

26.3.5 ENCAnFLG — ENCAn Status Flag Register

This register holds the status flags of the timer counter of ENCAn.

Access: This register can only 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 26.13 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 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 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.

26.3.6 ENCAnFGC — ENCAn Status Flag Clear Register

This register is used to clear the timer counter status flags of ENCAnFLG.

Access: This register can only 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 26.14 ENCAnFGC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing to these bits, 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).

26.3.7 ENCAnCCR0 — ENCAn Capture/Compare Register 0

This register is a 16-bit capture/compare register 0.

Access: This register can be read/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

Table 26.15 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.

26.3.8 ENCAnCCR1 – ENCAn Capture/Compare Register 1

This register is a 16-bit capture/compare register 1.

Access: This register can be read/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 26.16 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.

26.3.9 ENCAnCNT — ENCAn Counter Register

This register is the 16-bit timer counter register.

Access: This register can be read/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 26.17 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.

26.3.10 ENCAnTE — ENCAn Timer Enable Status Register

This register indicates the operating status of ENCAn.

Access: This register can only 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 26.18 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

26.3.11 ENCAnTS — ENCAn Timer Start Trigger Register

This register provides the trigger bit for setting the ENCAn to the operation enabled state.

Access: This register can only 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 26.19 ENCAnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, 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.

26.3.12 ENCAnTT — ENCAn Timer Stop Trigger Register

This register provides the trigger bit for setting the ENCAn to the operation stopped state.

Access: This register can only 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 26.20 ENCAnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, 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.

26.3.13 ENCAnEMU — ENCAn Emulation Register

This register controls operations by SVSTOP.

Access: This register can be read/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 26.21 ENCAnEMU Register Contents

Bit Position	Bit Name	Function
7	ENCAnSVSDIS	• (When EPC.SVSTOP bit = 0) The count clock is provided continuously when the debugger has control of the microcontroller (by break points, 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 has control of the microcontroller (by break points, etc.). 1: The count clock is provided continuously when the debugger has control of the microcontroller (by break points, etc.).
6 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

26.4 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.

26.4.1 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.

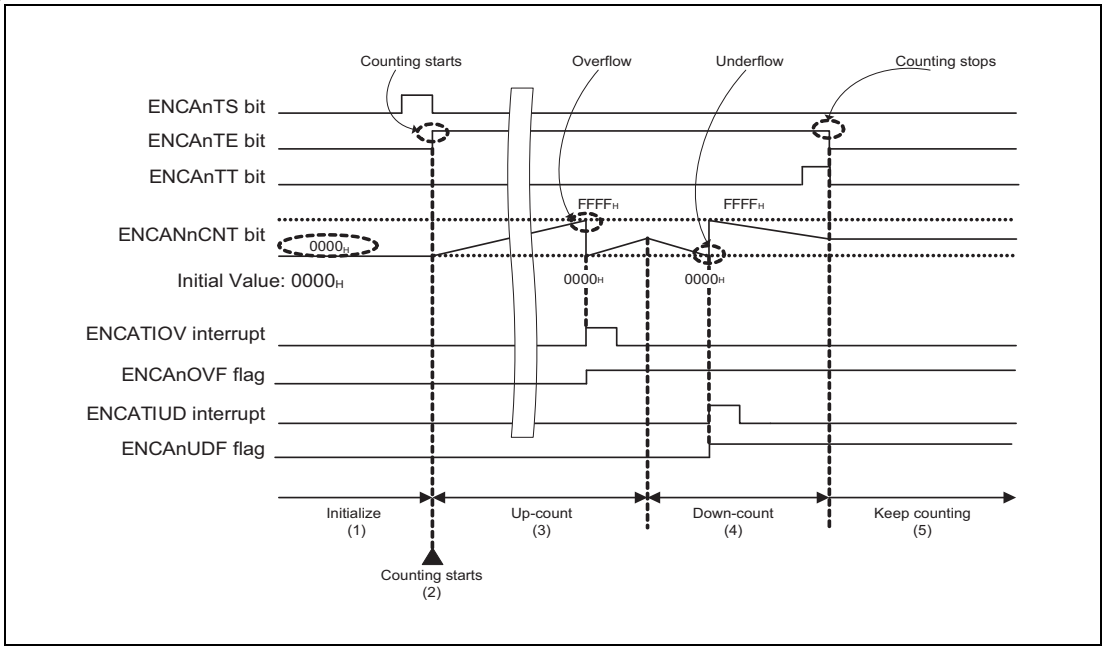


Figure 26.2 Timer Counter Initial Value Setting/Start/Stop

(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 26.6.1, 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 26.6.2, 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.

26.4.2 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.

26.4.2.1 When the ENCAnUDS1/ENCAnUDS0 Bits in the ENCAnCTL Register = 00_B

Table 26.22 When ENCAnUDS1/ENCAnUDS0 Bits = 00_B

		Operation Description		
ENCAnUDS1	**ENCAnUDS0**	**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 effective edges and levels of ENCAnE0 and ENCAnE1 overlap.

The following timing chart shows the count operation when ENCAnUDS1 and ENCAnUDS0 bits = 00_B.

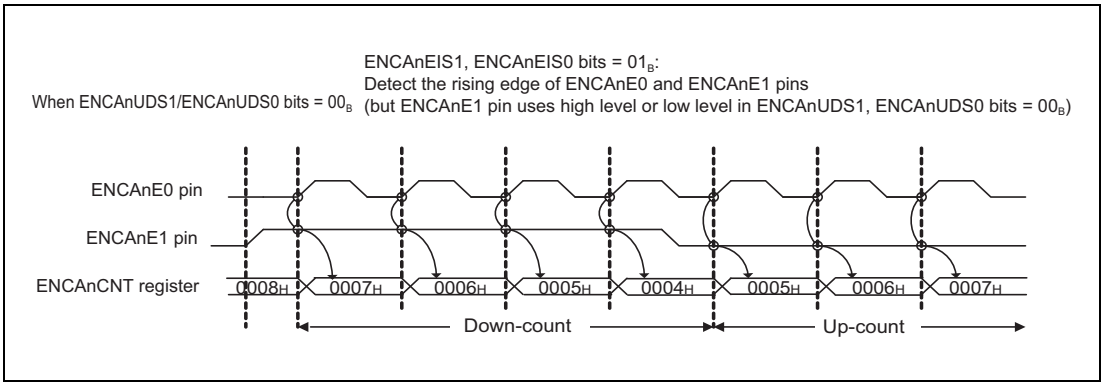


Figure 26.3 Count Operation when the ENCAnUDS1/ENCAnUDS0 Bits in the ENCAnCTL Register = 00_B

26.4.2.2 When the ENCAnUDS1/ENCAnUDS0 Bits in the ENCAnCTL Register = 01_B

Table 26.23 When the ENCAnUDS1/ENCAnUDS0 Bits = 01_B

		Operation Description		
ENCAnUDS1	**ENCAnUDS0**	**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 of ENCAnE0/ENCAnE1 pins and levels are matched, 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.

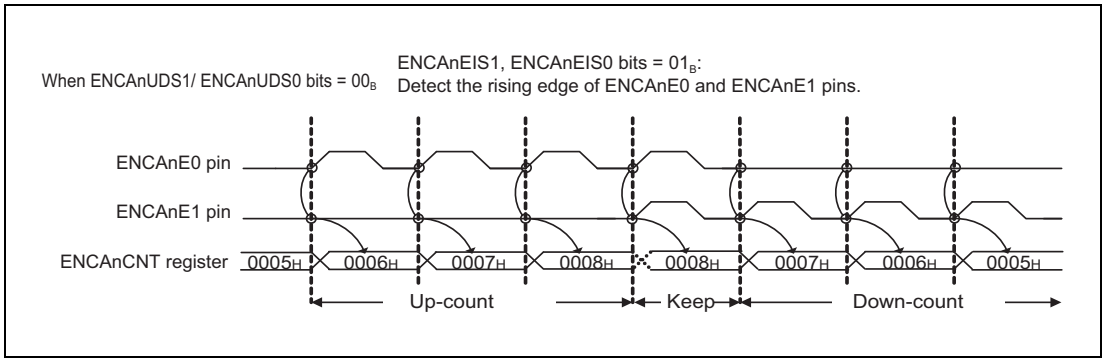


Figure 26.4 Count Operation when the ENCAnUDS1/ENCAnUDS0 Bits in the ENCAnCTL Register = 01_B

26.4.2.3 When the ENCAnUDS1 and ENCAnUDS0 Bits in the ENCAnCTL Register = 10_B

Table 26.24 When the ENCAnUDS1, ENCAnUDS0 Bits = 10_B

		Operation Description		
ENCAnUDS1	**ENCAnUDS0**	**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 edges for ENCAnE0 and ENCAnE1 are specified by setting ENCAnEIS1 and ENCAnEIS0.

The following timing chart shows the count operation when the ENCAnUDS1/ENCAnUDS0 bits = 10_B.

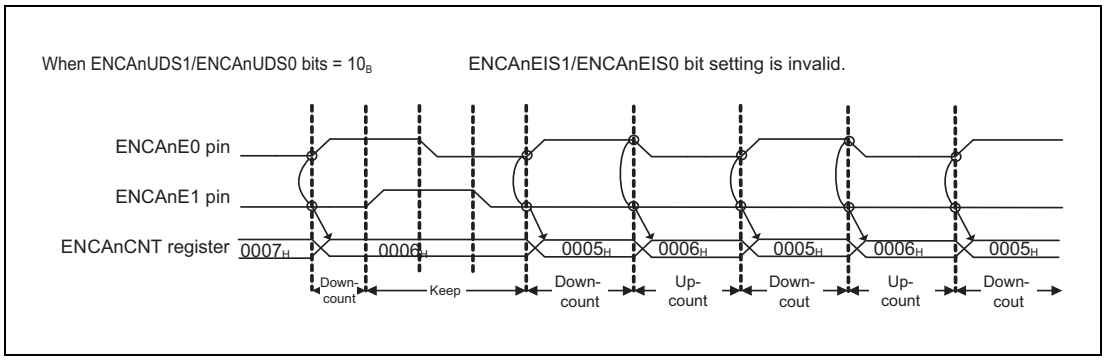


Figure 26.5 Count Operation when ENCAnUDS1/ENCAnUDS0 Bits = 10_B

26.4.2.4 When ENCAnUDS1 / ENCAnUDS0 Bits in the ENCAnCTL Register = 11_B

Table 26.25 When ENCAnUDS1/ENCAnUDS0 Bits = 11_B

		Operation Description		
ENCAnUDS1	**ENCAnUDS0**	**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.

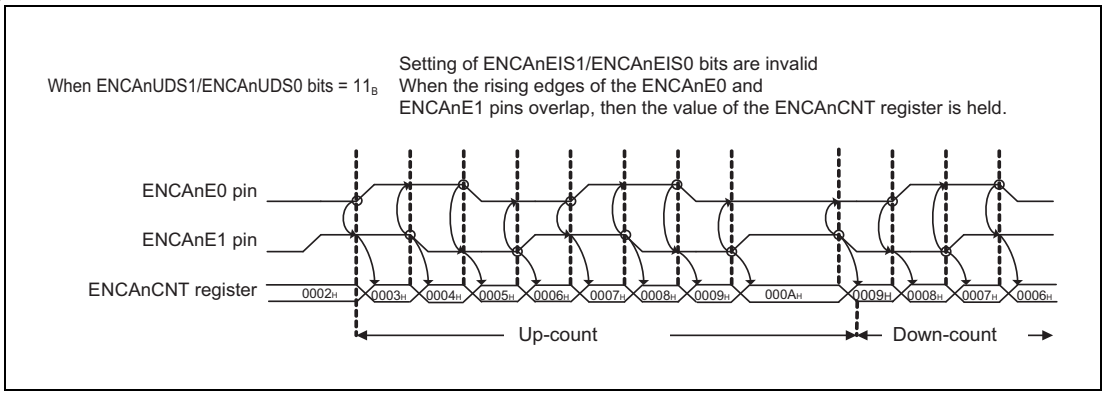


Figure 26.6 Count Operation when ENCAnUDS1 and ENCAnUDS0 Bits in the ENCAnCTL Register = 11_B

26.4.3 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 26.26 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

26.4.3.1 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 26.6.19, Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0**.

26.4.3.2 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 26.27 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

26.4.4 Functions of ENCAnCCR0

26.4.4.1 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 26.28 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 26.6.8, Using the ENCAnLDE Function Immediately after Startup** to **Section 26.6.12, Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input**.

26.4.4.2 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 26.6.14, Capture Operation between Count Clocks (ENCAnCCR0)** and **Section 26.6.17, Encoder Operation when Compare Match Clear Control is Disabled**.

26.4.5 Functions of ENCAnCCR1

26.4.5.1 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 26.29 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 count 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 ENCAnMCS = 0, the compare 1 match interrupt mask function is disabled by a write operation to the ENCAnCCR1 register.
- When ENCAnMCS = 1, the compare 1 match interrupt mask function is disabled 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.
- When ENCAnMCS = 1 and ENCAnLDE = 1, the compare 1 match interrupt mask function is disabled by a loading operation of the ENCAnCCR0 register to the timer counter upon underflow detection.
- Setting ENCAnECM1 to “1” is prohibited when enabling the compare 1 match interrupt mask function.

Table 26.30 Compare Match Interrupt Mask Function

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)	

26.4.5.2 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 26.6.13, Capture Operation between Count Clocks (ENCAnCCR1)**.

The operations for each of the ENCAnCTS settings are shown in the table below.

Table 26.31 Operations for Each of the ENCAnCTS Settings

ENCAnCCR1 Function	Capture Trigger Selection			
ENCAnCRM1	ENCAnCTS	Capture Trigger Signal	Timer Counter Clearing	Interrupt Occurrence
1 (Capture)	0	Capture trigger 1 input (ENCATTIN1)	Does not clear timer counter.	(1) Capture 1 interrupt (ENCATINT1)
	1	Encoder clear input (ENCAnEC pin)	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 26.6.3, Count Clearing and Capture Operation by Encoder Clear Input (ENCAnEC Pin), **Section 26.6.4, Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)**, **Section 26.6.5, Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)**, **Section 26.6.11, Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input (ENCAnEC Pin)** and **Section 26.6.12, Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input**.

26.4.5.3 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 26.32 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).

26.4.6 Startup/Stop of Timer Counter

26.4.6.1 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 27.8, Simultaneous Start Trigger Function.

26.4.6.2 Stop of Timer

When the ENCAnTT bit is set to “1”, the ENCAnTE bit becomes “0” and the timer stops.

26.5 ENCAn Setting Sequences

26.5.1 ENCAn Setting Procedure

The ENCAn setting procedure is described below.

Table 26.33 ENCAn Setting Procedure

Initial Setting	Action	Setting status
Initial setting	Reset release	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.

26.5.1.1 Initial Setting Procedure for the Counter

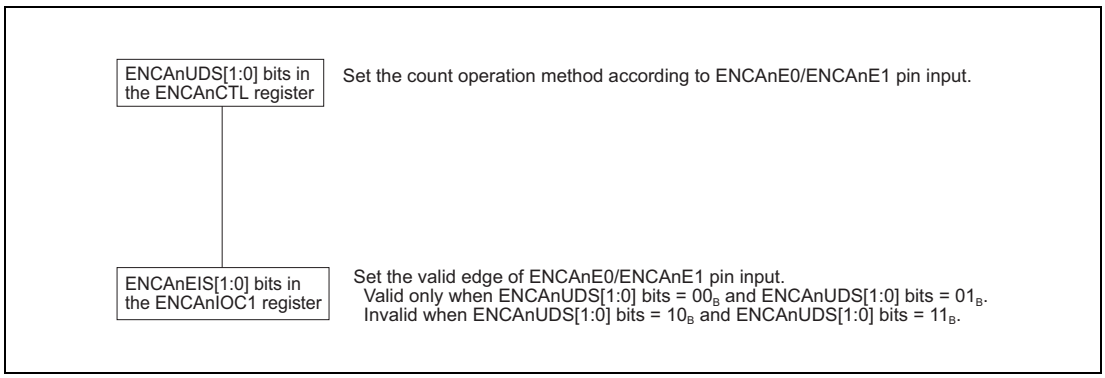


Figure 26.7 Initial Setting Procedure for the Counter

26.5.1.2 Initial Setting Procedure for Counter Clear

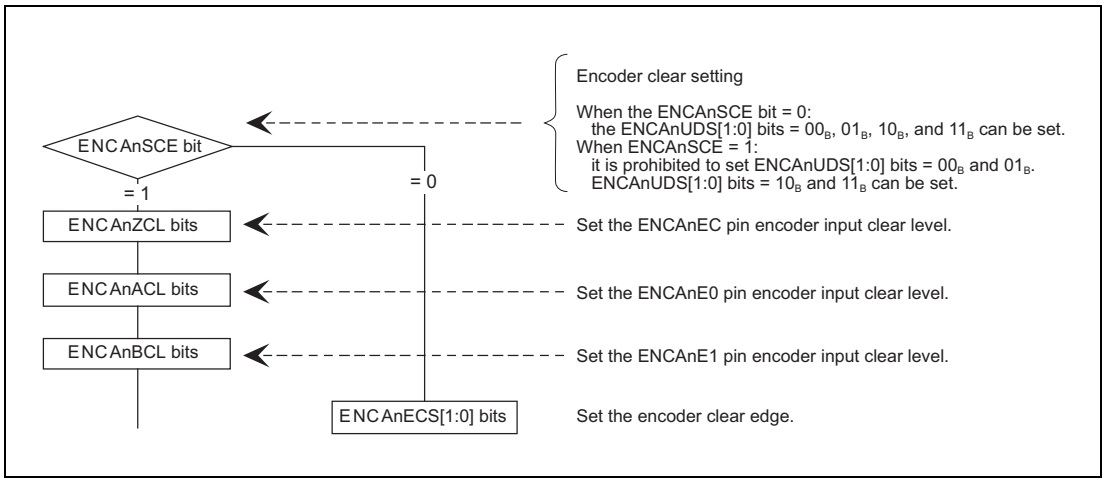


Figure 26.8 Initial Setting Procedure for Counter Clear

26.5.1.3 Setting Procedure for ENCAnCCR0 Register

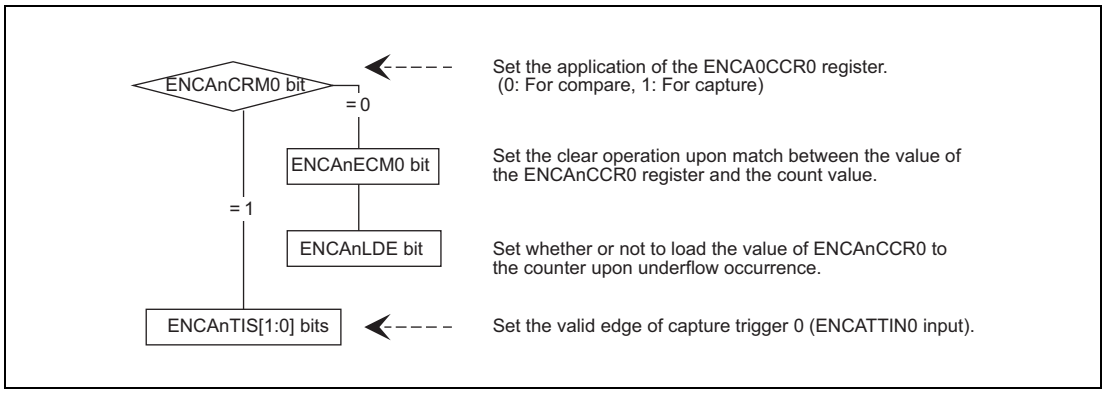


Figure 26.9 Setting Procedure for ENCAnCCR0 Register

26.5.1.4 Setting Procedure for ENCAnCCR1 Register

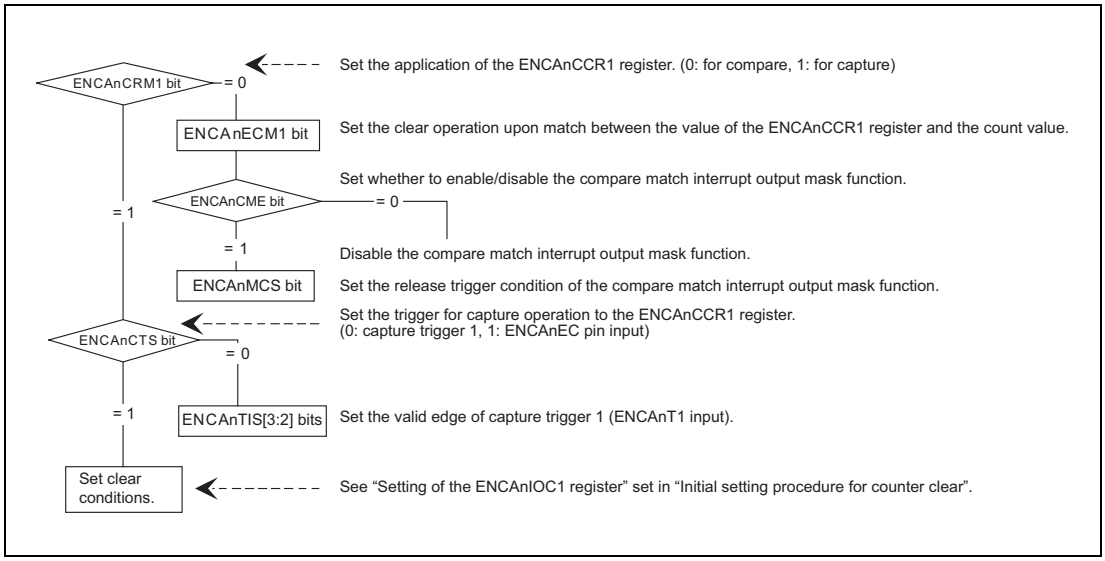


Figure 26.10 Setting Procedure for ENCAnCCR1 Register

26.6 Timing Chart

26.6.1 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.

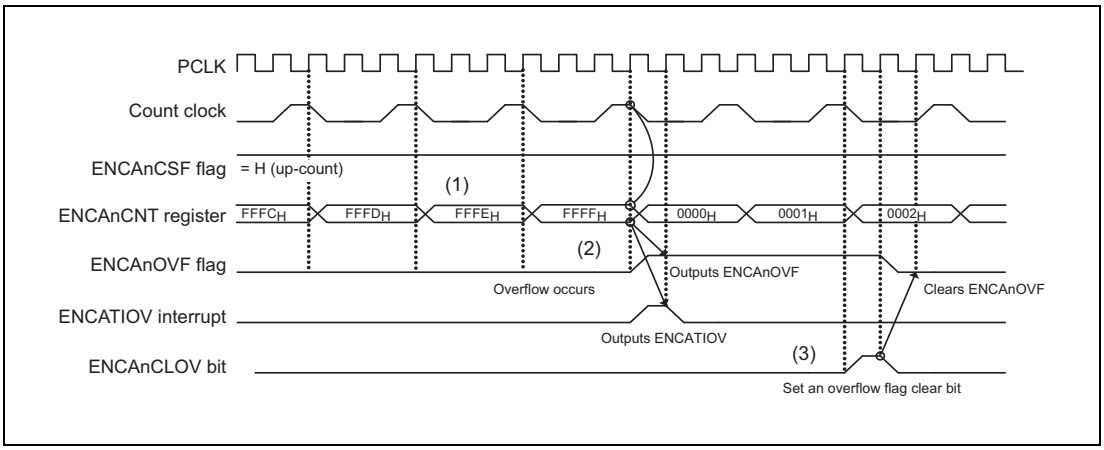


Figure 26.11 Settings of Overflow Occurrence and Overflow Flag Clear

(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”.

26.6.2 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.

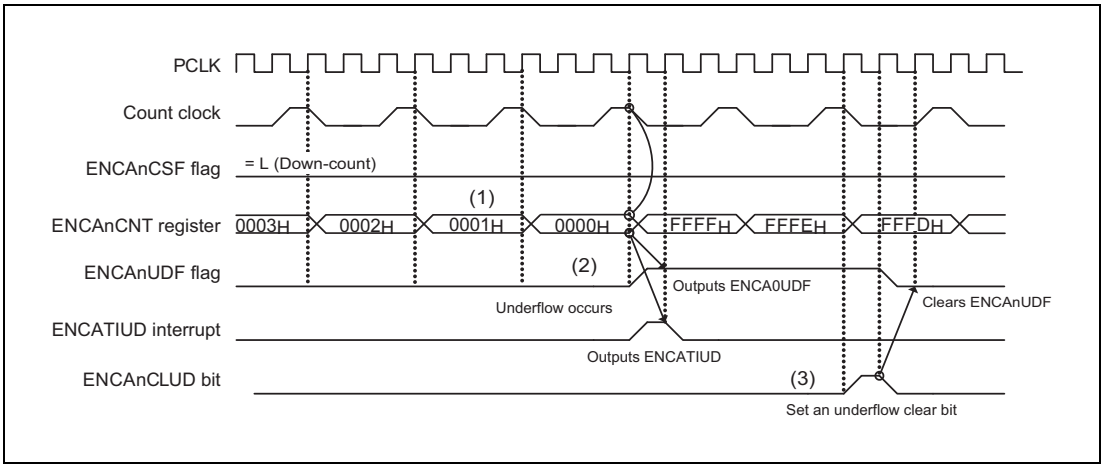


Figure 26.12 Settings of Underflow Occurrence and Underflow Flag Clear

(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”.

26.6.3 Count Clearing and Capture Operation by Encoder Clear Input (ENCAnEC Pin)

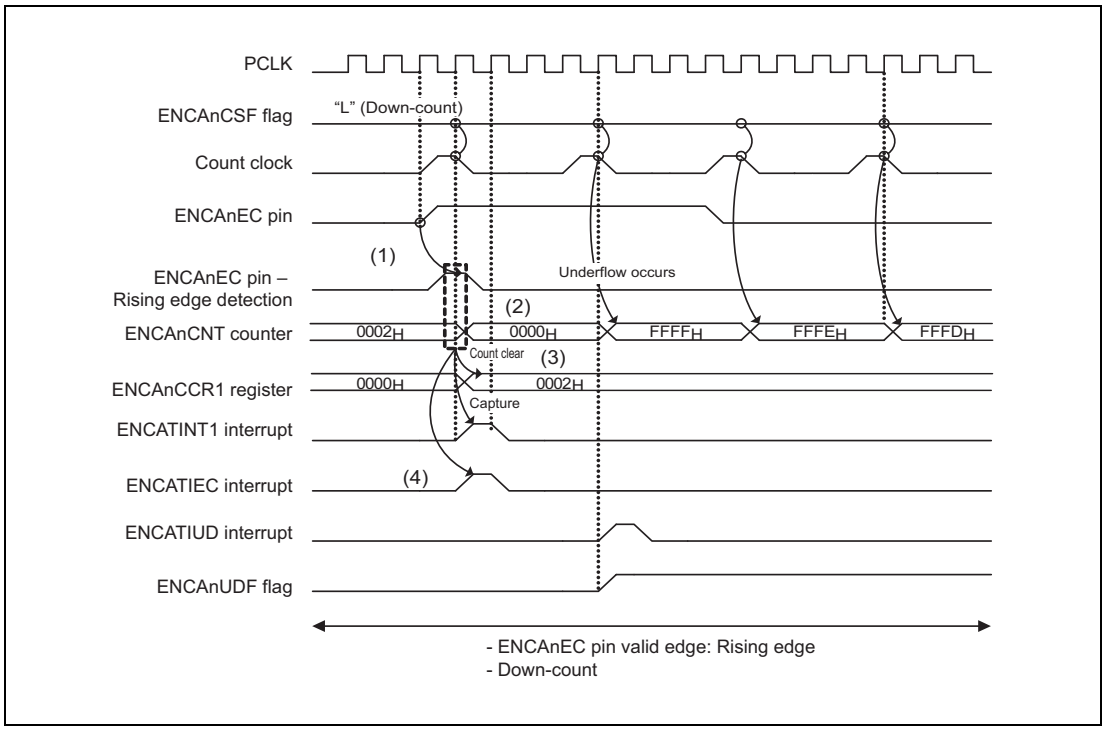


Figure 26.13 Timing Chart of Count Clearing and Capture Operation by Encoder Clear Input (ENCAnEC Pin)

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.

26.6.4 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

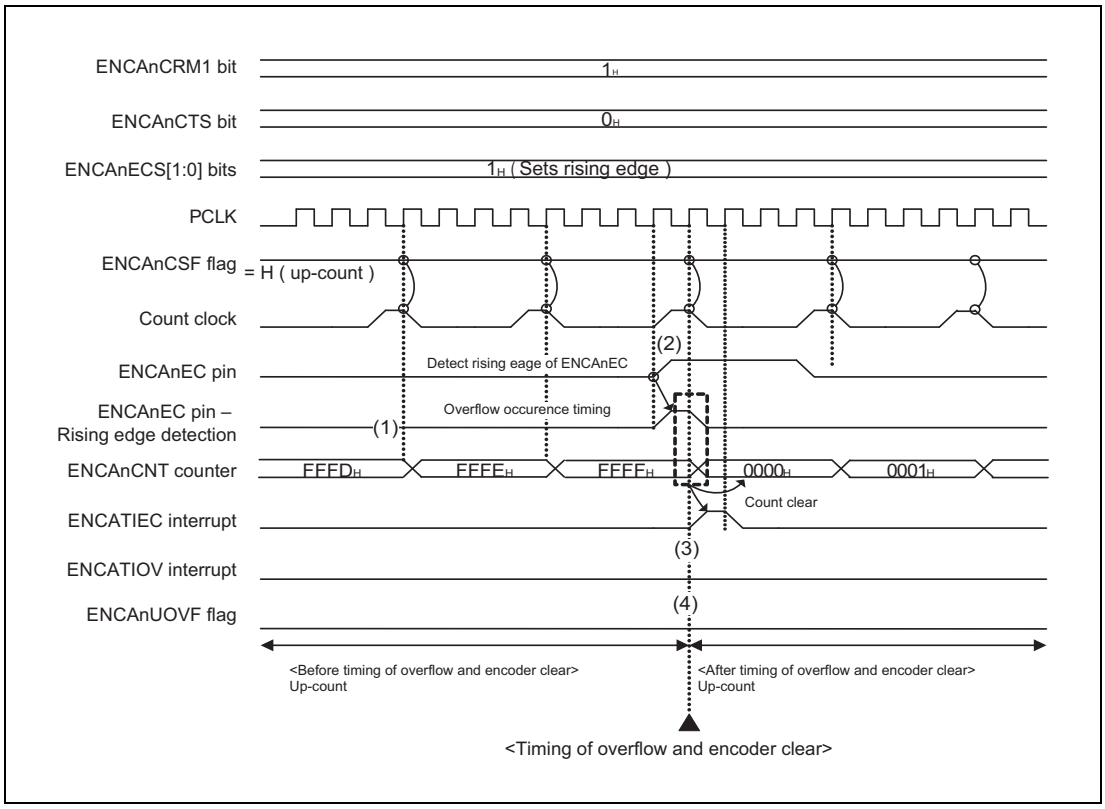


Figure 26.14 Conflict between Overflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC pin)

(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.

26.6.5 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

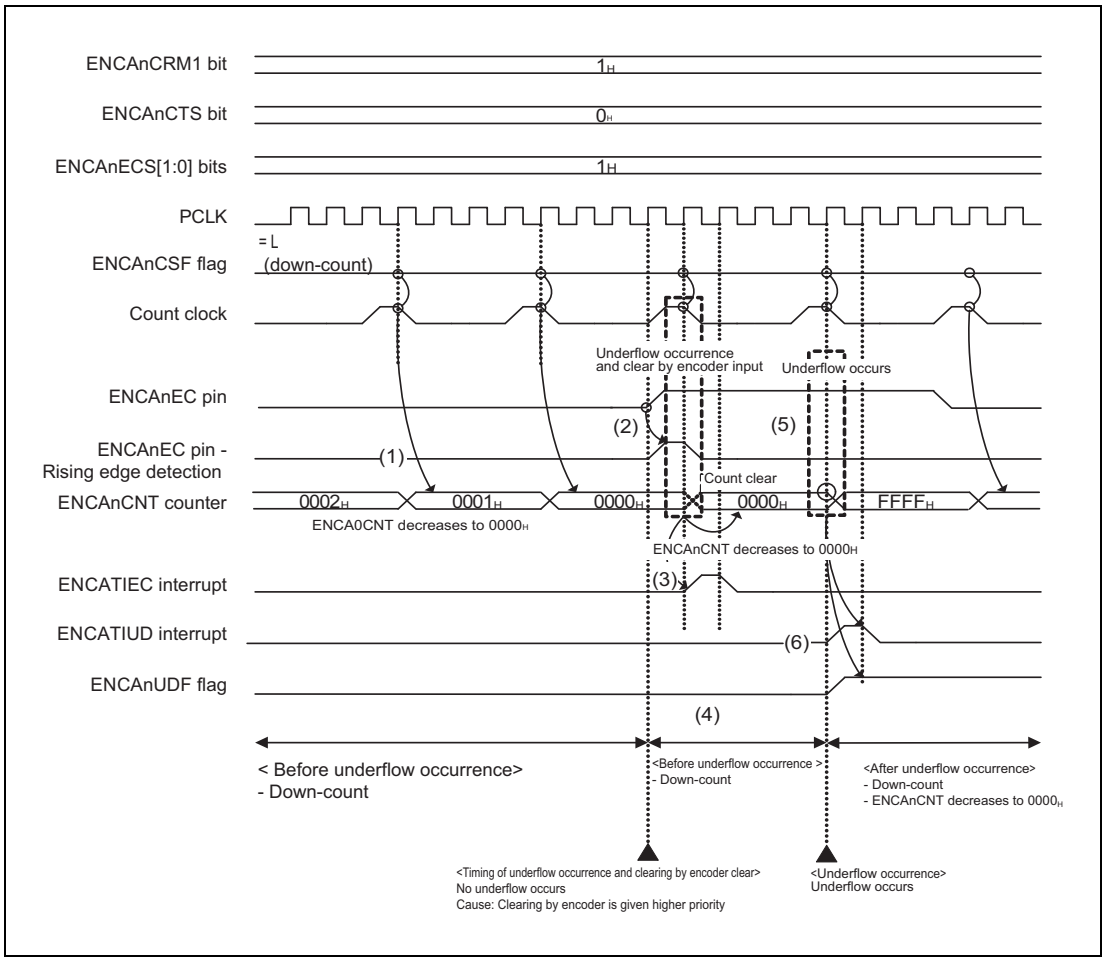


Figure 26.15 Conflict between Underflow Occurrence and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

(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.

26.6.6 Overflow Operation Immediately after Startup

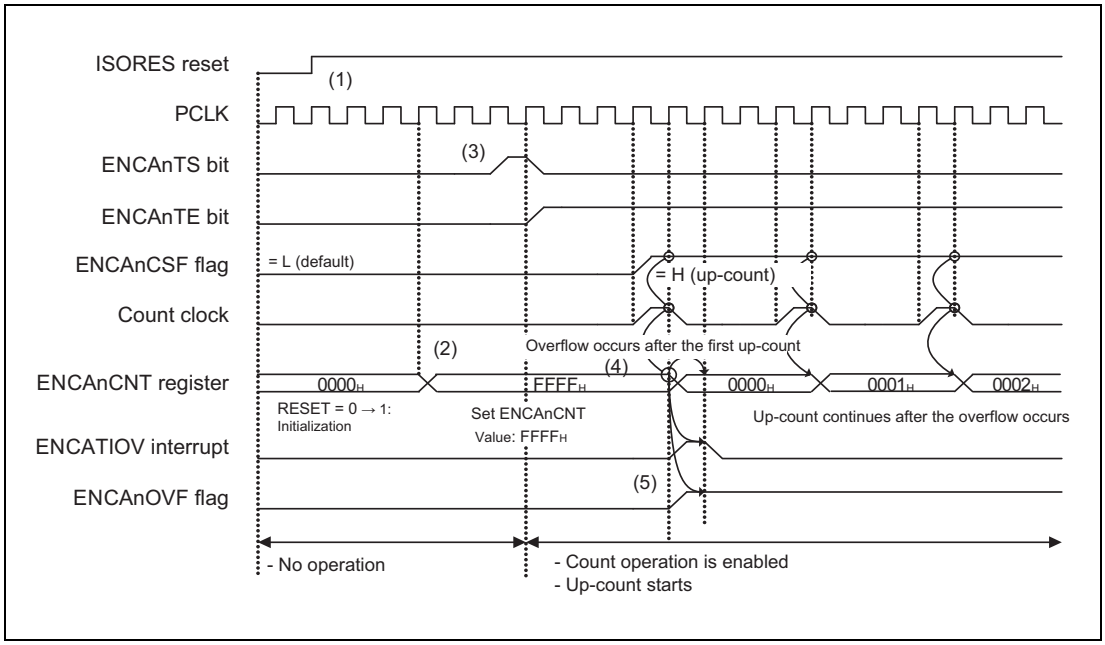


Figure 26.16 Overflow Operation Immediately after Startup

(1) When the $\overline{\text{RESET}}$ value changes from “0” to “1”, the status is changes from “reset” to “reset release”.

(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.

26.6.7 Underflow Operation Immediately after Startup

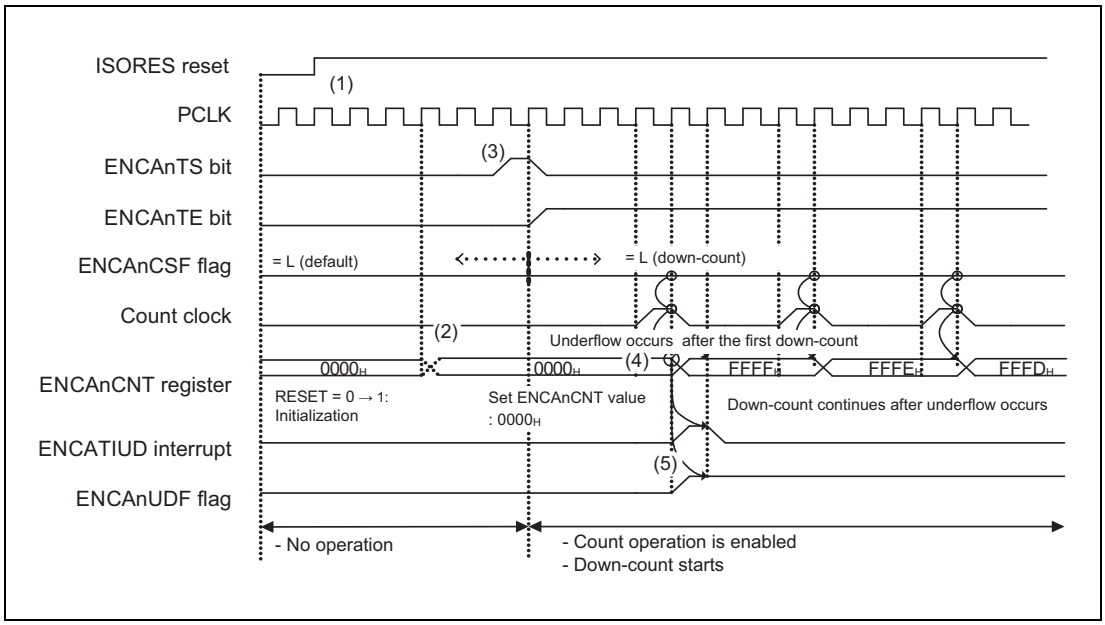


Figure 26.17 Underflow Operation Immediately after Startup

(1) When the $\overline{\text{RESET}}$ value changes from “0” to “1”, the status is changes from “reset” to “reset release”.

(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.

26.6.8 Using the ENCAnLDE Function Immediately after Startup

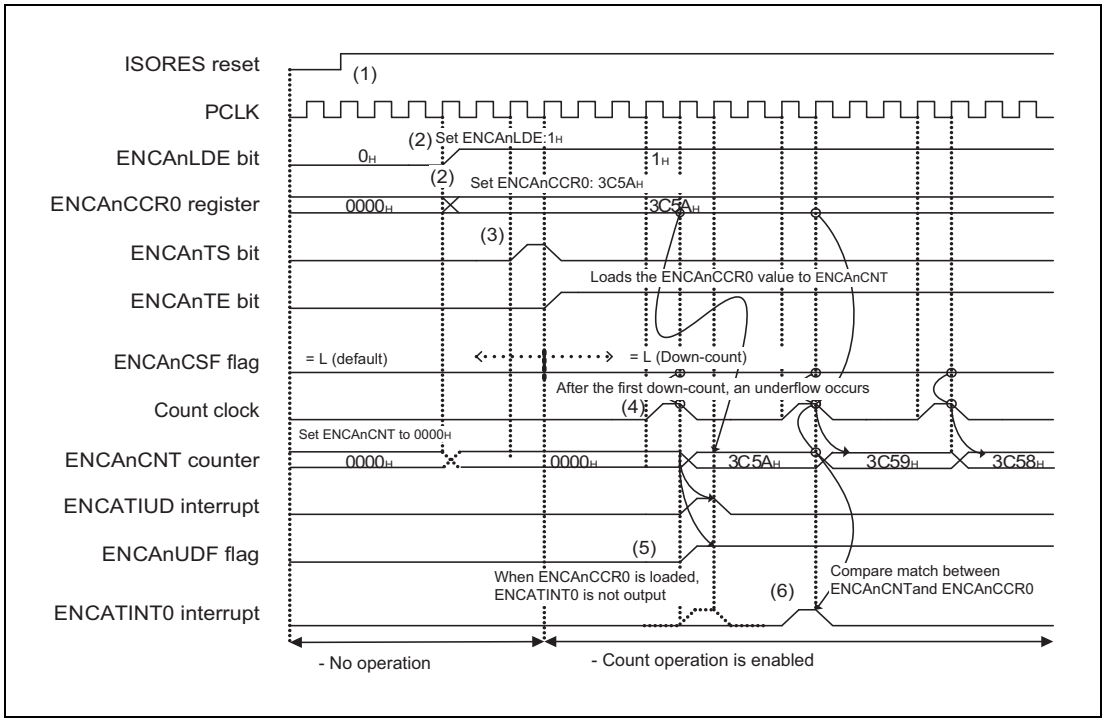


Figure 26.18 Using the ENCAnLDE Function Immediately after Startup

(1) When the $\overline{\text{RESET}}$ value changes from “0” to “1”, the status is changes from “reset” to “reset release”.

(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.

26.6.9 ENCAnLDE Function (Loading Count Value)

(1) <When ENCAnLDE = 0>

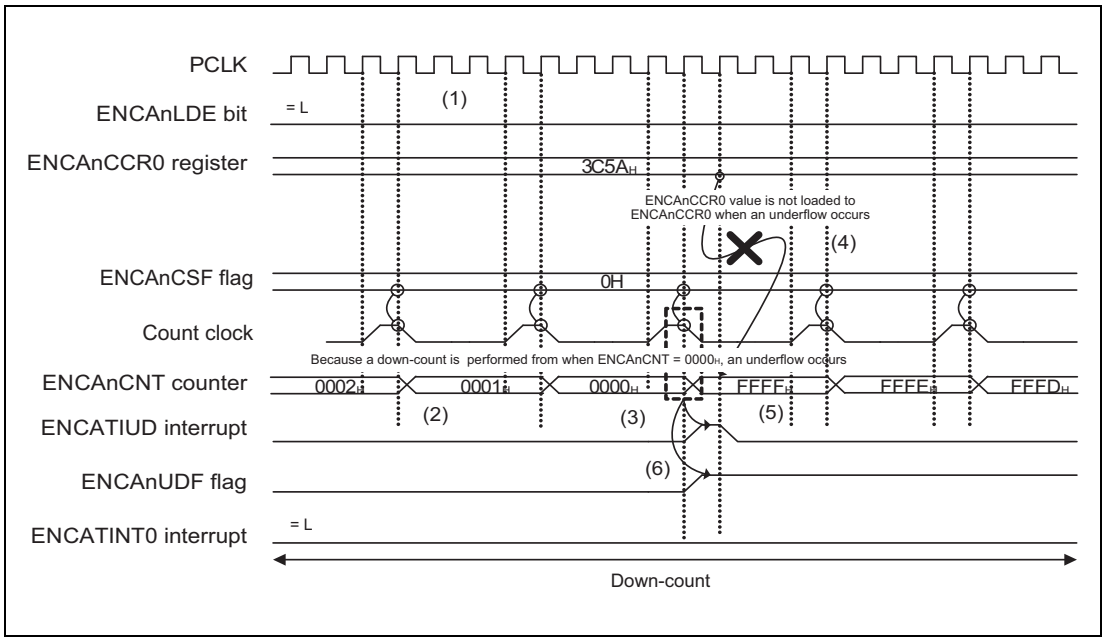


Figure 26.19 ENCAnLDE Function (when ENCAnLDE = 0)

(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>

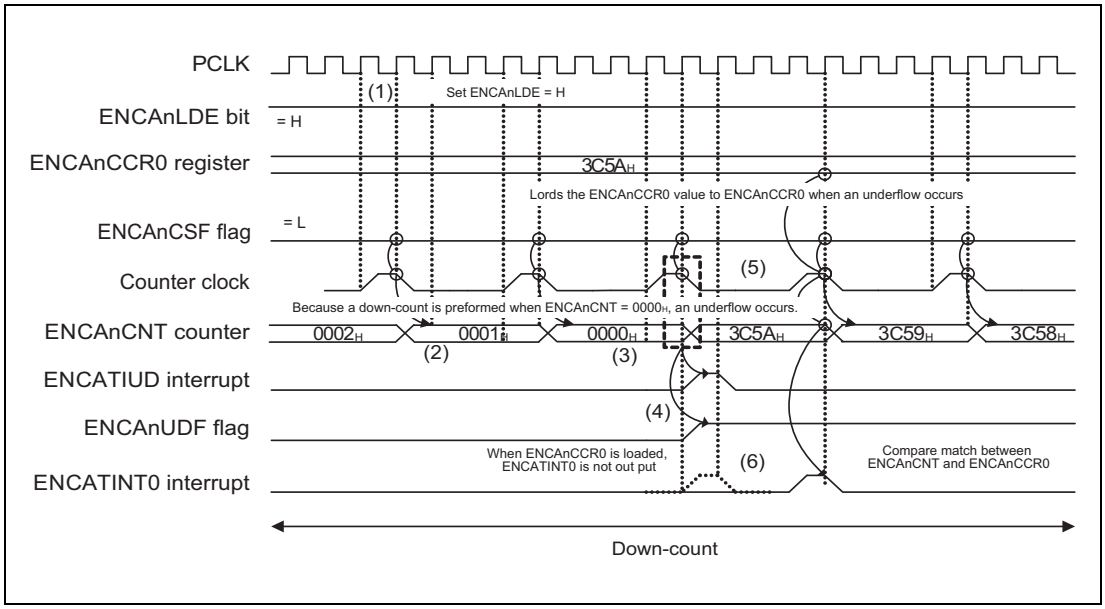


Figure 26.20 ENCAnLDE Function (when ENCAnLDE = 1)

(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.

26.6.10 Conflict between ENCAnLDE Function (Loading Counter Value) and Rewrite of ENCAnCCR0 Register

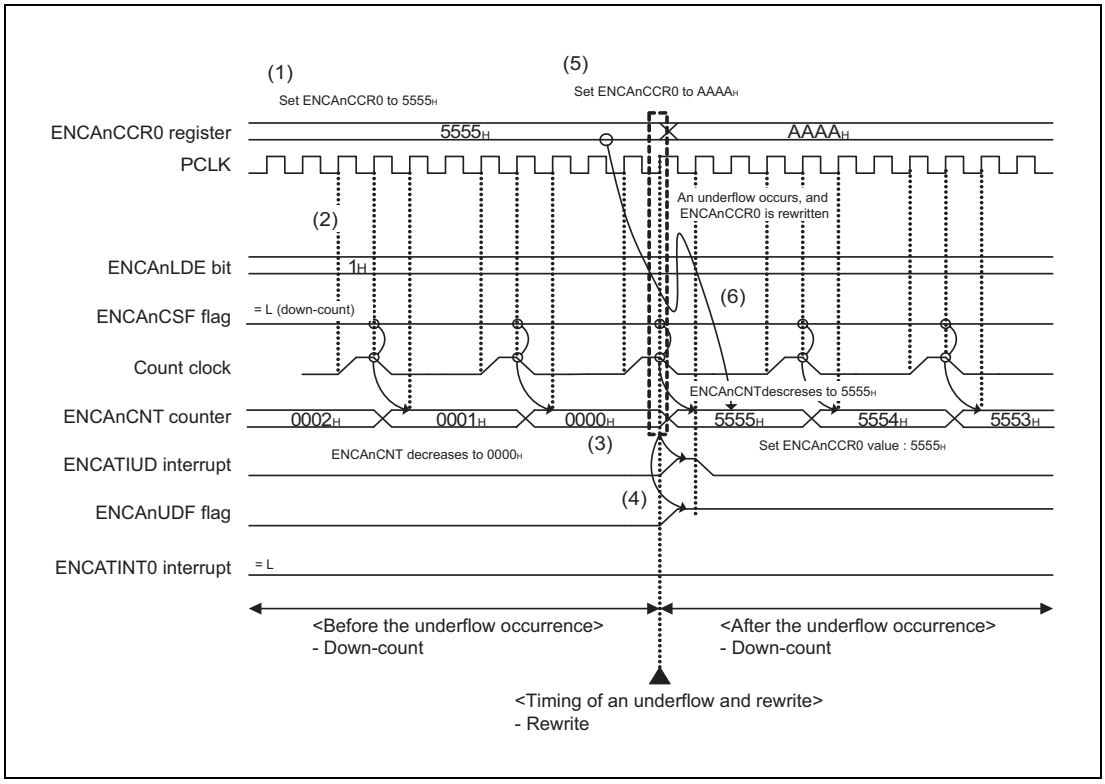


Figure 26.21 Conflict between ENCAnLDE Function and Rewrite Of ENCAnCCR0 Register

(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.

26.6.11 Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input (ENCAnEC Pin)

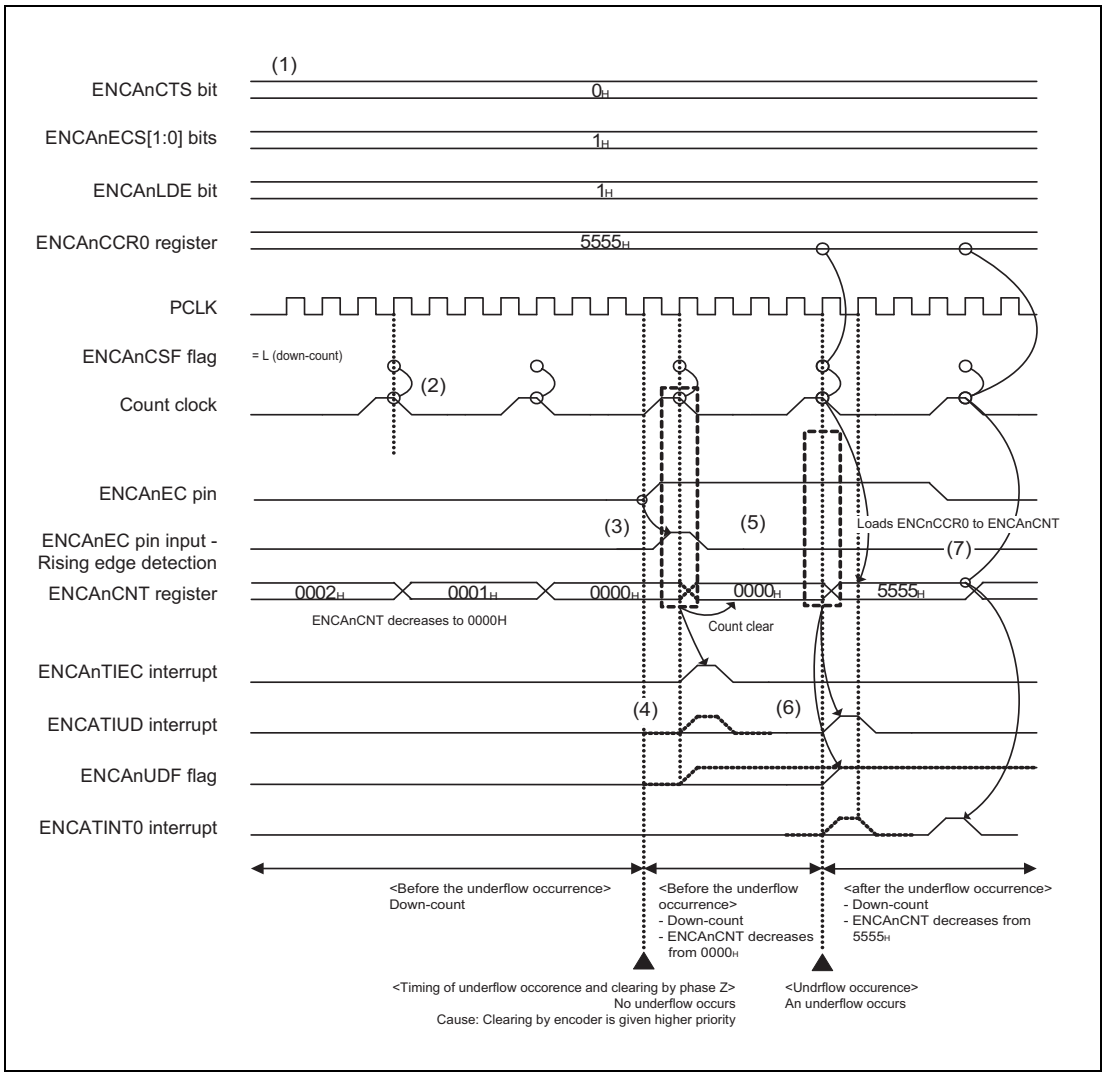


Figure 26.22 Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input

(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.

26.6.12 Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear Input

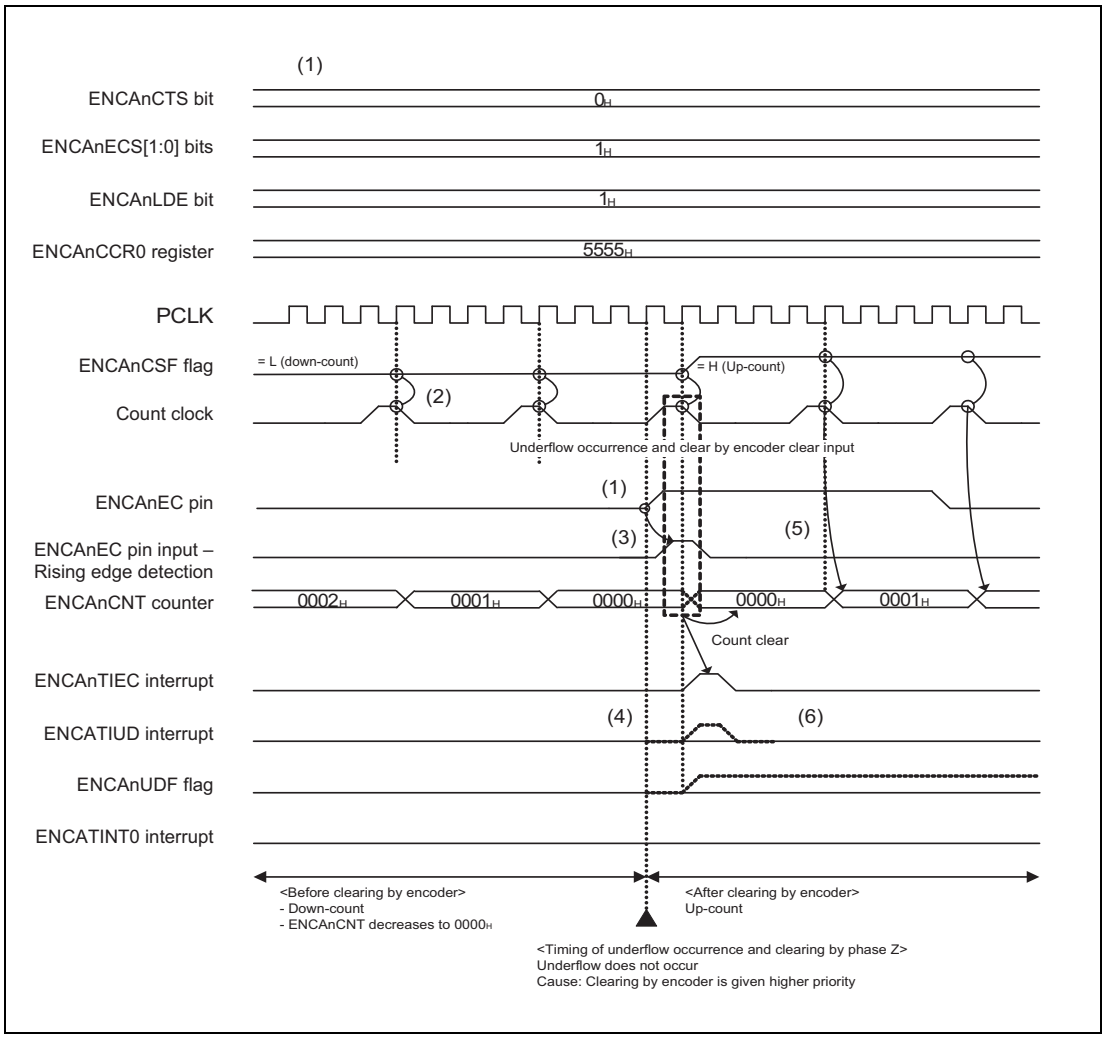


Figure 26.23 Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear

(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.

26.6.13 Capture Operation between Count Clocks (ENCAnCCR1)

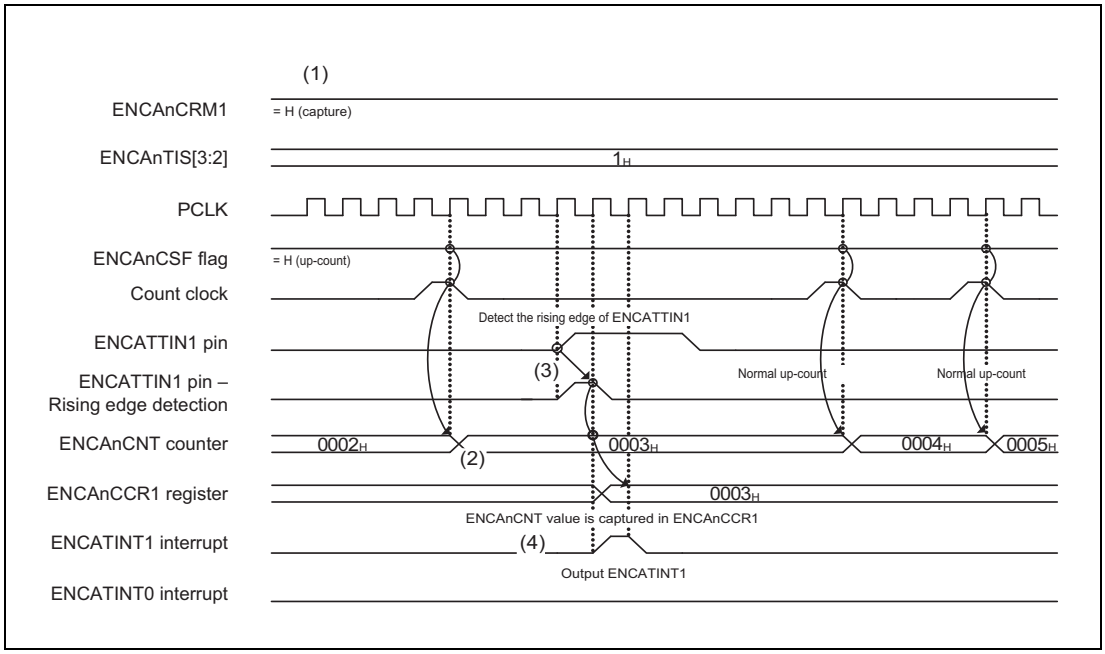


Figure 26.24 Capture Operation between Count Clocks (ENCAnCCR1)

(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.

26.6.14 Capture Operation between Count Clocks (ENCAnCCR0)

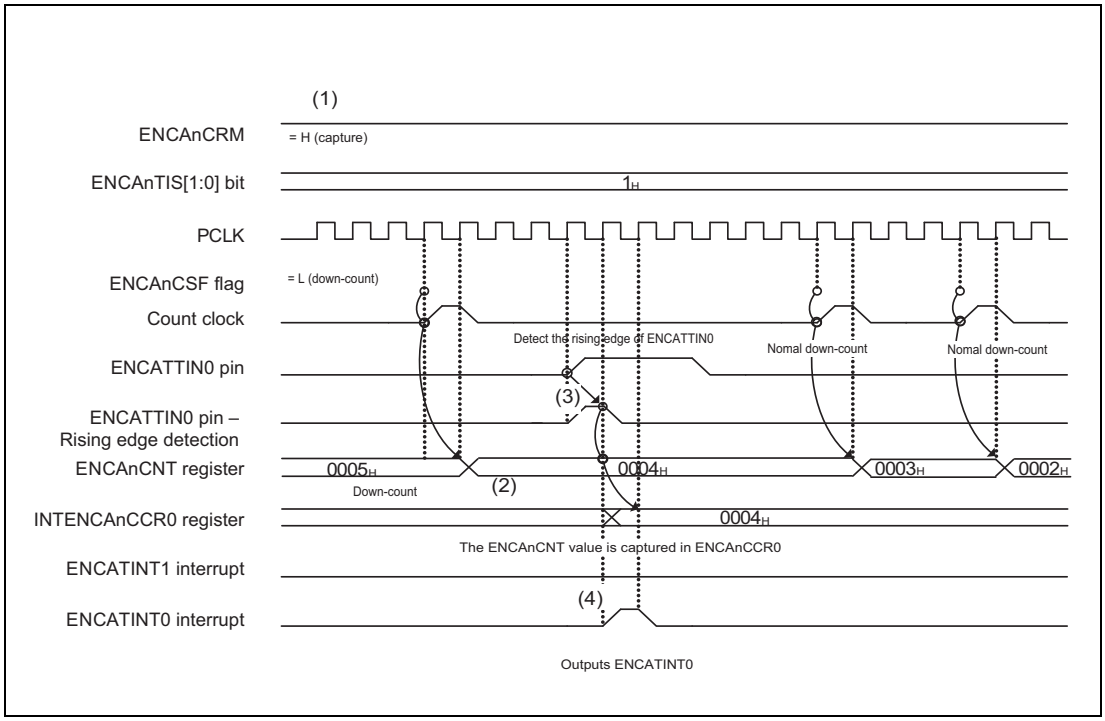


Figure 26.25 Capture Operation between Count Clocks (ENCAnCCR0)

(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.

26.6.15 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 0

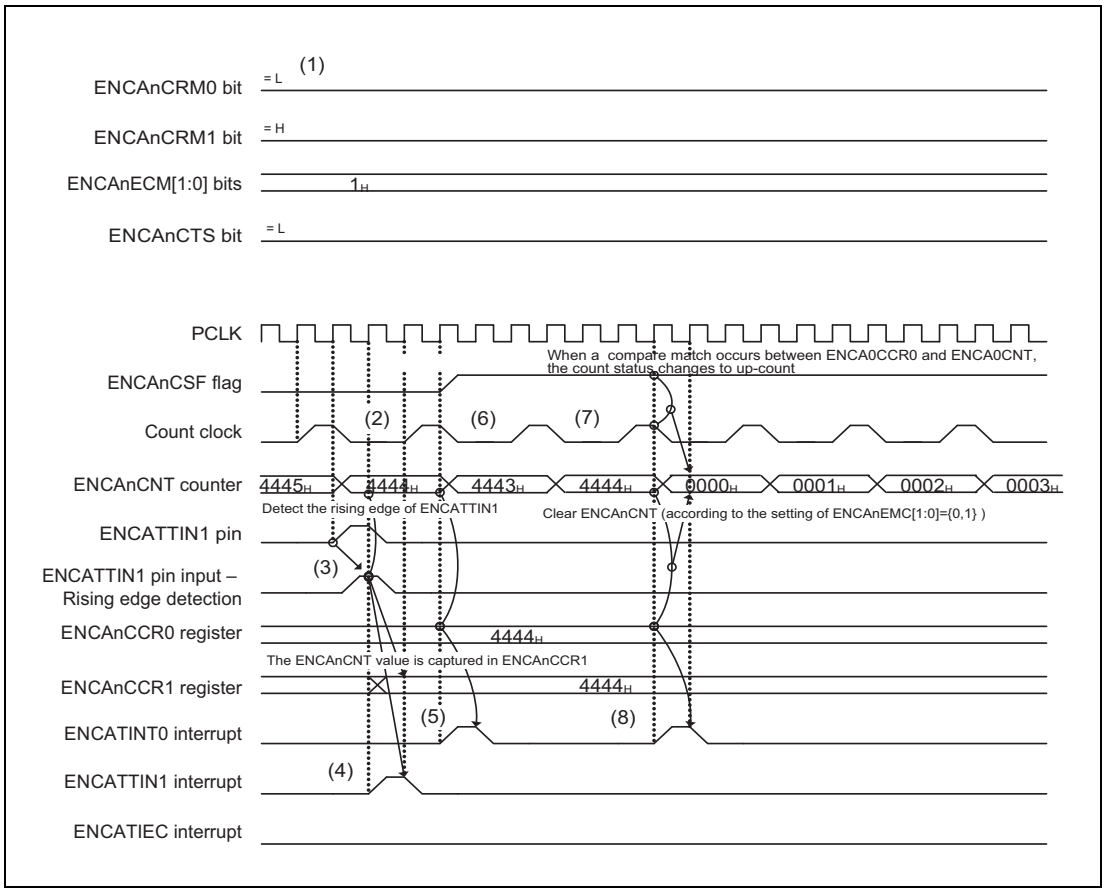


Figure 26.26 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 0

(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 (ENCATIEC) 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.

26.6.16 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 1

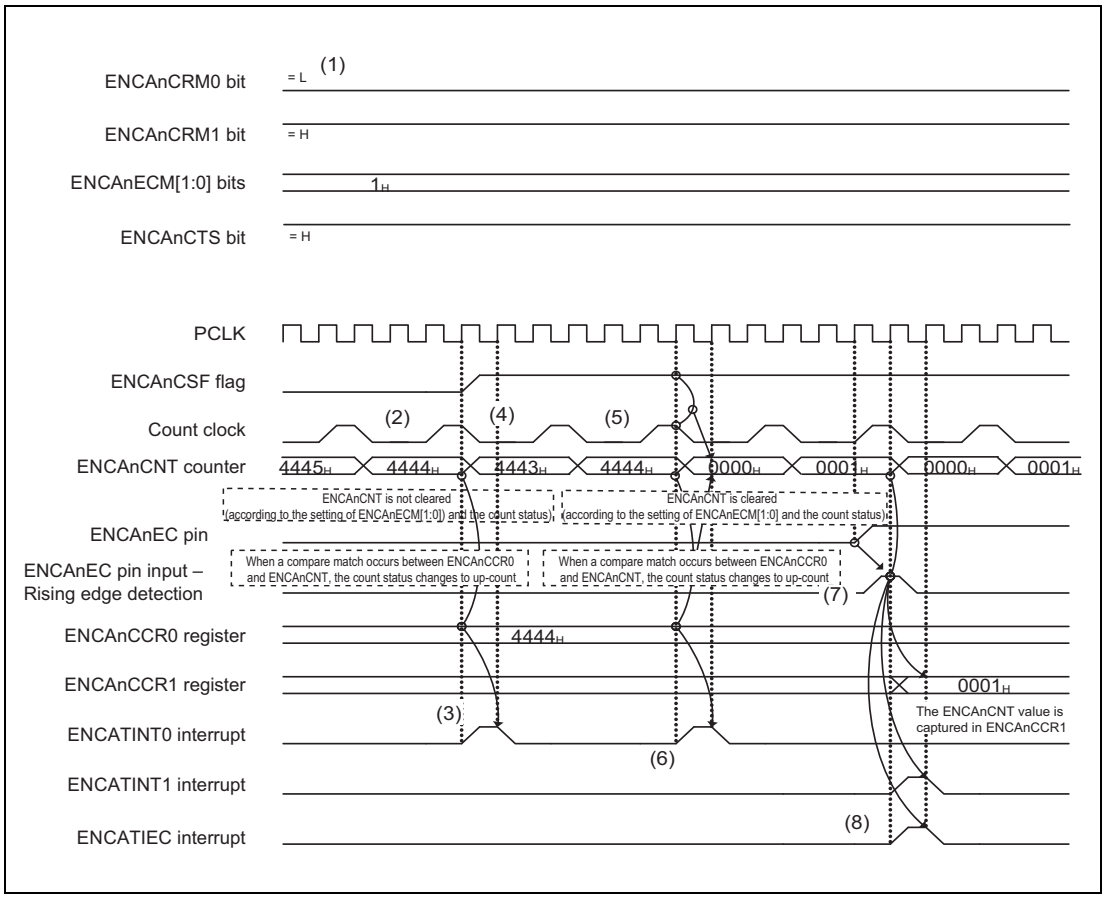


Figure 26.27 Encoder Operation when Compare Match Clear Control is Enabled and ENCAnCTS = 1

(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.

26.6.17 Encoder Operation when Compare Match Clear Control is Disabled

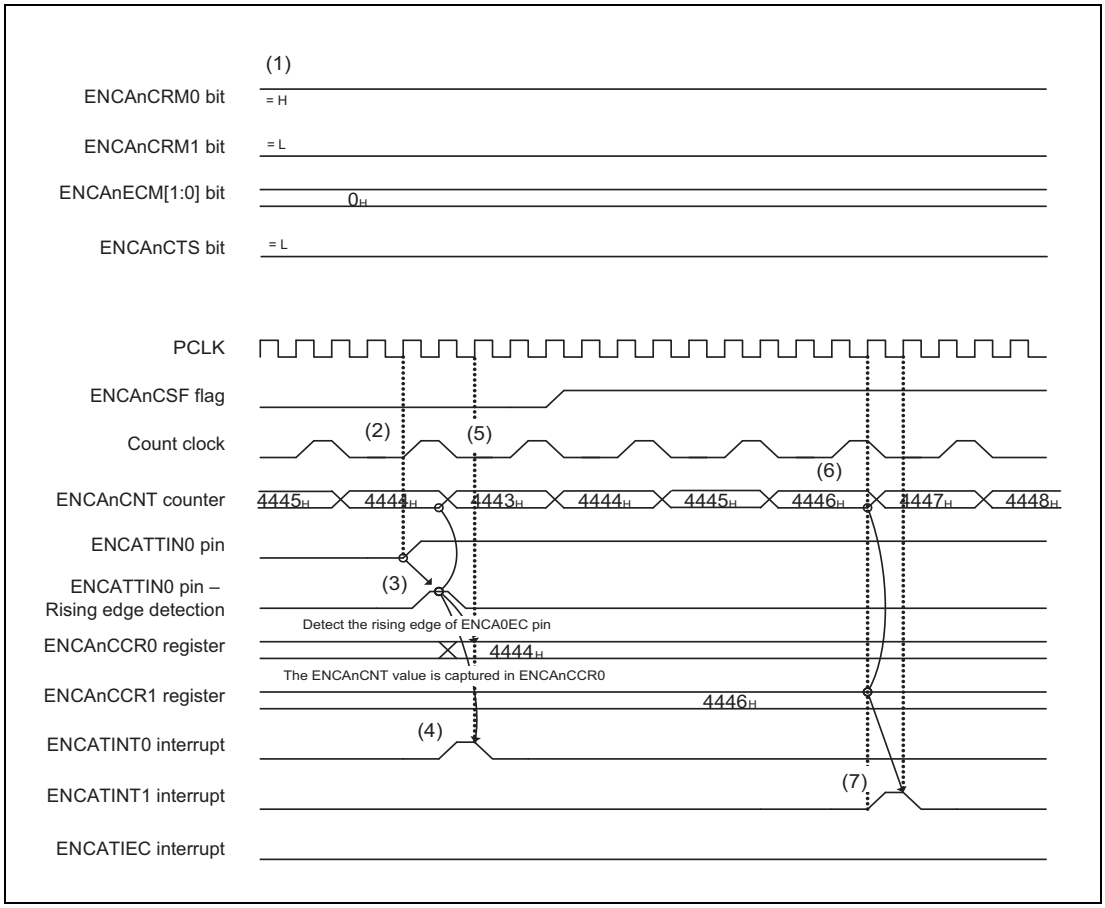


Figure 26.28 Encoder Operation when Compare Match Clear Control is Disabled

(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.

26.6.18 Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 1

26.6.18.1 Accompanying capture operation

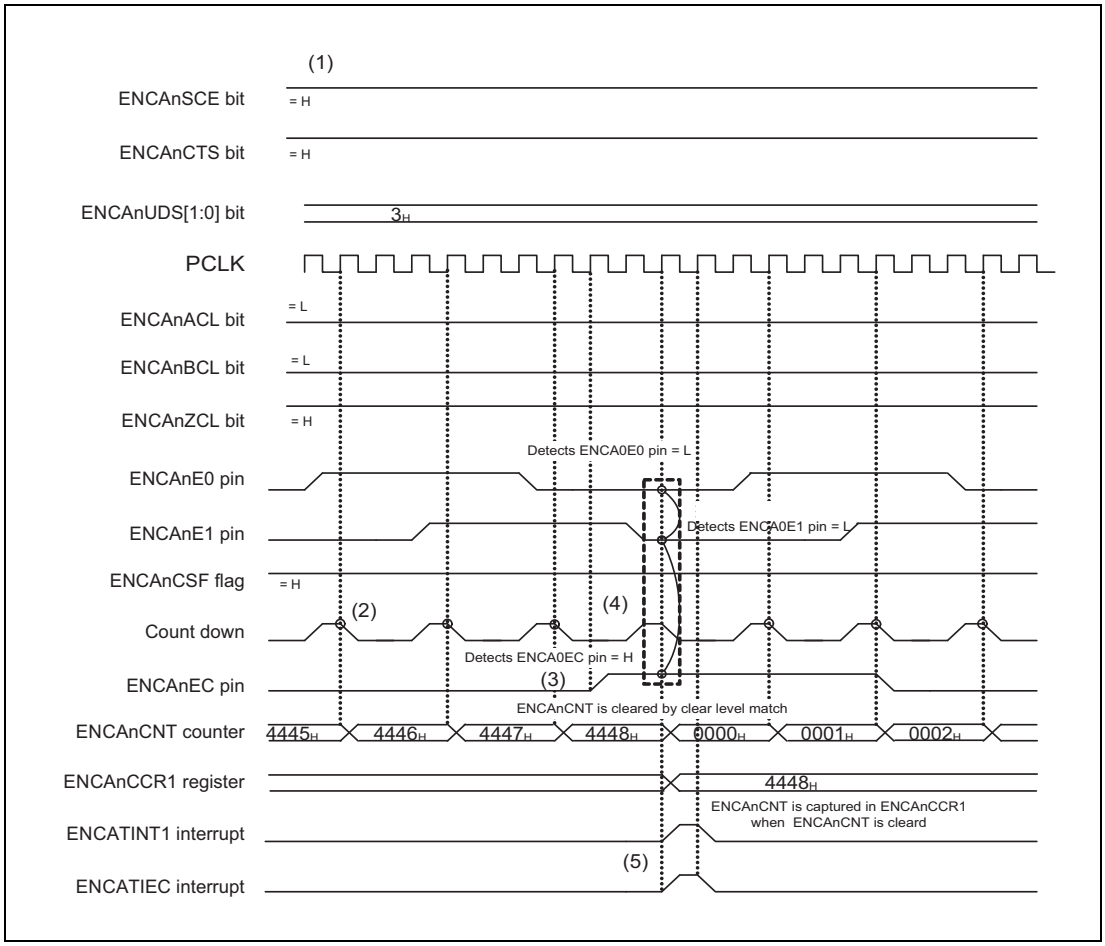


Figure 26.29 Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 1

(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.

26.6.18.2 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)

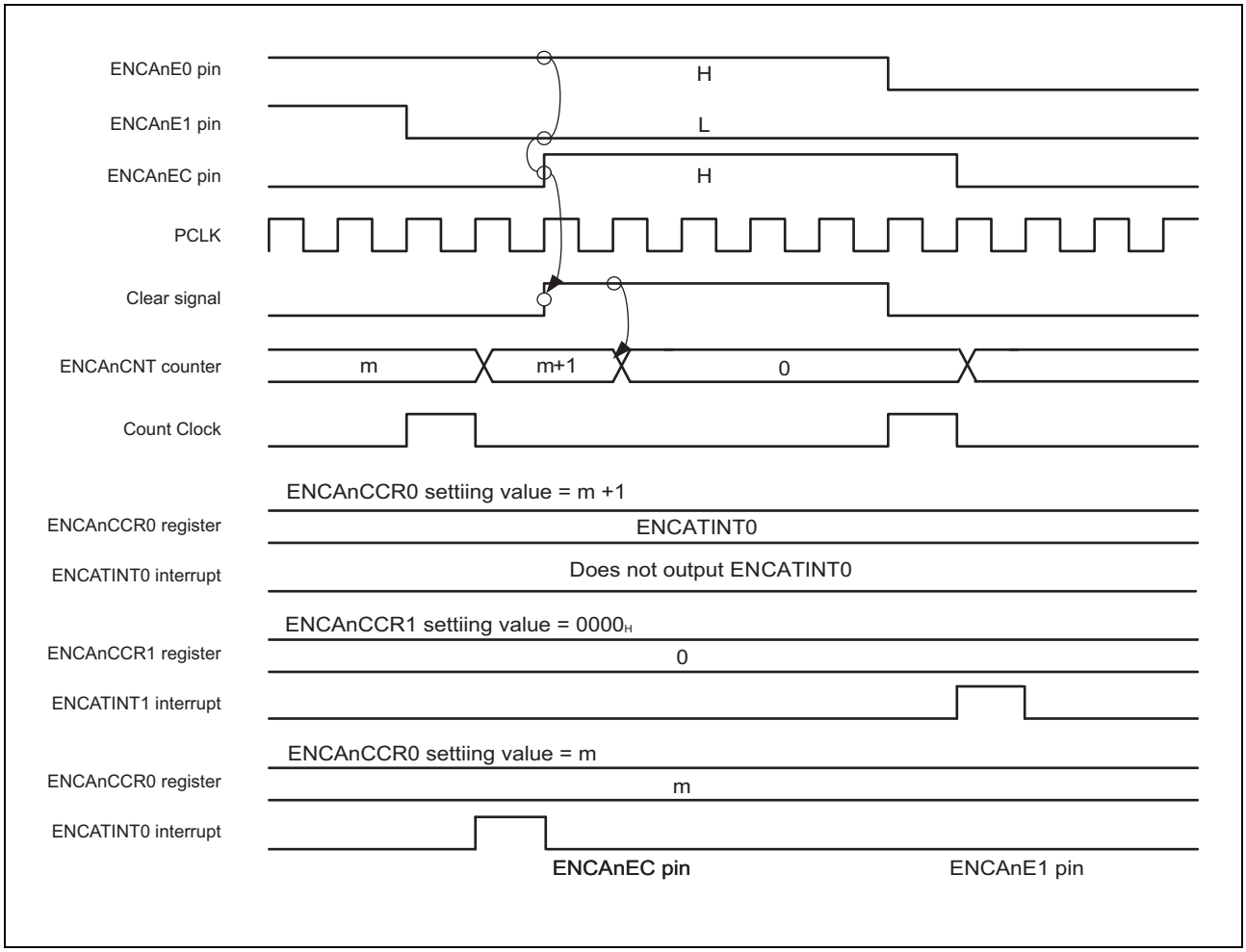


Figure 26.30 Clearing for when the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Up-count

26.6.18.3 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)

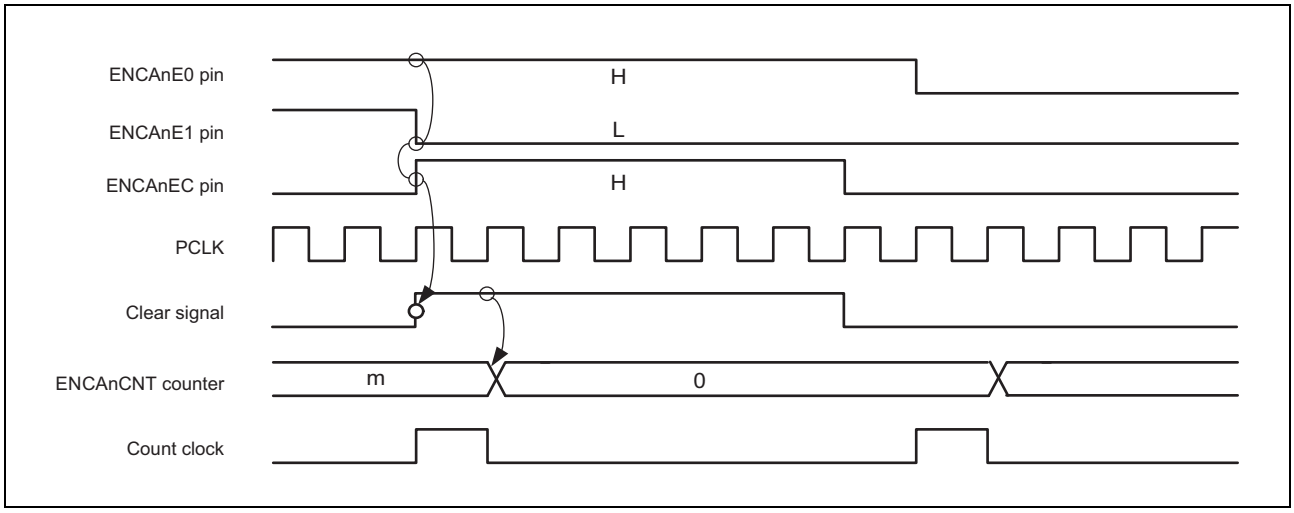


Figure 26.31 Clearing for when the Timing of the ENCAnEC Input is the Same as that of the ENCAnE1 Input during Up-count

26.6.18.4 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)

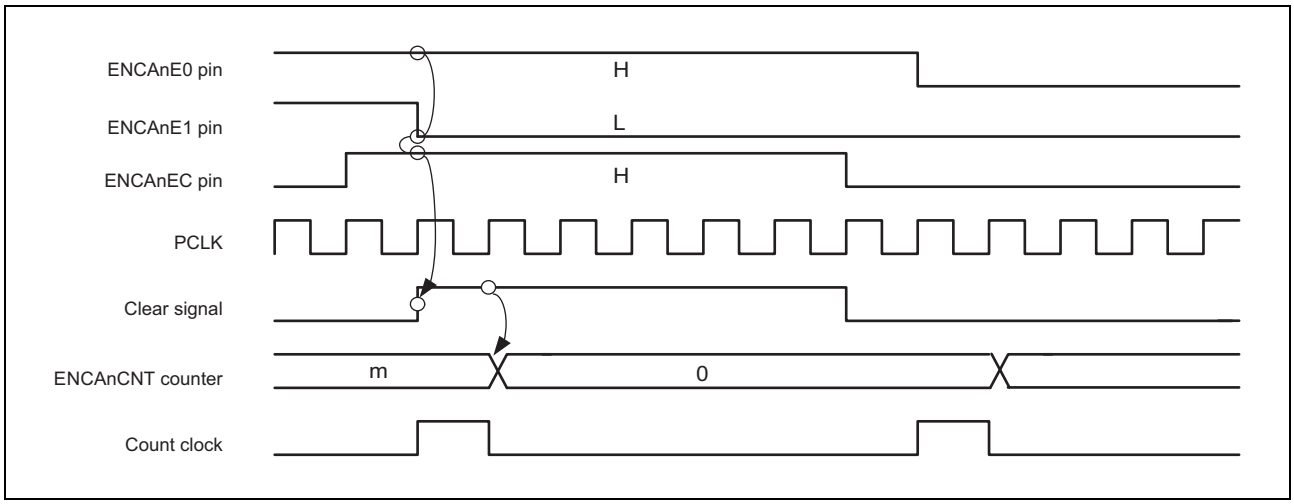


Figure 26.32 Clearing for when the Timing of the ENCAnEC Input is Earlier than that of the ENCAnE1 Input during Up-count

26.6.18.5 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)

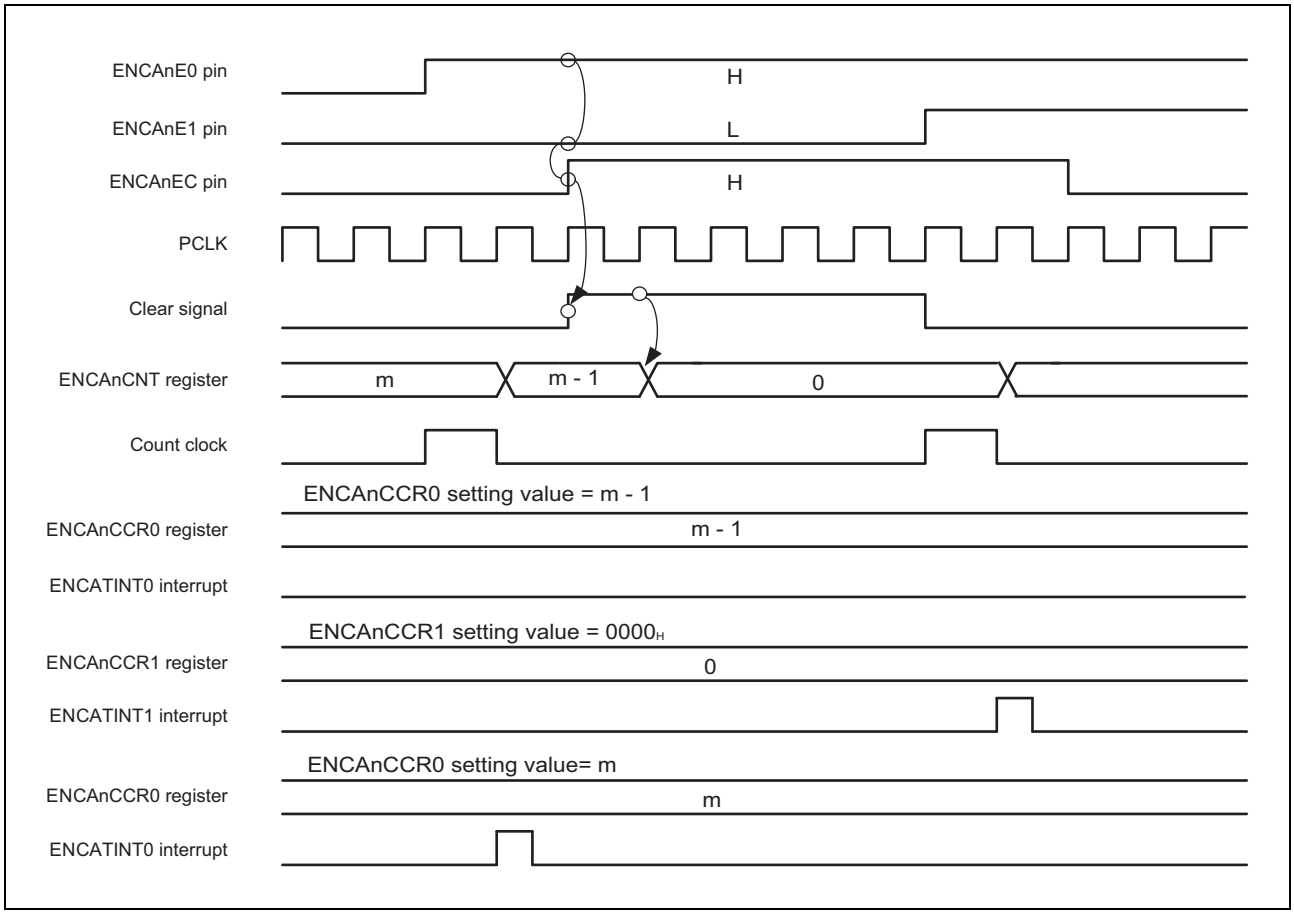


Figure 26.33 Clearing for when the Timing of the ENCAnEC Input is Later than that of the ENCAnE1 Input during Down-count

26.6.19 Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0

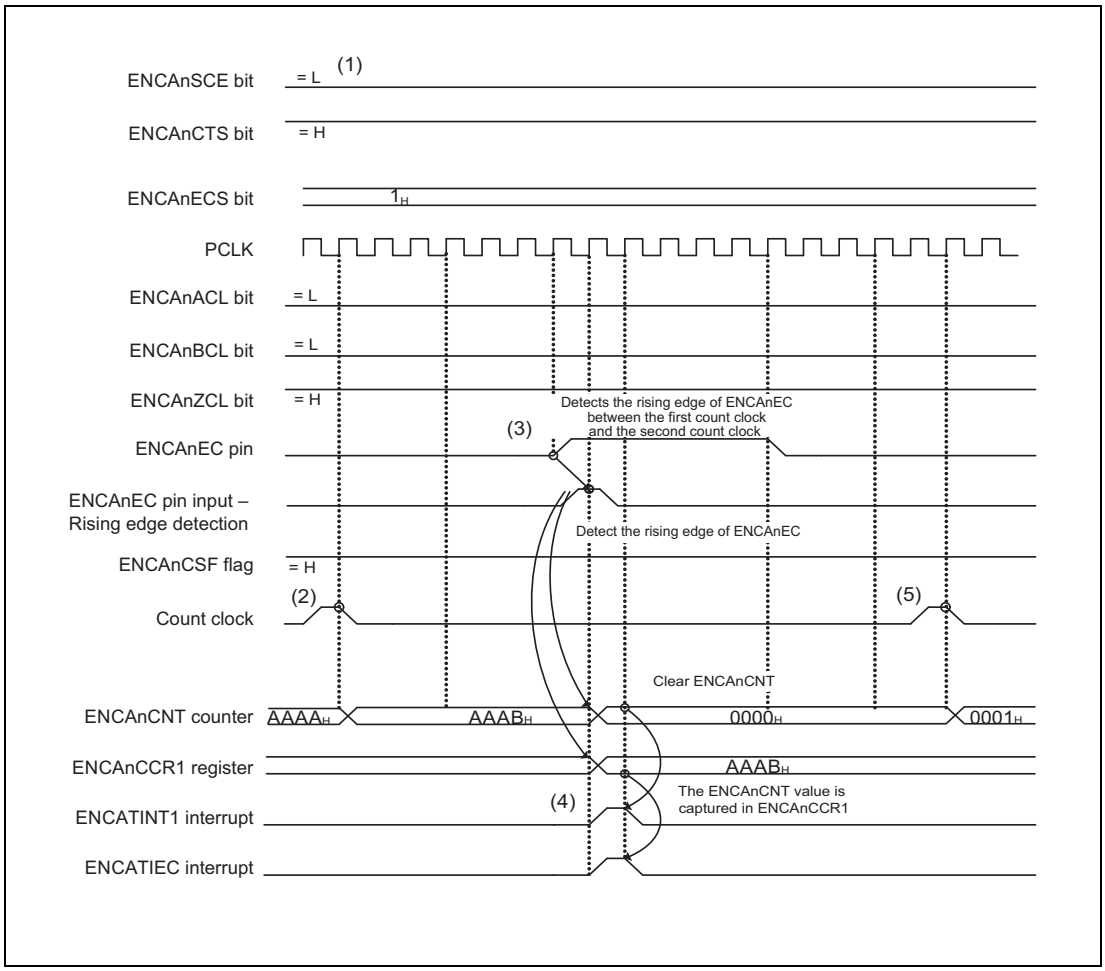


Figure 26.34 Capture Operation Performed upon Clearing by ENCAnEC when ENCAnSCE = 0

(1) The values are set as follows: ENCAnSCE = 0, ENCAnCTS = 1, and ENCAnECS1 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 27 Motor Control

This section contains a generic description of the Motor Control.

The first part in this section describes all RH850/F1L 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.

27.1 Features of RH850/F1L Motor Control

27.1.1 Number of Units and Channels

Motor control function comprises the timer motor control units (TAPA) and the peripheral interconnection (PIC) to generate motor control waves by combining peripheral timers and A/D converters.

This microcontroller has the following number of units of the TAPA and PIC.

Table 27.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
TAPA						
Number of Units	1					
Name	TAPAn (n = 0)					
PIC						
Number of Units	1					
Name	PIC0					

Table 27.2 Index

Index	Meaning
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 shows values indicated by the indexes of each product.

Table 27.3 Indexes of Products

Indexes of Each Product
All Products
m = 0 to 15 (e.g. TAUDn)
x = 1 to 3
j = 0 to 2

27.1.2 Register Base Address

Base addresses of TAPAn and PIC0 are listed in the following table.

Register addresses of TAPAn and PIC0 are given as offsets from the base addresses in general.

Table 27.4 Register Base Address

Base Address Name	Base Address
<TAPA0_base>	FFE9 0000$_H$
<PIC0_base>	FFDD 0000$_H$

27.1.3 Clock Supply

The TAPAn and PIC0 clock supply are shown in the following table.

Table 27.5 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
TAPAn	PCLK	CKSCLK_IPERI1
PIC0	PCLK	CKSCLK_IPERI1

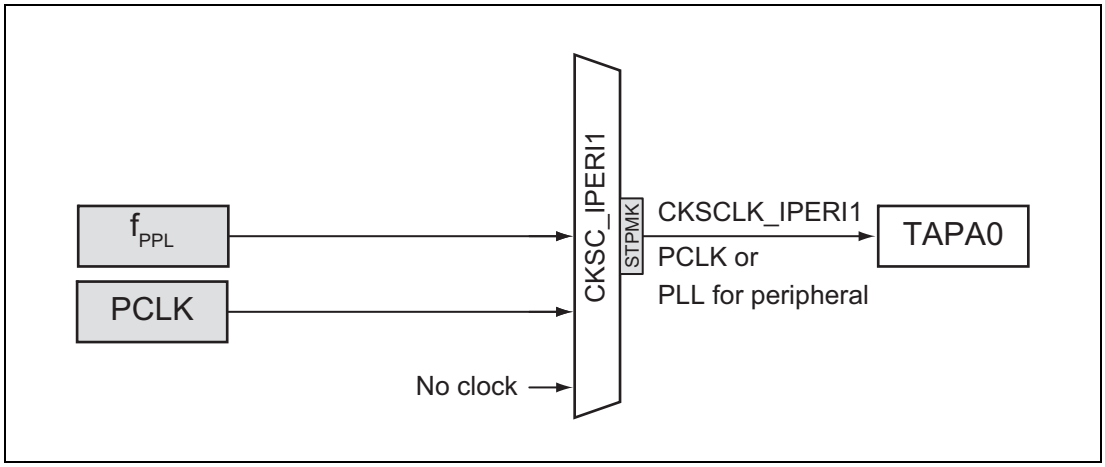


Figure 27.1 TAPA Clock Supply

27.1.4 Interrupt Request

TAPA0 interrupt requests are listed in the following table.

Table 27.6 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
TAPA0			
TAPATIPEK0	TAPA0 peak interrupt 0	8, 108	—
TAPATIVLY0	TAPA0 valley interrupt 0	9, 109	—

27.1.5 Reset Sources

Reset sources of TAPAn and PIC0 are listed in the following table. TAPAn and PIC0 are initialized by these reset sources.

Table 27.7 Reset Sources

Unit Name	Reset Source
TAPA0	All reset sources (ISORES)
PIC0	All reset sources (ISORES)

27.1.6 External Input/Output Signal

External output signals of TAPAn and PIC0 are listed below.

Table 27.8 External Input/Output Signals

Unit Signal Name	Outline	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

27.1.7 Internal Output Signal

Internal output signals of TAPAn and PIC0 are listed below.

Table 27.9 Internal Output Signals

Unit Signal Name	Outline	Connected to
TAPA0		
$\overline{\text{TAPATHZOUT0}}$	TAPA0UP/TAPA0UN output buffers Hi-Z control output*1	Port
$\overline{\text{TAPATHZOUT1}}$	TAPA0VP/TAPA0VN output buffers Hi-Z control output*1	Port
$\overline{\text{TAPATHZOUT2}}$	TAPA0WP/TAPA0WN output buffers Hi-Z control output*1	Port
TAPATADOUT0	A/D trigger signal 0 output*2	ADCA0 hardware trigger expansion
TAPATADOUT1	A/D 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 1match detect*4 (ADCA0 hardware trigger expansion: ADOPA2ADCATTIN00)	TAPA0

Note 1. See **Section 27.4.6, TAPA0 Hi-Z Control Input Selection** for details.

Note 2. These signals are selected by the A/D Converter ADCA0 H/W trigger selection. See **Table 29.47, 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 can be used to as a trigger source to start the A/D Converter. See **Table 29.47, List of A/D Conversion Hardware Triggers**.

27.2 Overview

27.2.1 Functional Overview

The motor control function can provide 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 the INTn signals output by TAUDn.

Additionally, the motor control function can also provide 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 ECNAn 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 with an 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.

27.2.2 Basic Structure of Motor Control

The peripheral block configuration of a motor control function is shown below.

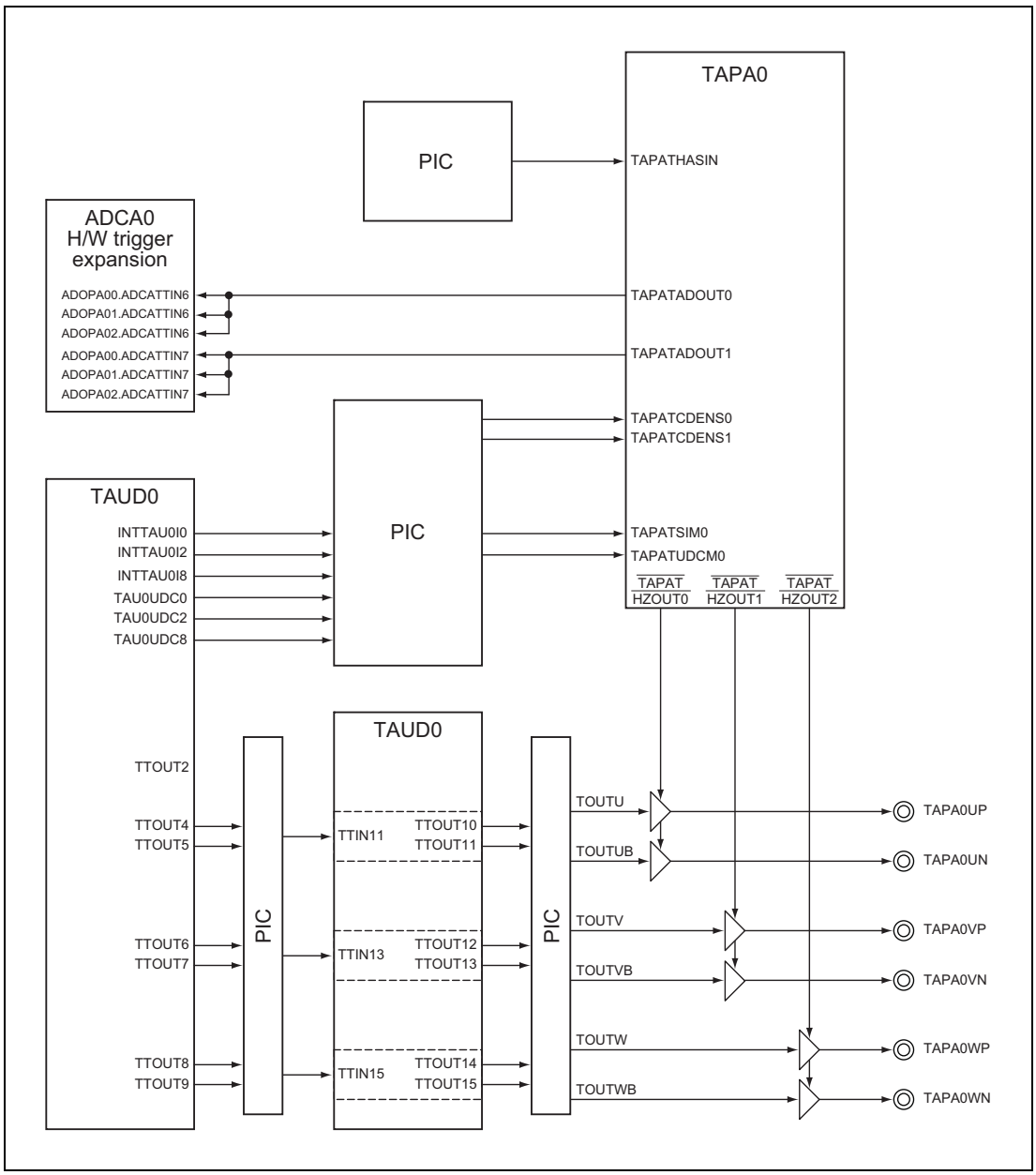


Figure 27.2 Configuration of Motor Control

For the generation of motor control output signals (three-phase PWM output signals with dead time), TAUDn and PIC are used.

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.

27.2.3 Block Diagram

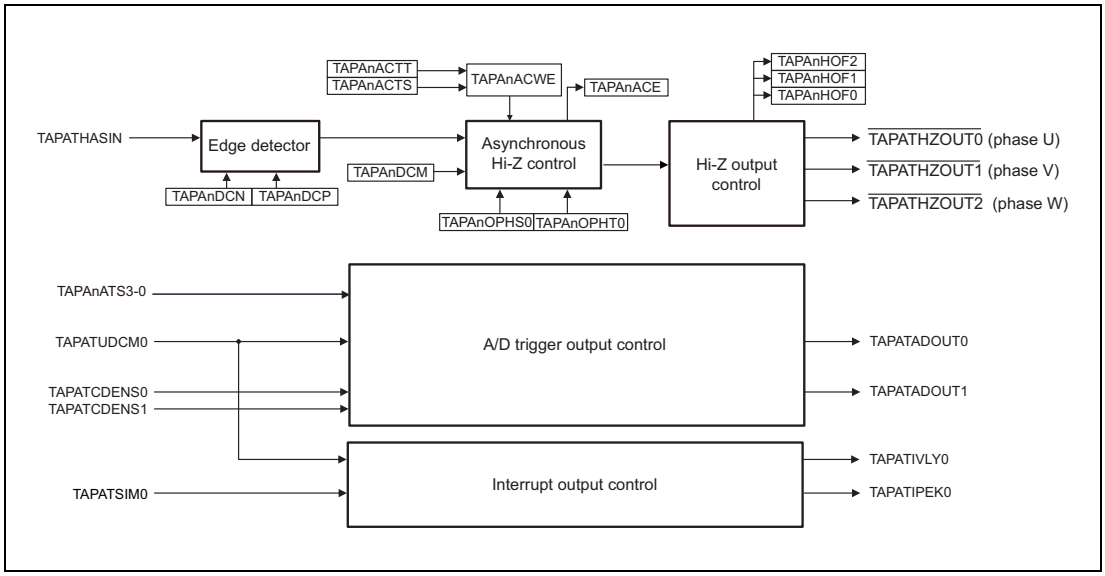


Figure 27.3 TAPA Peripheral Block Diagram

NOTE

For the PIC peripheral block diagram, see the section of each function explanation.

27.2.4 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.

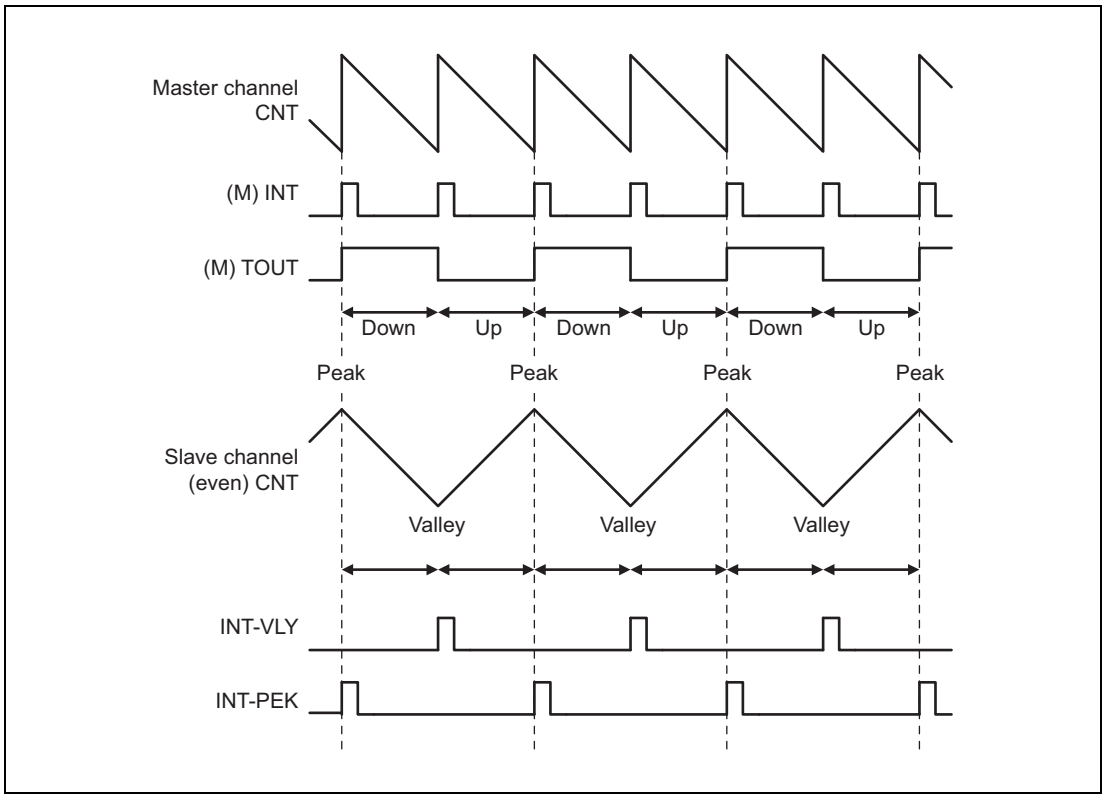


Figure 27.4 Peak and Valley Interrupts

27.3 Registers

27.3.1 List of Registers

Registers of TAPAn and PIC0 are listed in the following table.

For details about <TAPAn_base> and <PIC0_base>, see **Section 27.1.2, Register Base Address**.

Table 27.10 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

27.3.2 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 27.11 TAPAnCTL0 Register Contents

Bit Position	Bit Name	Function
15 to 5	Reserved	When read, the value after reset is returned. When written, 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 written, write the value after reset.

27.3.3 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 27.12 TAPAnCTL1 Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When written, 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).
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 ATS3	TAPAn ATS2	Description
0	0	INT signal while the triangle wave is falling (counting down)
0	1	INT signal while the triangle wave is rising (counting up)
1	0	INT signal while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal and valley interrupt TAPATIVLY0 signal while the triangle wave is rising (counting up) or falling (counting down)

TAPAn ATS1	TAPAn ATS0	Description
0	0	INT signal while the triangle wave is falling (counting down)
0	1	INT signal while the triangle wave is rising (counting up)
1	0	INT signal while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal and valley interrupt TAPATIVLY0 while the triangle wave is rising (counting up) or falling (counting down)

27.3.4 TAPAnFLG — TAPA Flag Register

This flag register is for asynchronous Hi-Z control.

Access: This register can only 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 27.13 TAPAnFLG Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned.
10 to 8	TAPAnHOFm	Hi-Z Control Monitor (m = 0, 1, 2) These bits are used to monitor the Hi-Z control status. 0: The present output of TAPAnTHZOUTm is high level 1: The present output TAPAnTHZOUTm is low level.
7 to 1	Reserved	When read, the value after reset is returned.
0	TAPAnACE	Asynchronous Hi-Z Control Enable This bit indicates the status of the asynchronous Hi-Z control signal TAPATHASIN. 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

27.3.5 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 27.14 TAPAnACWE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When written, 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.

27.3.6 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 write only 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 27.15 TAPAnACTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, 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 to this bit is ignored (no function). 1: Enables asynchronous Hi-Z control when TAPAnACE is 1.

27.3.7 TAPAnACTT — TAPA Asynchronous Hi-Z Control Stop Trigger Register

This bit enables the stop trigger for asynchronous Hi-Z control.

Access: This register is write only 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	R/W

Table 27.16 TAPAnACTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When written, 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 to this bit is ignored (no function). 1: Disables asynchronous Hi-Z control when TAPAnACE is 0.

27.3.8 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 write only 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 27.17 TAPAnOPHS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When written, 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 to this bit is ignored (no function). 1: Starts Hi-Z control.

27.3.9 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 write only 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 27.18 TAPAnOPHT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When written, 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 to this bit 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.

27.3.10 TAPAnEMU — TAPA Emulation Register

This register controls SVSTOP for emulation.

Access: This register can be read/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 27.19 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 written, write the value after reset.

NOTE

For details about PIC-related registers, see the section of each function explanation.

27.4 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.

27.4.1 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.

27.4.2 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 an external error detection signal is received, an interrupt is generated and the level of the motor control outputs is simultaneously 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.

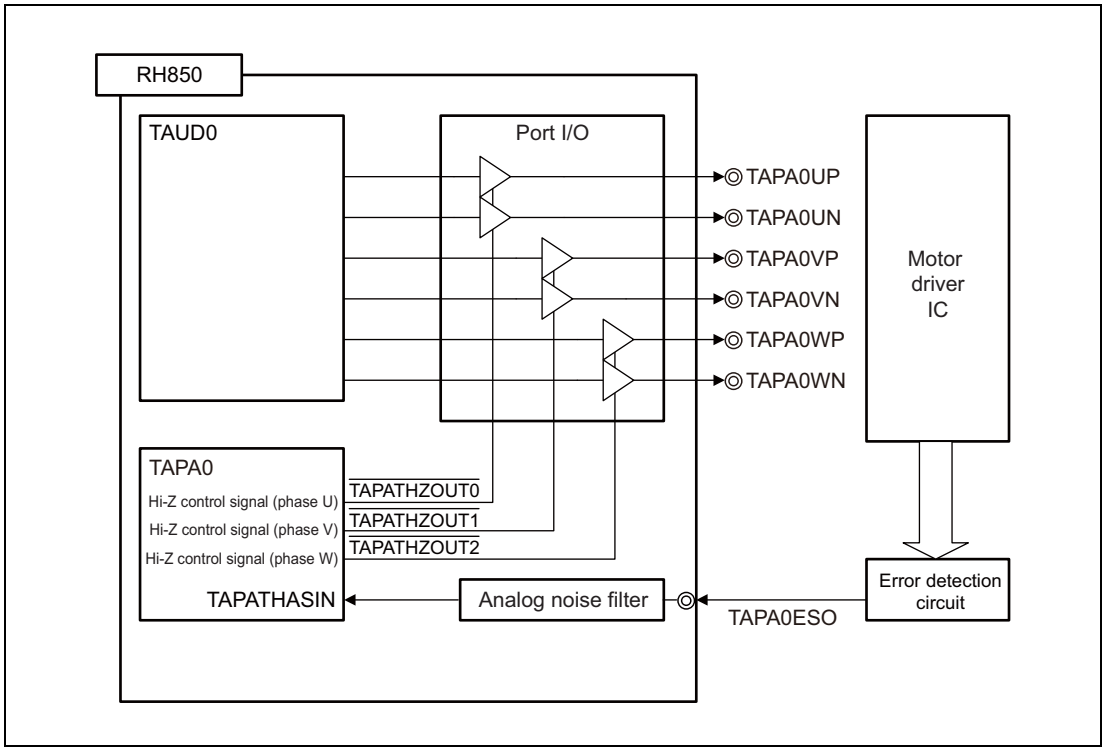


Figure 27.5 System Configuration Example of Asynchronous Hi-Z Control for Pin Input

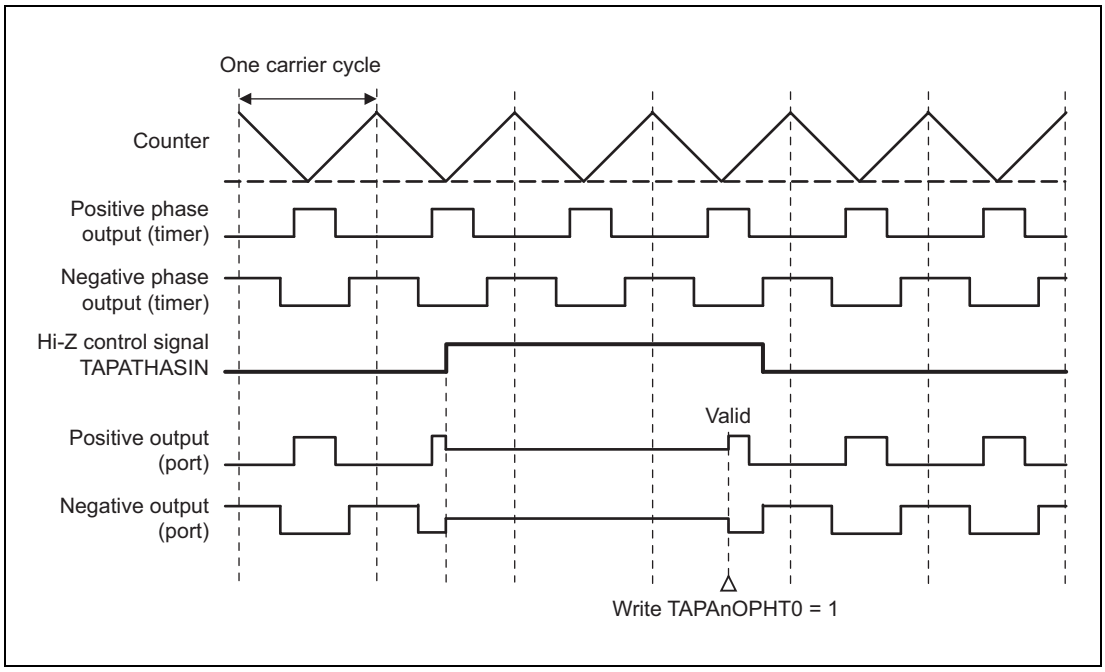
27.4.3 Basic Operation

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.

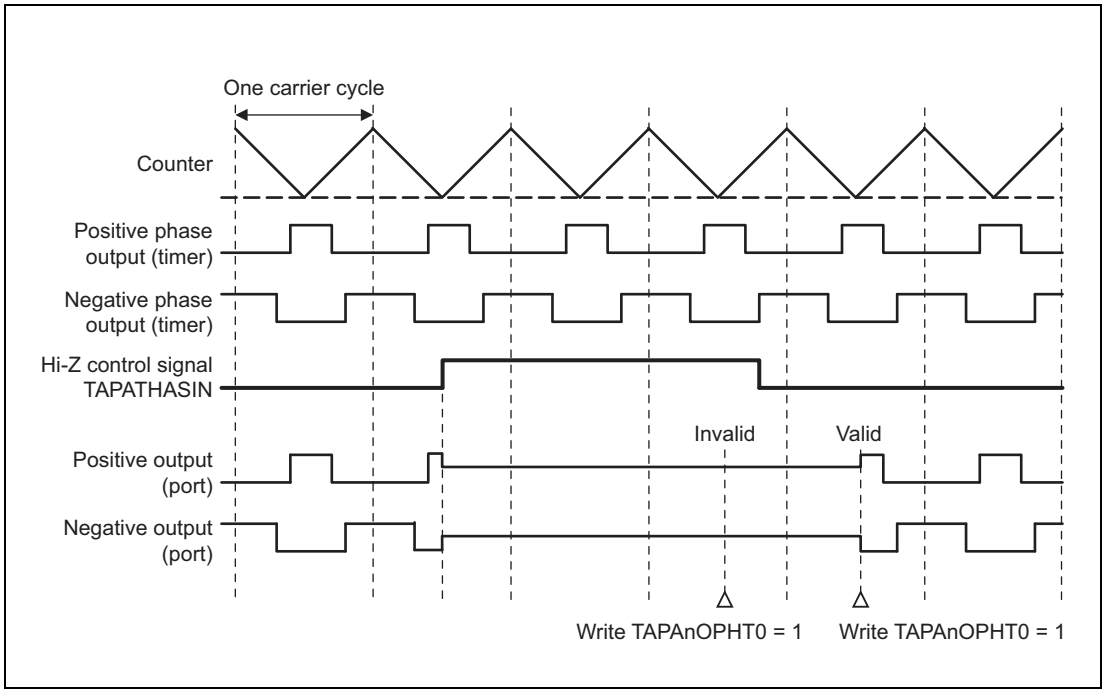
(1) Operation when TAPAnCTL0.TAPAnDCM = 0, TAPAnDCP = 1, and TAPAnDCN = 0



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).

27.4.4 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

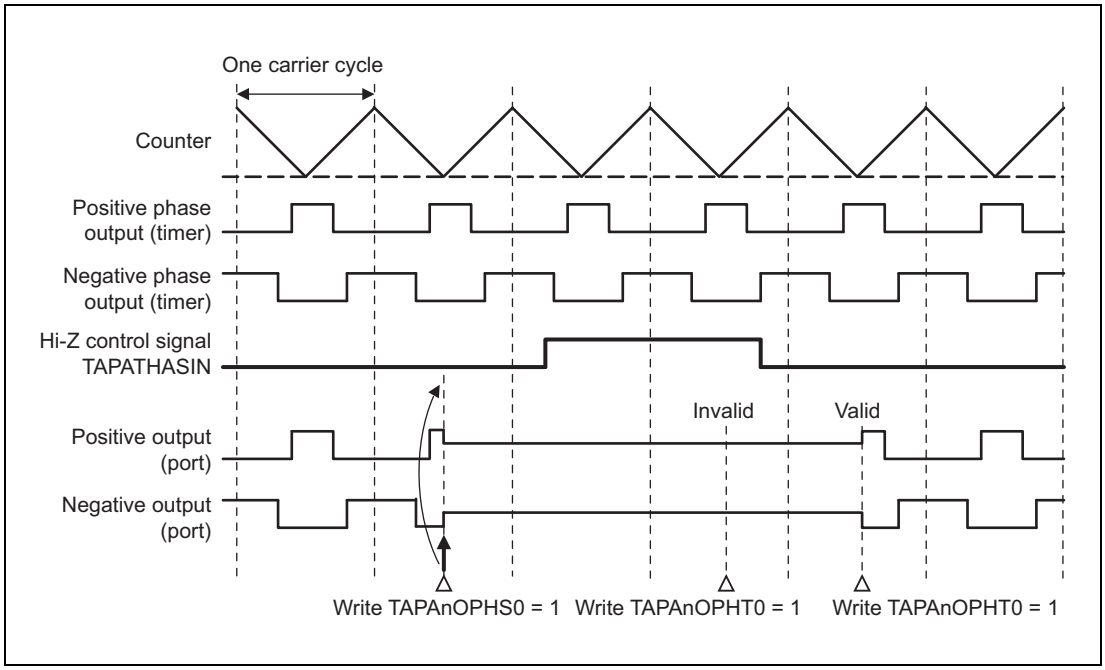
TAPAnDCM	Function
0/1	Writing 1 to TAPAnOPHS0 starts Hi-Z control and forcibly stops the motor control outputs (Hi-Z output).

(2) Function of Hi-Z control stop trigger bit TAPAnOPHT.TAPAnOPHT0

Whether the Hi-Z control stop trigger is valid or invalid 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 outputs. If TAPATHASIN is active, writing 1 to TAPAnOPHT0 is ignored.

(3) Operation when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0



The motor control outputs are forcibly stopped (Hi-Z output) when 1 is written to TAPAnOPHS0.

After that, the levels of the motor control outputs remain Hi-Z even if a 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 a 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).

27.4.5 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.	Writing to TAPAnACTS is enabled.
	Set up the TAPAnACTS register. Set TAPAnACTS to 1.	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.	Writing to TAPAnACTT is enabled.
	Set up the TAPAnACTT register. Set TAPAnACTT to 1.	Asynchronous Hi-Z control stopped (TAPAnFLG.TAPAnACE = 0)

Restart

27.4.6 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.

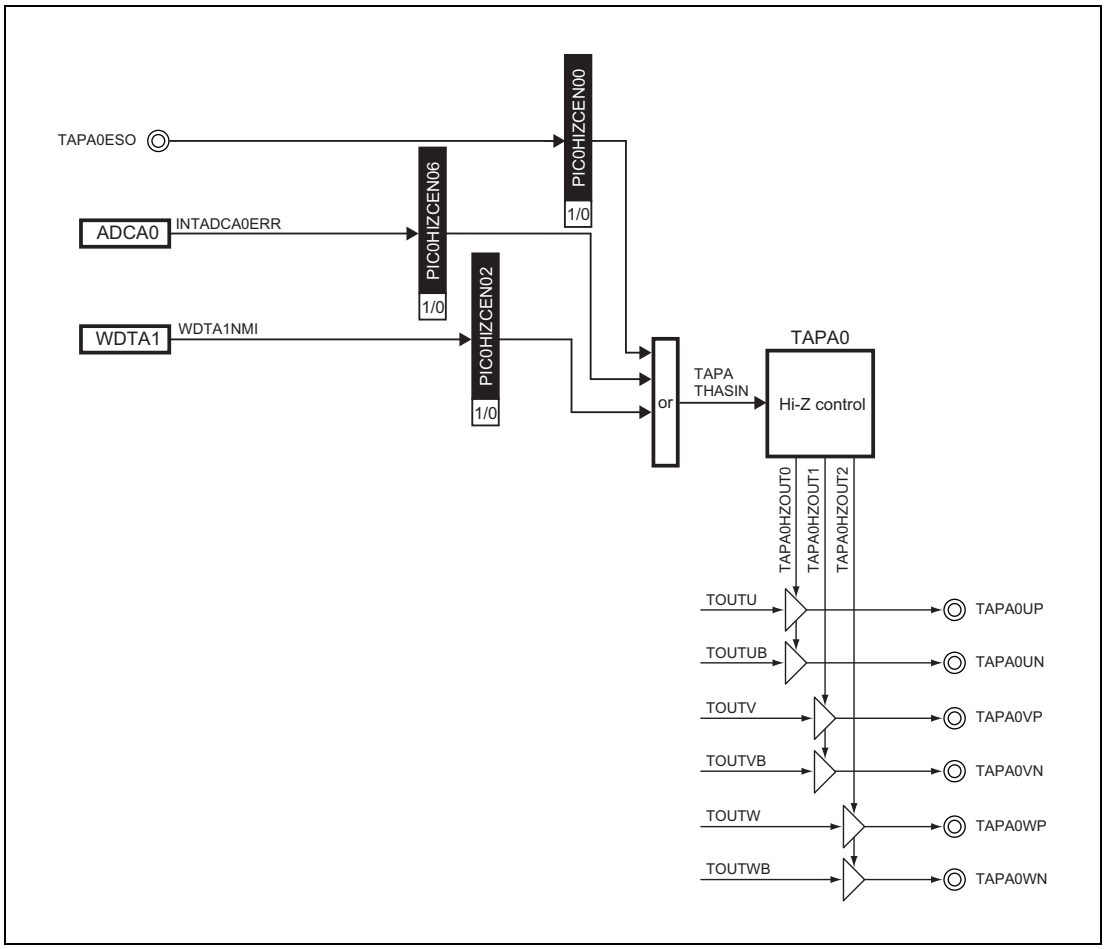


Figure 27.6 Hi-Z Control Block Diagram

Switching into a 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 the respective description.

27.4.7 Registers

27.4.7.1 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 27.20 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, 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 to these bits, write the value after reset.
2	PIC0HIZCENn2	Hi-Z Output Control by the WDTA1NMI Interrupt Signal Enable 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
0	PIC0HIZCENn0	Hi-Z Output Control by the TAPA0ESO External Input Enable 0: Disable 1: Enable

27.5 INT Signal Output Selection Function

27.5.1 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 27.1.7, Internal Output Signal**.

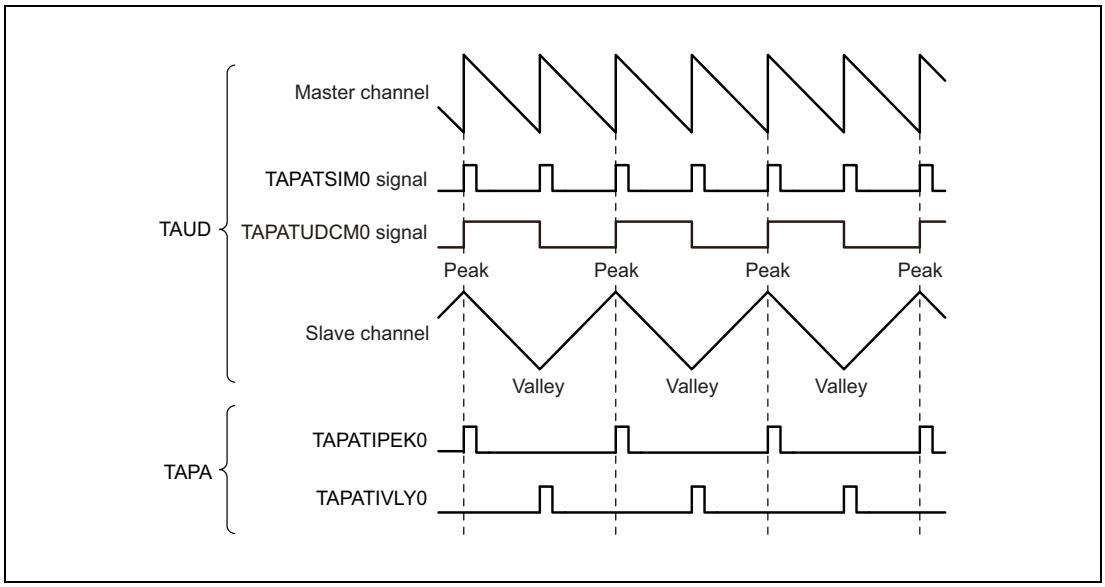


Figure 27.7 Basic Timing Chart of Signals for the INT Signal Output Selection Function

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.

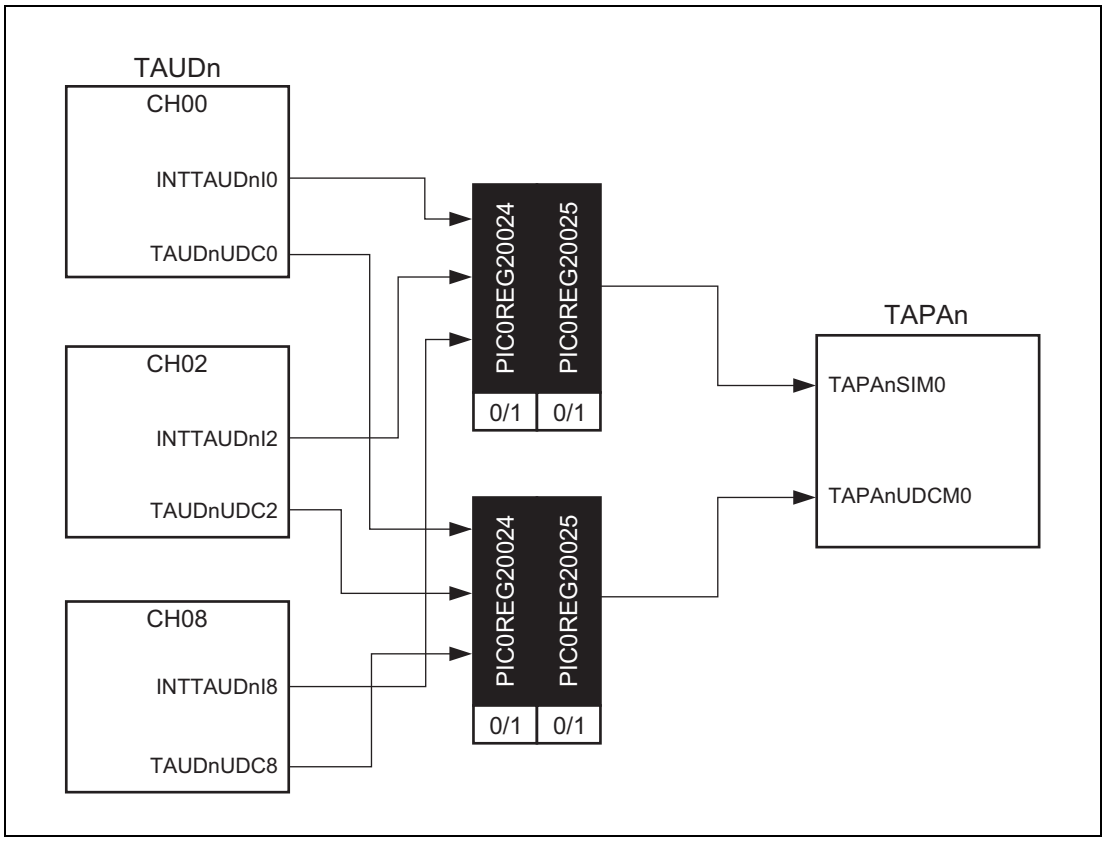
CAUTION

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.

27.5.2 Block Diagram

TAUDn and TAPAn are connected in the registers shown below by INT signal output selection function.



27.5.3 Registers

27.5.3.1 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 27.21 PIC0REG2n0 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	*1
25, 24	PIC0REG2n025 PIC0REG2n024	Select the TAUDn channel used by TAPATSIM0 and TAPATUDCM0. 00: Not selected 01: TAUD0 channel 0 selected 10: TAUD0 channel 2 selected 11: TAUD0 channel 8 selected
23 to 0	Reserved	*1

Note 1. Some of the bits defined as 0 in the PIC0REG2n0 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

27.6 A/D Conversion Trigger Selection Function

This function outputs the A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1 from the signals TAPATCDENS0 and TAPATCDENS1, which are connected to a compare match interrupt with TAUD's triangular carrier cycle.

27.6.1 Configuration of A/D Conversion Trigger Selection Function

Table 27.22 Signals Used for TAPATADOUT Generation

Output Signal	Slave Match Detection Signal	Valley Interrupt Signal
TAPATADOUT0	TAPATCDENS0	TAPATIVLY0
TAPATADOUT1	TAPATCDENS1	TAPATIVLY0

Table 27.23 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 and valley interrupt TAPATIVLY0 from TAPATADOUT1 while the triangle wave is rising (counting up) or falling (counting down).

Table 27.24 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 and valley interrupt TAPATIVLY0 from TAPATADOUT0 while the triangle wave is rising (counting up) or falling (counting down).

27.6.2 Block Diagram

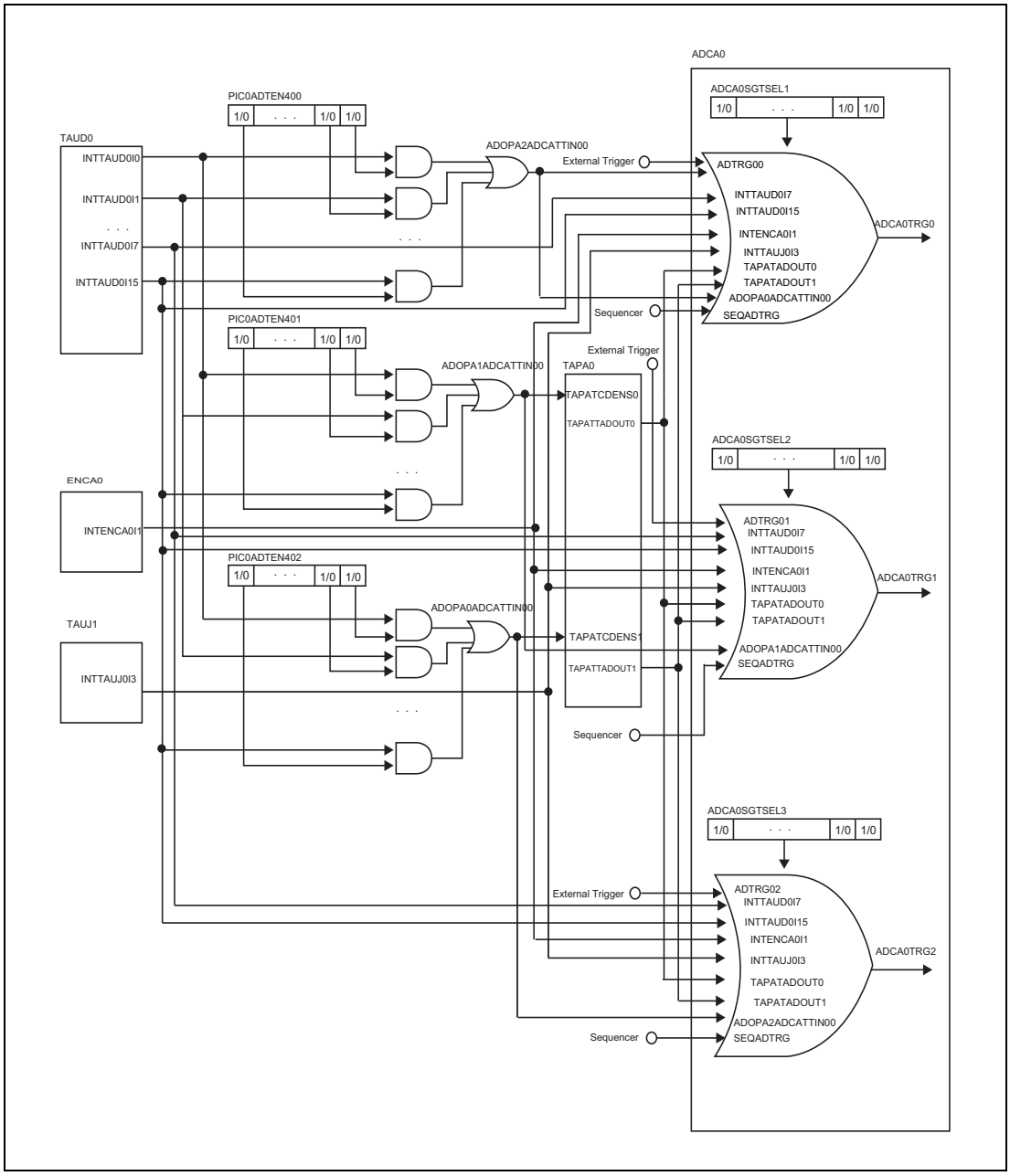


Figure 27.8 Block Diagram of A/D Conversion Trigger Selection Function

NOTE

See **Section 29.3.4.1, ADCAnSGTSELx — Scan Group x Start Trigger Control Register x** for the details on the settings of ADCA0SGTSEL register.

27.6.3 Waveforms of A/D Conversion Trigger Output Control Operation in Triangle PWM Mode

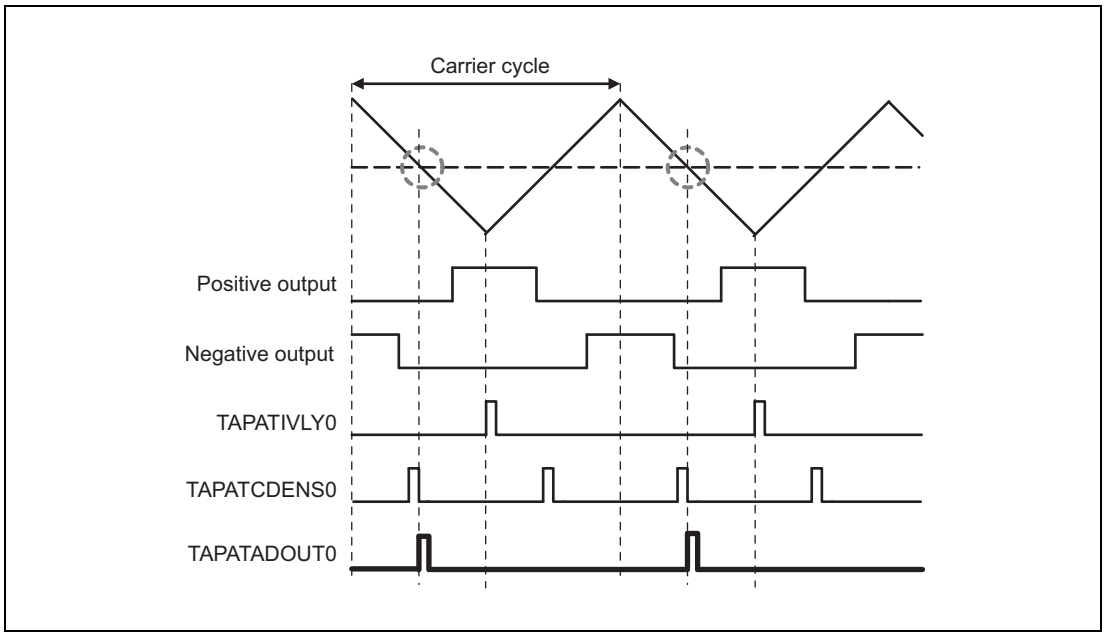


Figure 27.9 TAPAnATS[1:0] bits = 00_B: Output of INT Signal while the Triangle Wave is Falling (Counting Down)

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.

In this case, no A/D conversion trigger signal is output while the triangle wave is rising (counting up).

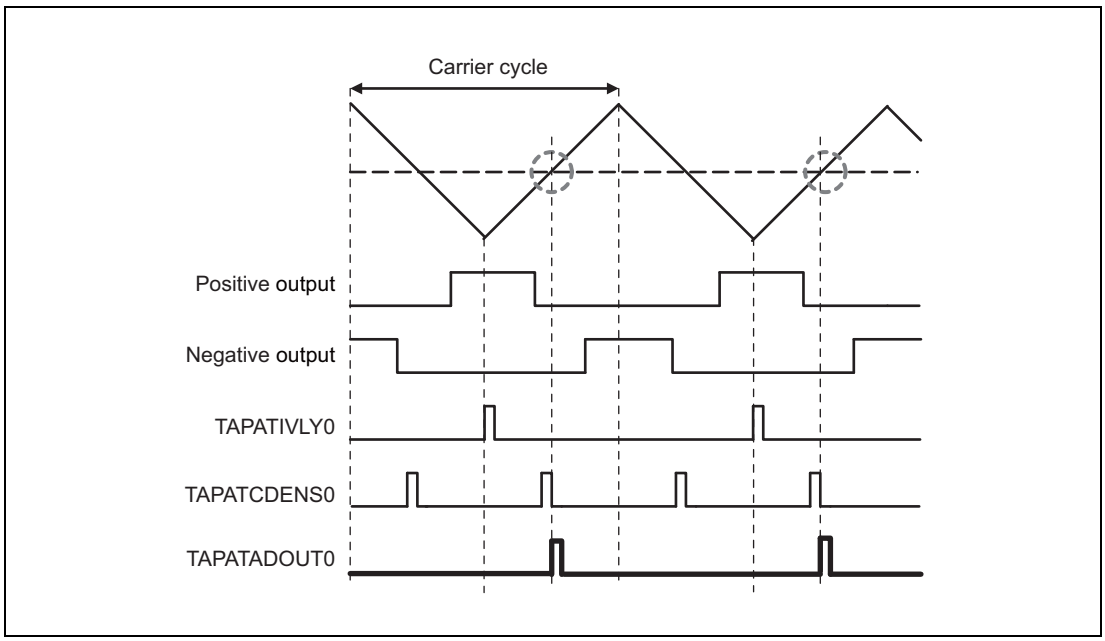


Figure 27.10 TAPAnATS[1:0] bits = 01_B: Output of INT Signal while the Triangle Wave is Rising (Counting Up)

While the triangle wave is rising (counting up), the TAPATCDENS0 and TAPATCDENS1 signals are output as A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

In this case, no A/D conversion trigger signal is output while the triangle wave is falling (counting down).

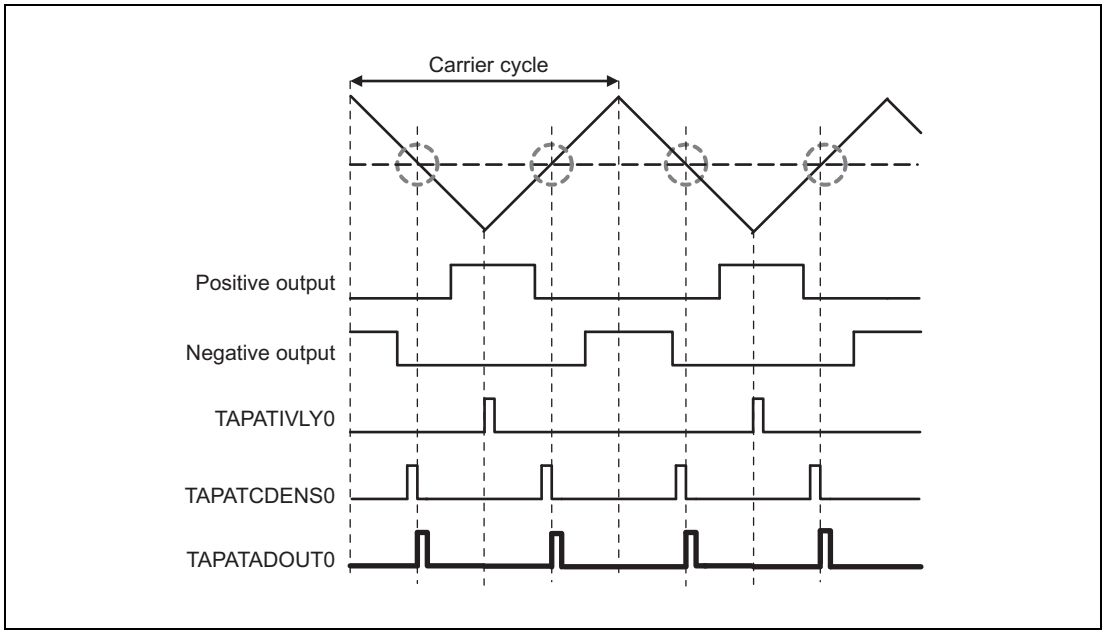


Figure 27.11 TAPAnATS[1:0] bits = 10_B: Output of INT Signal while the Triangle Wave is Rising (Counting Up) or Falling (Counting Down)

The signals TAPATCDENS0 and TAPATCDENS1 are output as the A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

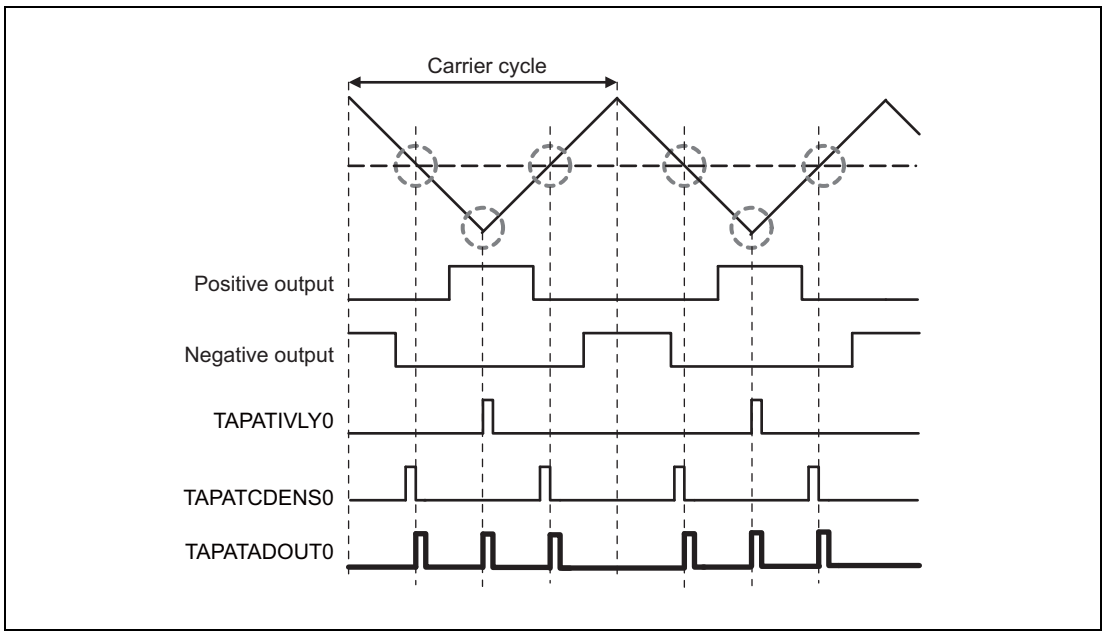


Figure 27.12 TAPAnATS[1:0] bits = 11_B: Output of INT Signal and Valley Interrupt while the Triangle Wave is Rising (Counting Up) or Falling (Counting Down)

The signals TAPATCDENS0 and TAPATCDENS1 and valley interrupt TAPATIVLYn are output as the A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

27.6.4 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).	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.

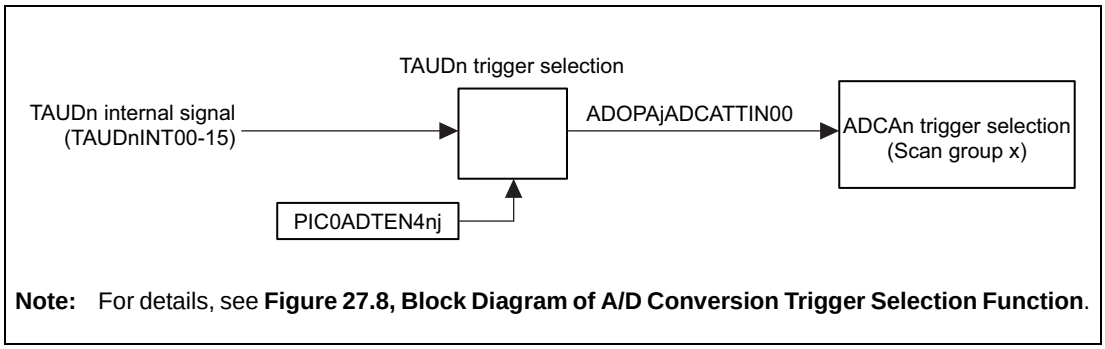
Restart

27.7 ADCA Trigger Selection Function

27.7.1 Outline of Functions

This function generates ADCA hardware trigger signals by using TAUDn channel output.

27.7.2 Configuration



Note: For details, see **Figure 27.8, Block Diagram of A/D Conversion Trigger Selection Function**.

Figure 27.13 Block Diagram of ADCA Trigger Selection Function

27.7.3 Registers

27.7.3.1 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
	PIC0ADTEN4nj15	PIC0ADTEN4nj14	PIC0ADTEN4nj13	PIC0ADTEN4nj12	PIC0ADTEN4nj11	PIC0ADTEN4nj10	PIC0ADTEN4nj09	PIC0ADTEN4nj08	PIC0ADTEN4nj07	PIC0ADTEN4nj06	PIC0ADTEN4nj05	PIC0ADTEN4nj04	PIC0ADTEN4nj03	PIC0ADTEN4nj02	PIC0ADTEN4nj01	PIC0ADTEN4nj00
Value after 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 27.25 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.

27.7.4 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 to select 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 AD is stopped.

(3) Setting of the A/D conversion trigger selection control register (ADCA0SGTSELj):
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 AD is stopped.

(4) Allowance of TAUD0 timer operation
Each channel in the TAUD0 timer set in (1) starts.

27.7.5 Setup Flow

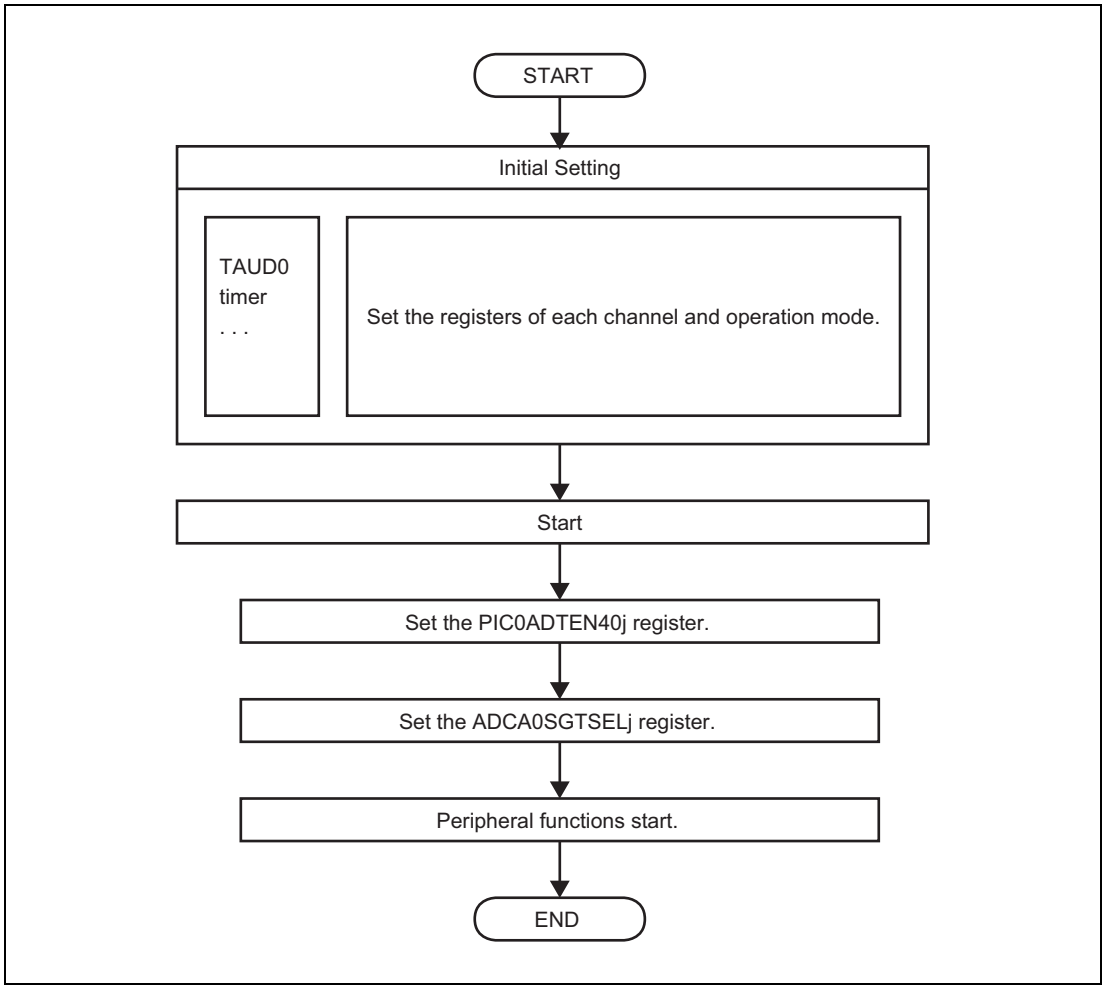


Figure 27.14 Setup Flow (j = 0 to 2)

27.8 Simultaneous Start Trigger Function

27.8.1 Outline of Functions

The timers (TAUD0, TAUJ1, ENCA0) can be simultaneously started in any combinations.

27.8.2 Configuration

(1) Configuration

Table 27.26 Configuration of Simultaneous Start Trigger Function

Configuration/Timer Function	Timer
Configuration of Timer	TAUD0, TAUJ1, ENCA0

(2) Block Diagram

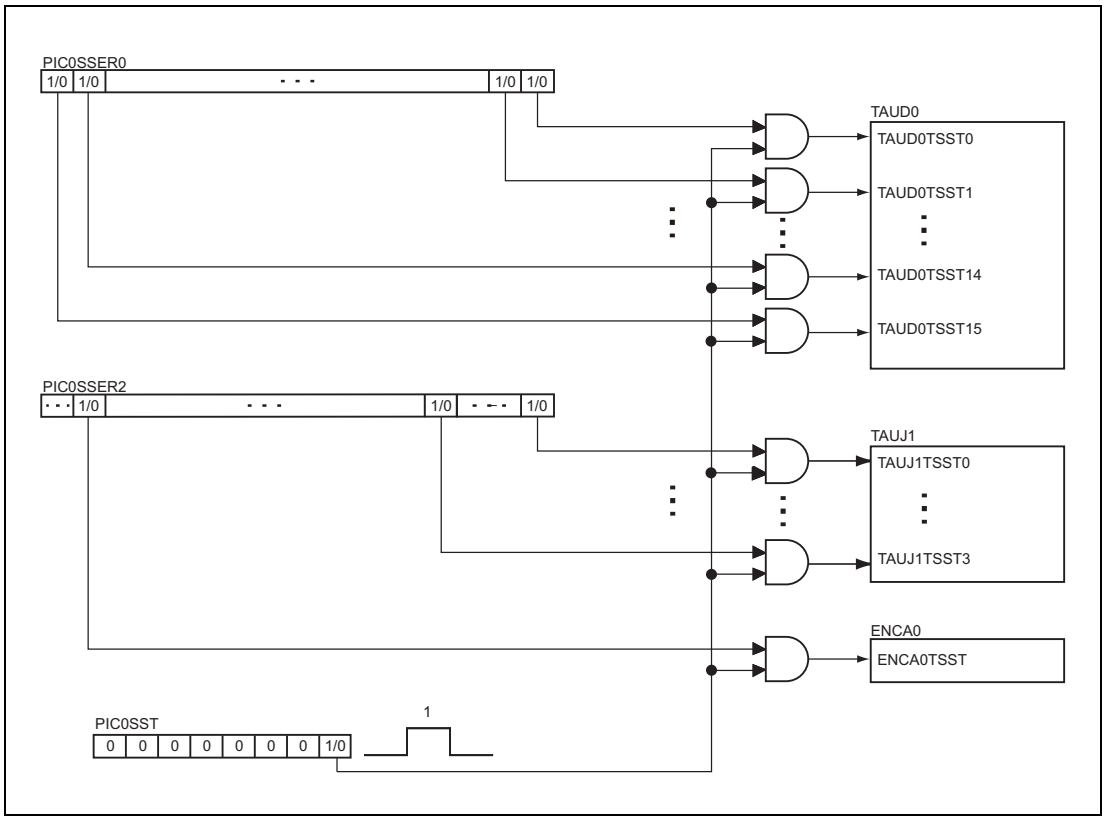


Figure 27.15 Block Diagram of Simultaneous Start Trigger

27.8.3 Registers

27.8.3.1 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 27.27 PIC0SSER0 Register Contents

Bit Position	Bit Name	Function
15 to 0	PIC0SSER0m	Enable a simultaneous start trigger for the CHm in the TAUD0 timer. 0: Simultaneous start trigger is disabled. 1: Simultaneous start trigger is enabled.

27.8.3.2 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 27.28 PIC0SSER2 Register Contents

Bit Position	Bit Name	Function
15	Reserved	When read, the value after reset is returned. When writing to this bit, 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 to these bits, write the value after reset.
3 to 0	PIC0SSER20m	Set a simultaneous start trigger for the CHm in the TAUD1 timer. 0: Simultaneous start trigger is disabled. 1: Simultaneous start trigger is enabled.

27.8.3.3 PIC0SST — Simultaneous Start Control Register

Access: This register is write-only 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 27.29 PIC0SST Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, 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).

27.8.4 Example of Operation

(1) Example of timer operation:
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 to simultaneously start them.

(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.

27.8.5 Setup Flow

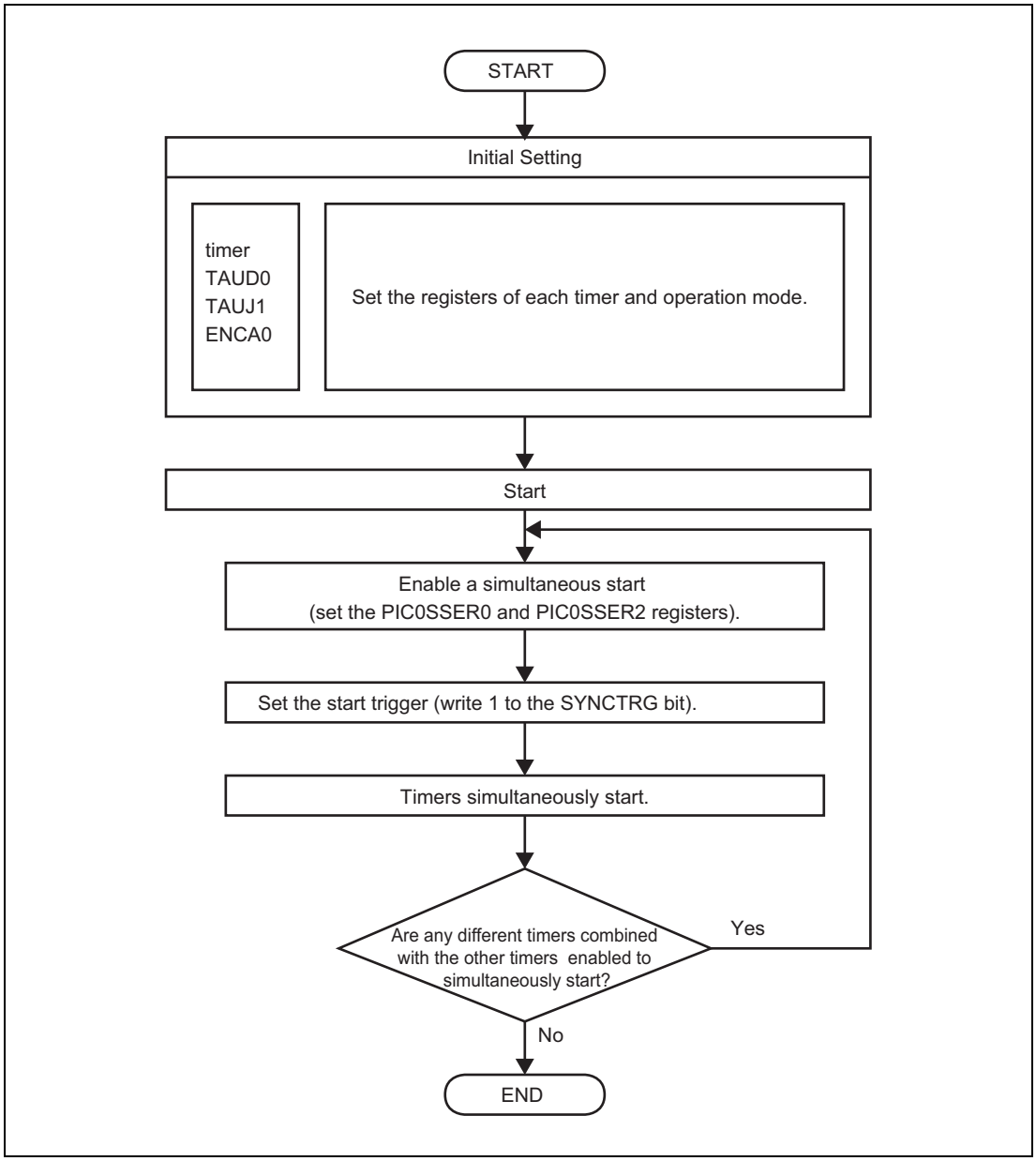


Figure 27.16 Setup Flow

27.9 Trigger & Pulse Width Measuring Function

27.9.1 Overview of Functions

This function allows measurement of trigger periods by inputting the trigger signal output from ENCA0 to TAUJ1 and TAUD0.

27.9.2 Configuration

(1) Configuration

Table 27.30 Configuration of Trigger & Pulse Width Measuring Function

Configuration/Timer Function	Timer
Configuration of Timer	ENCA0, TAUD0, TAUJ1

Table 27.31 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
	01	TINm input pulse interval measurement function	S	ENCAT0IEC
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

(2) Block Diagram

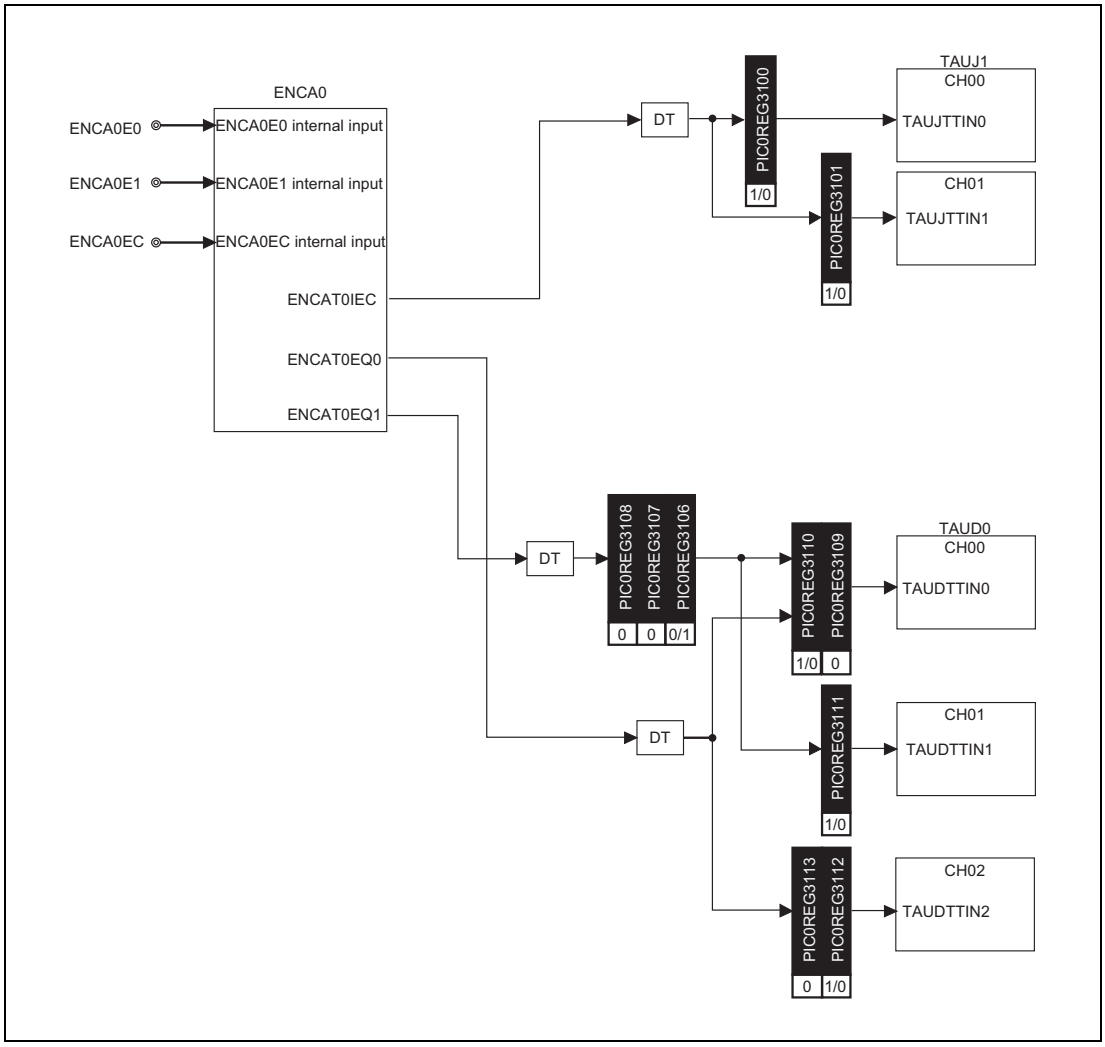


Figure 27.17 Block Diagram of Trigger & Pulse Width Measuring Function

27.9.3 Registers

27.9.3.1 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 27.32 PIC0REG31 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
13, 12	PIC0REG3113 to PIC0REG3112	Selects 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	Selects 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	Selects 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	Selects 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 to these bits, write the value after reset.

Table 27.32 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

27.9.4 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 in the PIC, described below, is used to convert the trigger signal input to TINm into the 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 the level-sensitive toggle signal.

As shown in **Figure 27.18, Operation of DT Circuit**, the output signal is toggled on each input trigger signal generation.

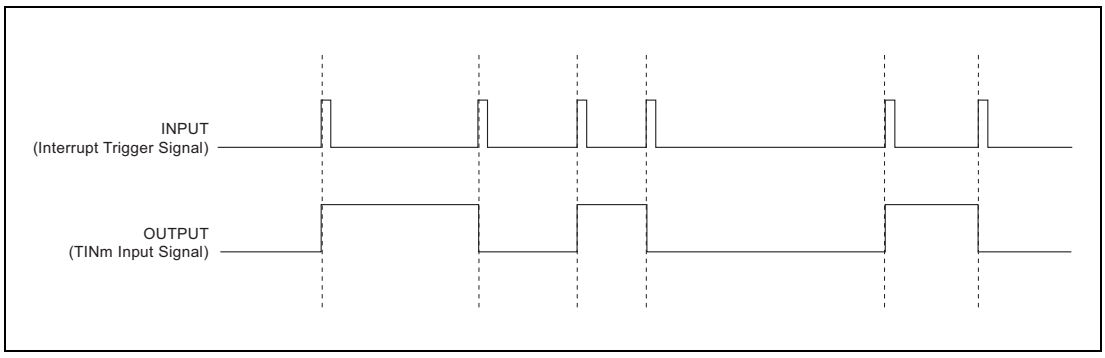


Figure 27.18 Operation of DT Circuit

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.

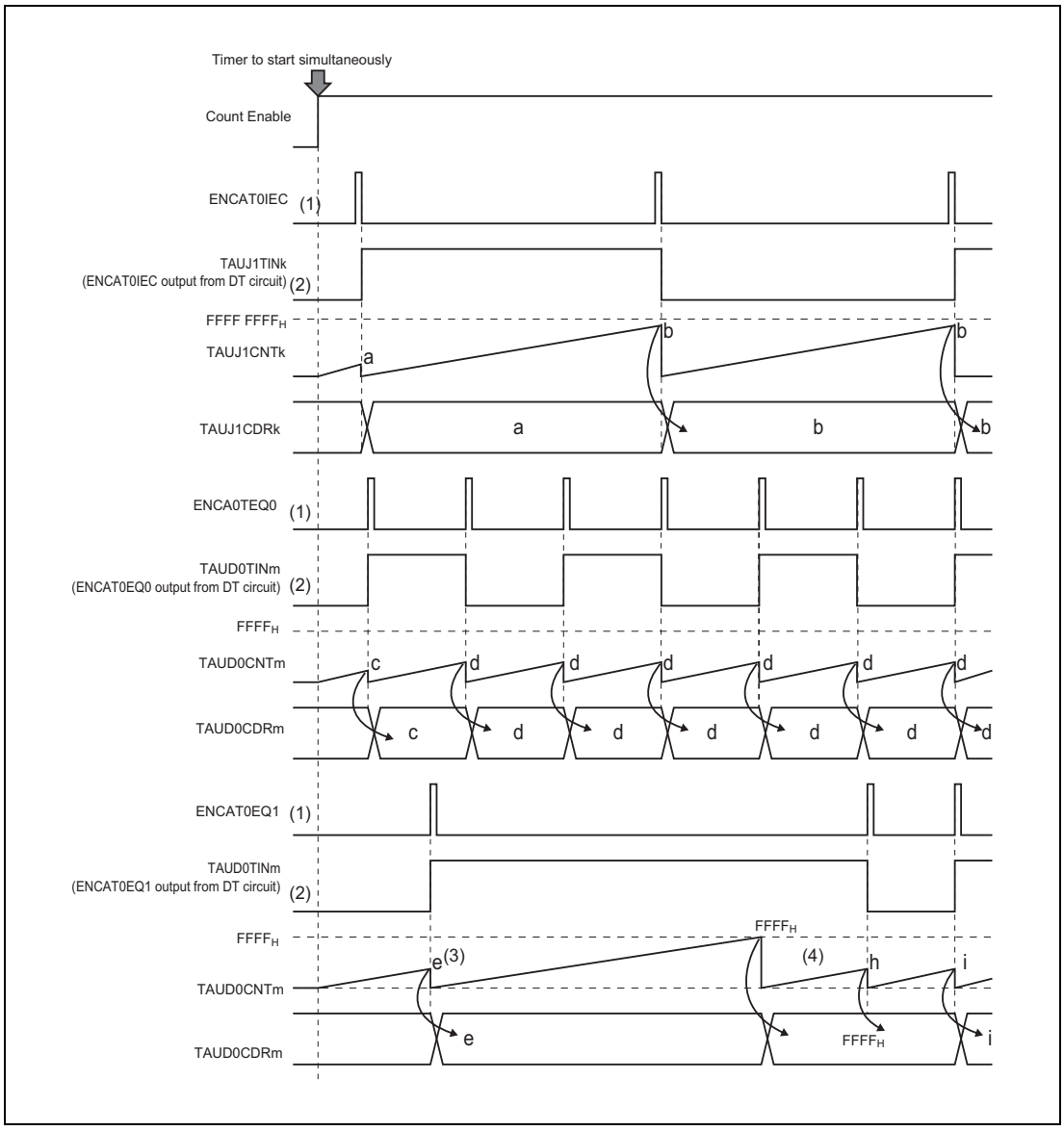


Figure 27.19 Operation Example of the Function of Trigger and Pulse Width Measurement (m = 0 to 2, k = 0, 1)

(1) Following signal is output from ENCA0 as a trigger:

- 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 the level-sensitive toggle signal by the DT circuit and is output to TINm of TAUD0 and TAUJ1.

(3) By setting the both TINm edges (rising and falling edges) of TAUD0 and TAUJ1 as valid, the CNTm value is captured into CDRm on the TINm toggle timing and cleared to 0000_H. This operation 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 to the 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.

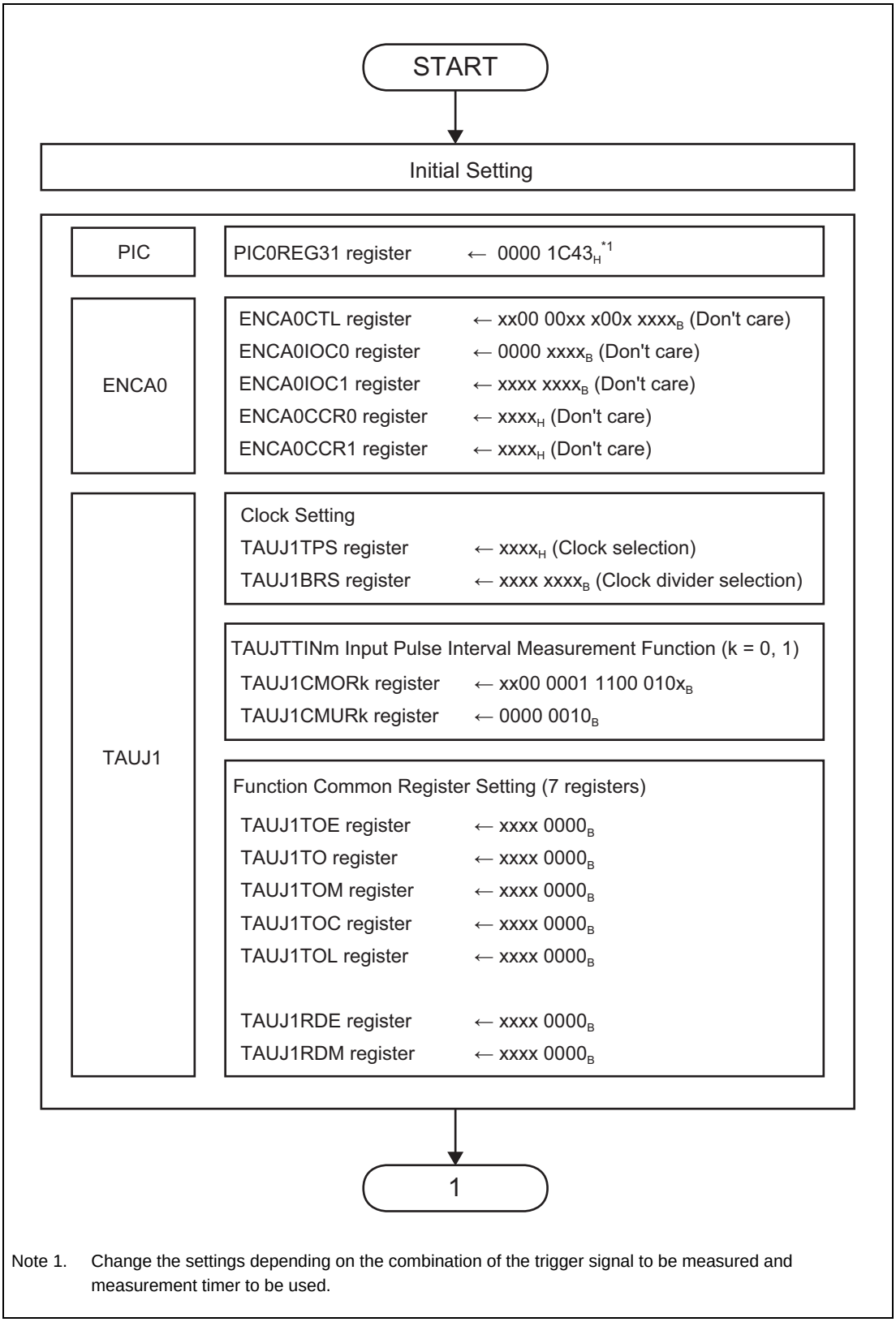
Table 27.33 Combinations of Trigger Signals and Measurement Timers

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 = 1
	TAUD0 CH1		PIC0REG3111 = 1

27.9.5 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. Change the settings indicated with the symbol “*” depending on the trigger signal to be measured and measurement timer to be used. For the combinations of the trigger signals and measurement timers, see **Table 27.33, 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 27.20 Setup Flow

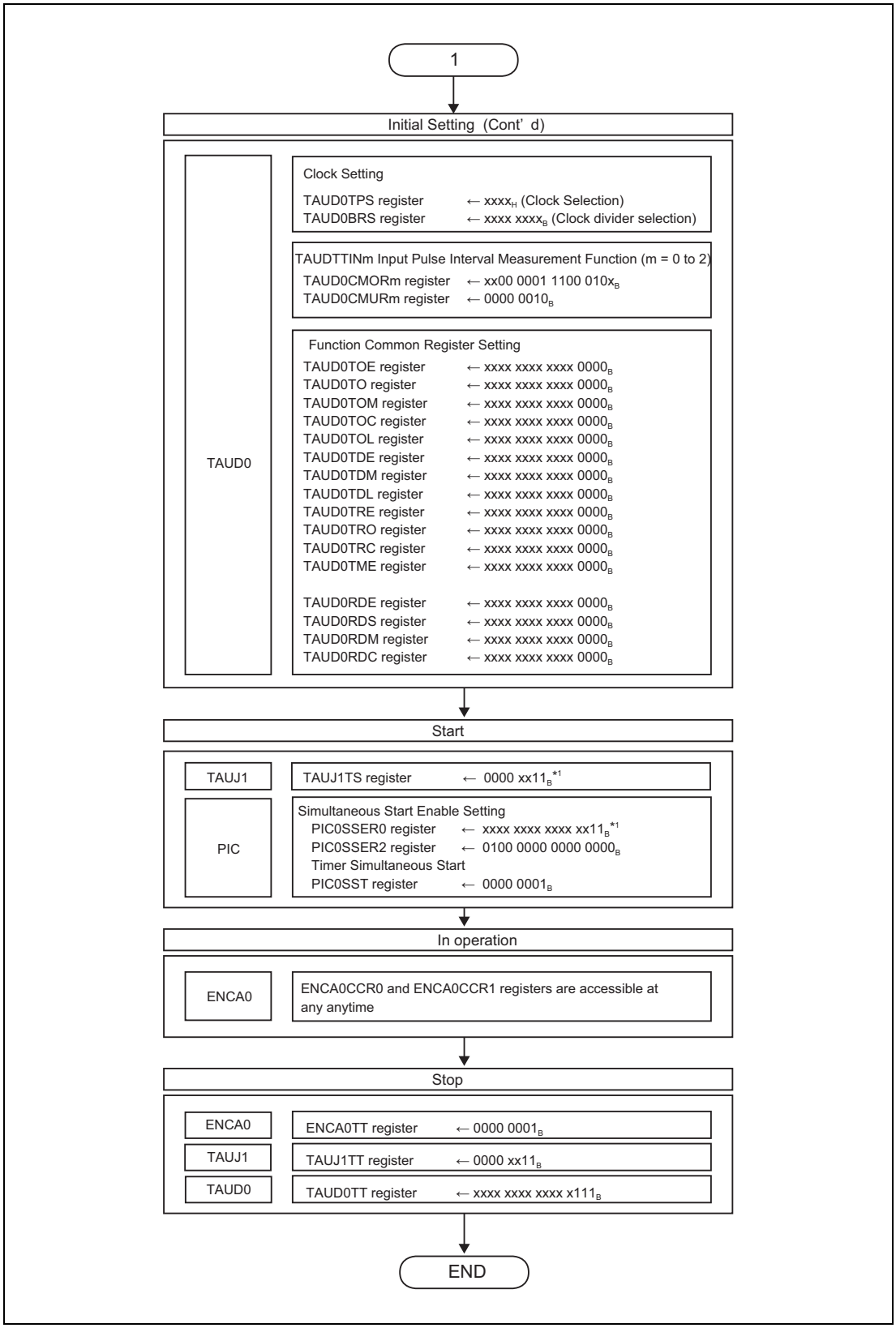


Figure 27.21 Setup Flow (Cont'd)

Change the settings depending on the combination of the trigger signal to be measured and measurement timer to be used.

27.9.6 Example of Setting Up Operation Functions

The register setting examples are shown below.

The tables in this section indicate the setting example to measure the pulse interval, which applies to all the following combinations. For the combinations of the trigger signals and measurement timers, see **Table 27.33, 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 27.34 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 of 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 of 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 effective edge for capture trigger 1 (ENCA0I1)
	1, 0	ENCA0TIS[1:0]	Don't care	Selects the effective edge for capture trigger 0 (ENCA0I0)
ENCA0IOC1	7	ENCA0SCE	Don't care	Enables the special encoder clear
	6	ENCA0ZCL	Don't care	Selects the clear level of Z phase for a special encoder clear
	5	ENCA0BCL	Don't care	Selects the clear level of B phase for a special encoder clear
	4	ENCA0ACL	Don't care	Selects the clear level of A phase for a special encoder 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 27.35 TAUJ1 Setting (k = 0, 1)
TAUJ1 (TAUDTTINm 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	1	
	10, 9, 8	TAUJSTS[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 24, 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 27.36 TAUD0 Setting (m = 0 to 2)
TAUD0 (TAUD0TTINm 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	1	
	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 as specified above for this function.

Other control bits have fixed values as specified above. For details, see **Section 23, 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 27.37 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

27.10 A/D Trigger Encoder Capture Function

27.10.1 Overview of Functions

The value of encoder counter synchronizing with A/D conversion can be obtained by using an A/D conversion trigger signal as a capture signal of the ENCA0.

27.10.2 Configuration

(1) Configuration

Table 27.38 Configuration of A/D Trigger Encoder Capture Function

A/D Converter	Encoder Timer
ADCA0	ENCA0

(2) Block Diagram

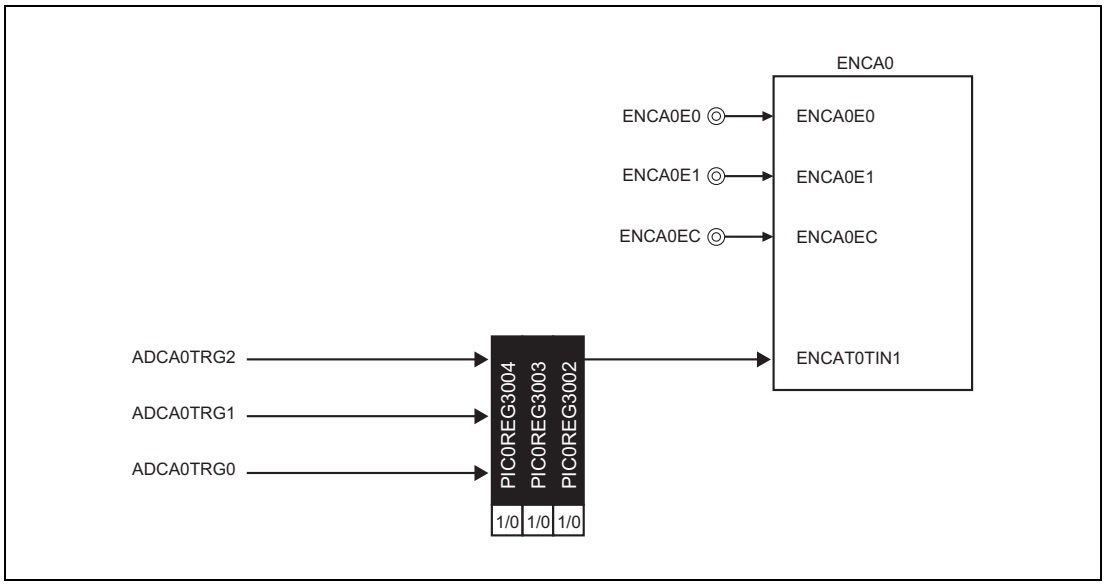


Figure 27.22 Block Diagram of A/D Trigger Encoder Capture Function

27.10.3 Registers

27.10.3.1 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 27.39 PIC0REG30 Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
4 to 2	PIC0REG3004 to PIC0REG3002	Selects an input signal to ENCAT0TIN1.
1, 0	Reserved	When read, the value after reset is returned. When writing to these bits, 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

27.10.4 Example of Operation

The AD trigger encoder capture function is implemented by connecting the AD conversion trigger signal ADCAnTRGi (n = 0, i = 0 to 2) to ENCA0.

CAUTION

When using this function, the ENCA0 interrupt signal ENCAT0INT1 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 → ENCAT0INT1 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.

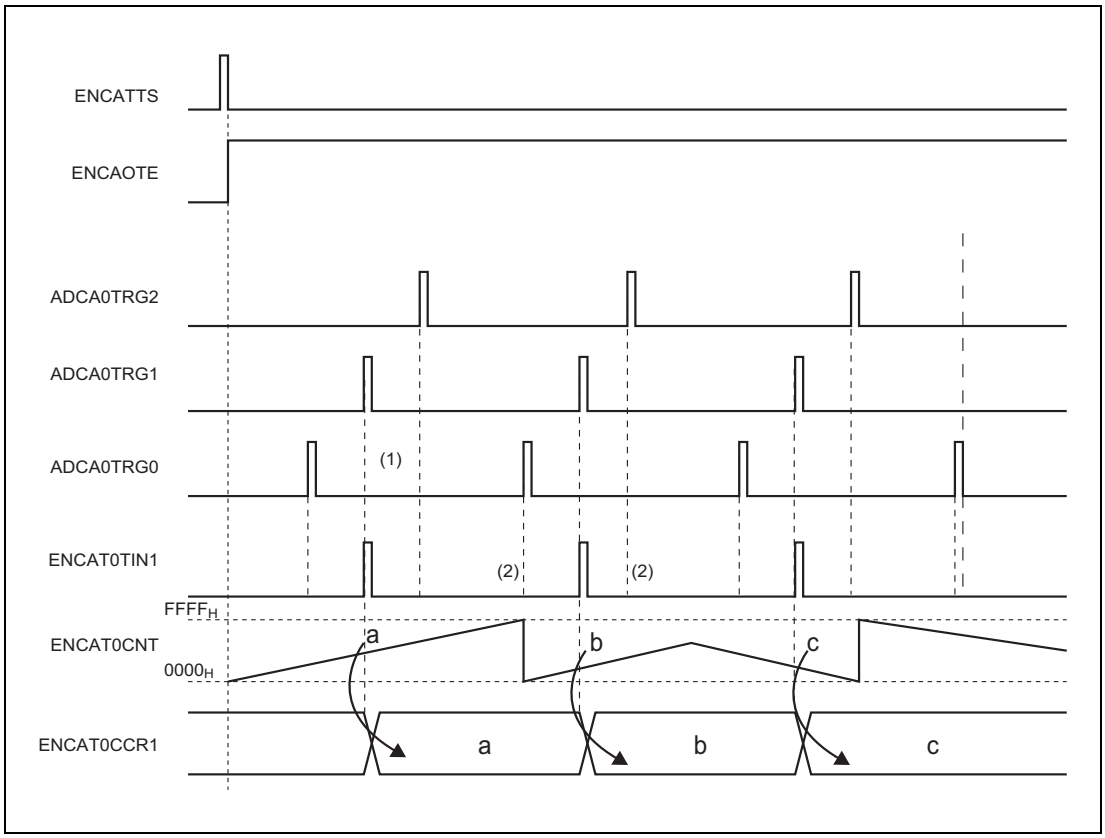


Figure 27.23 Operation Example of Trigger Encoder Capture function

(1) When ADCA0TRG1 is selected as the ENCA0 capture trigger 1 signal ENCAT0TIN1, the valid ADCA0TRG1 is input to ENCA0 as the ENCAT0TIN1 signal and ENCA0 is captured.

(2) When a hardware trigger signal (ADCA0TRG0, ADCA0TRG2) other than ADCA0TRG1 is generated, the ENCAT0TIN1 signal is not generated and ENCA0 is not captured.

27.10.5 Setup Flow

The setup flow in this section shows the general setup flow to perform capture operation of encoder timer ENCA0 based on the ADCA0TRG1 signal. Change the settings indicated with the symbol “*” depending on the hardware trigger to perform the capture operation.

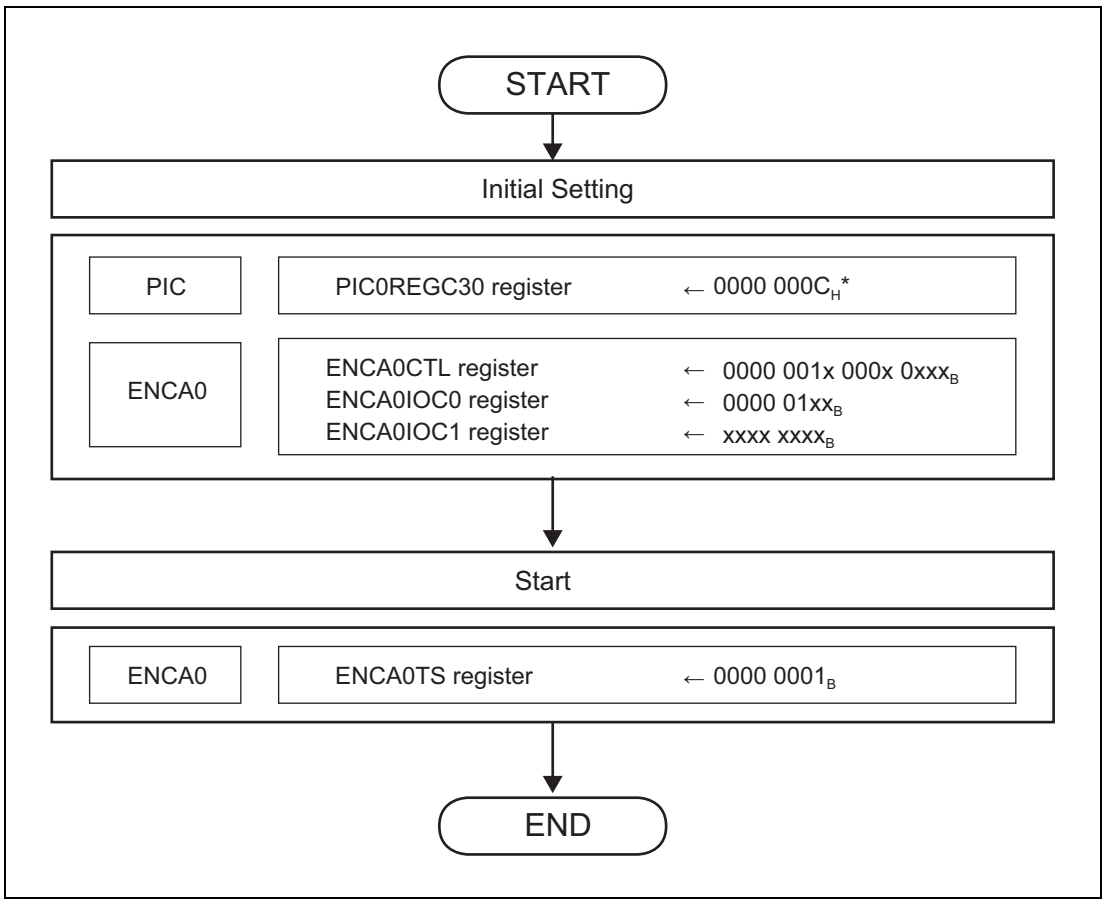


Figure 27.24 Setup Flow

27.10.6 Example of Setting Up Operation Functions

The register setting examples are shown below.

The tables in this section show the setting value to perform 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 27.40 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 of capture operation
	6, 5		0	Fixed to 0
	4	ENCAnLDE	Don't care	Enables or disables reload operation when ENCAnCCR0 register underflow is generated
	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 effective edge of capture trigger 1 (ENCATTIN1) for the rising edge detection
	1, 0	ENCAnTIS[1:0]	Don't care	Selects the effective edge of capture trigger 0 (ENCATTIN0)
ENCAnIOC1	7	ENCAnSCE	Don't care	Enables the special encoder clear
	6	ENCAnZCL	Don't care	Selects the clear level (input level) of Z phase for a special encoder clear
	5	ENCAnBCL	Don't care	Selects the clear level (input level) of B phase for a special encoder clear
	4	ENCAnACL	Don't care	Selects the clear level (input level) of A phase for a special encoder 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 clear input (A or B phase) edge
PIC0REG30	4	PIC0REG3004	Don't care	Selects ADCA0 trigger signal of ENCAT0TIN1
	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 set arbitrarily.

27.11 Three-Phase PWM Output with Dead Time

27.11.1 Functional Overview

This feature generates each of the set signals (assert timing signals) and clear signals (deassert 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 allows more flexible PWM output with dead time possible.

27.11.2 Configuration

The unit and channel configuration for this feature are shown below. (n = 0)

Table 27.41 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 27.42 TAUDn Configuration (1/2)

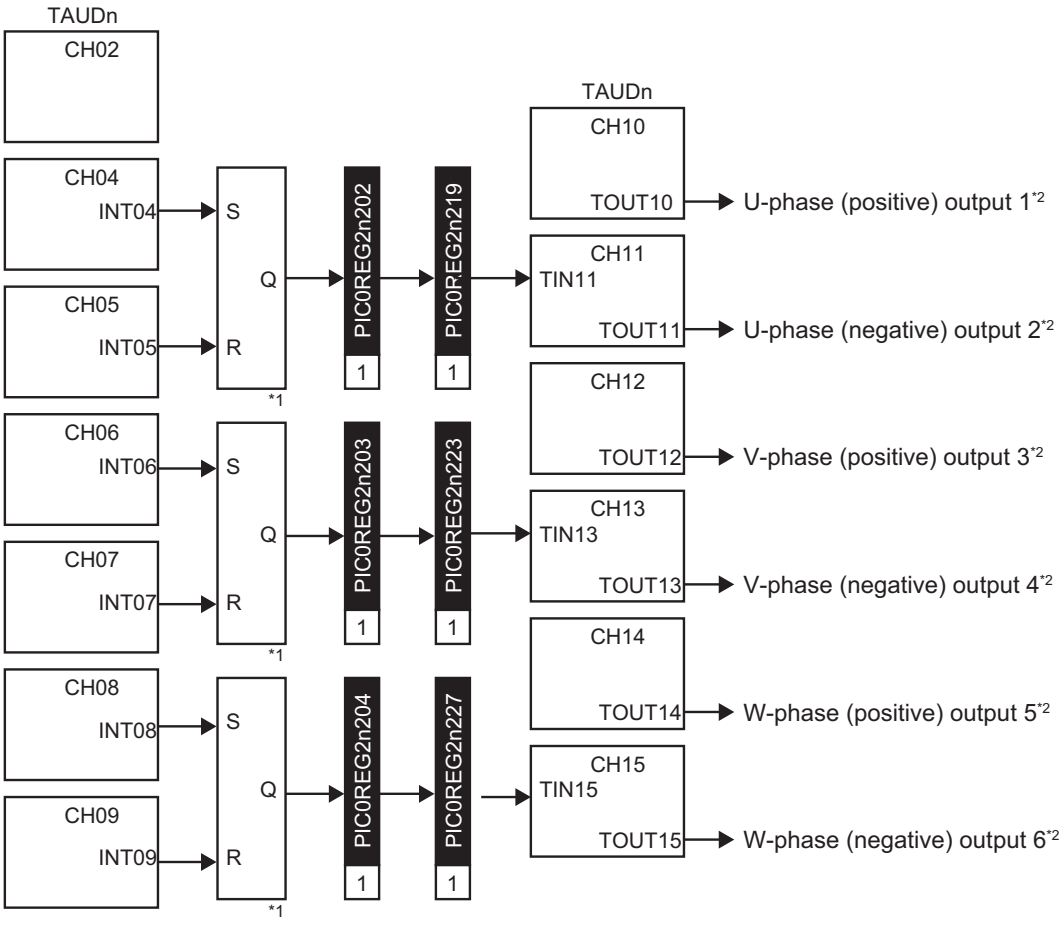
CH	Function Name	M/S	CDR Setting	Description
2	PWM output (CH2 is the master channel for CH4 to CH9.)	M	Cycle	
4		S	Duty (U-phase signal setting)	
5		S	Duty (U-phase signal clearing)	
6		S	Duty (V-phase signal setting)	
7		S	Duty (V-phase signal clearing)	
8		S	Duty (W-phase signal setting)	
9		S	Duty (W-phase signal clearing)	
10	Any feature that does not use TOUT10	S		TOUT10: U-phase output
11	One-phase PWM output	S	Dead time (U phase)	TOUT11: UB-phase output
12	Any feature that does not use TOUT12	S		TOUT12: V-phase output

Table 27.42 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 feature that does 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

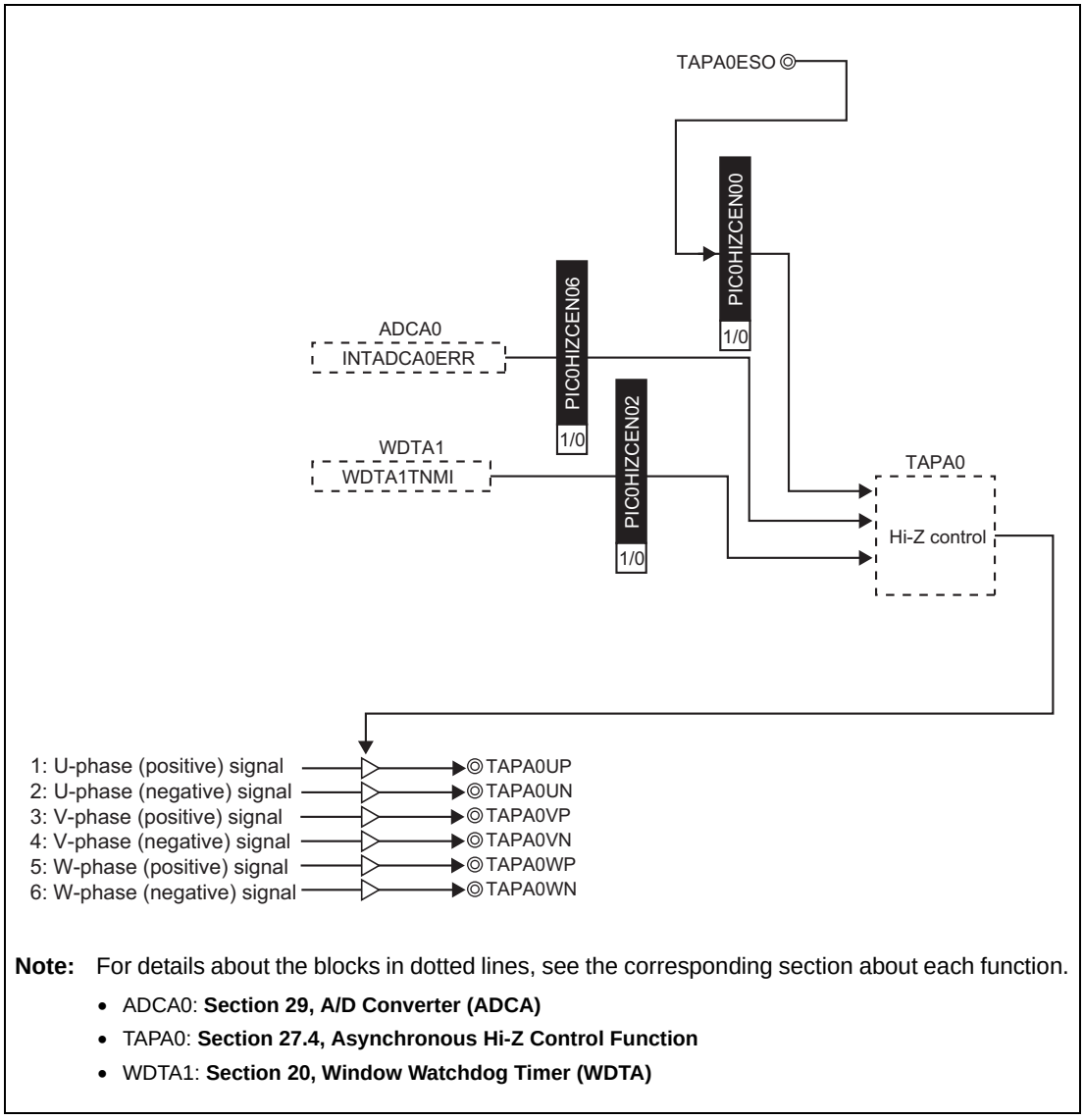
(2) Block diagram



Note 1. SR flip-flop circuit

Note 2. For the connection destination, see **Figure 27.26, Block Diagram: Motor Output Buffer Control**.

Figure 27.25 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 29, A/D Converter (ADCA)**
- TAPA0: **Section 27.4, Asynchronous Hi-Z Control Function**
- WDTA1: **Section 20, Window Watchdog Timer (WDTA)**

Figure 27.26 Block Diagram: Motor Output Buffer Control

27.11.3 Registers

27.11.3.1 PIC0REG2n2 — Timer I/O Control Register 2n2

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
	0	0	0	0	0	0	0	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 27.43 PIC0REG2n2 Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	*1
27	PIC0REG2n227	Select the signal input to TAUDnTTIN15. **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 signal input to TAUDnTTIN13. **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 signal input to TAUDnTTIN11. **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 TIN input signal to TAUDnTTIN15. 0: Setting prohibited 1: Select the set/clear output according to INTTAUDnI8 and INTTAUDnI9.
3	PIC0REG2n203	Select the TIN input signal to TAUDnTTIN13. 0: Setting prohibited 1: Select the set/clear output according to INTTAUDnI6 and INTTAUDnI7.
2	PIC0REG2n202	Select the TIN input signal to TAUDnTTIN11. 0: Setting prohibited 1: Select the set/clear output according to INTTAUDnI4 and INTTAUDnI5.
1, 0	Reserved	*1

Note 1. Some of the bits defined as 0 in the PIC0REG2n2 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

27.11.3.2 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 27.44 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, 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 to these bits, write the value after reset.
2	PIC0HIZCENn2	Select whether to enable or disable Hi-Z output control by the WDTA1TNMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing to this bit, 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

27.11.4 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 peripheral interconnections are used to create the PWM waveform supplied from the set and clear signals generated during PWM output to the input TINm signal (m = 11, 13, or 15) of one-phase PWM output:

- 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 between phases is the assigned channel, only one phase (the U phase) is described below.

27.11.4.1 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.

27.11.4.2 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 23, Timer Array Unit D (TAUD)**.

27.11.4.3 SR Flip-Flop Circuit

The PWM waveform supplied to the input TIN11 signal 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.

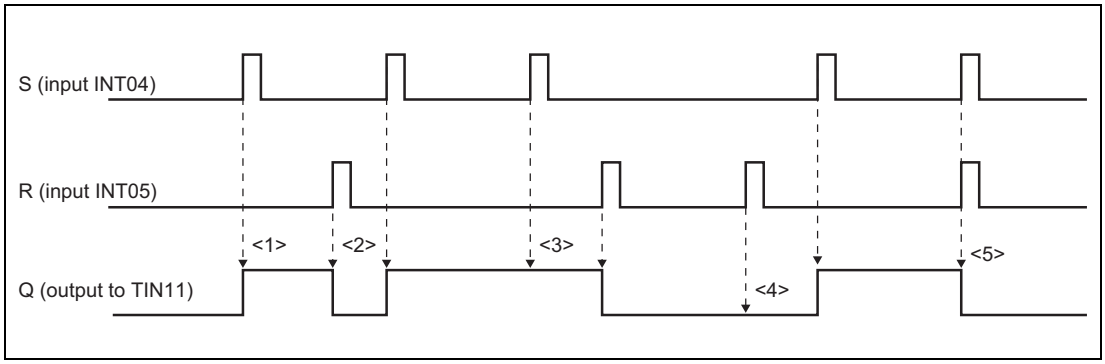


Figure 27.27 SR Flip-Flop Circuit Operation Timing Chart (U-phase example)

(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 is prioritized 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.

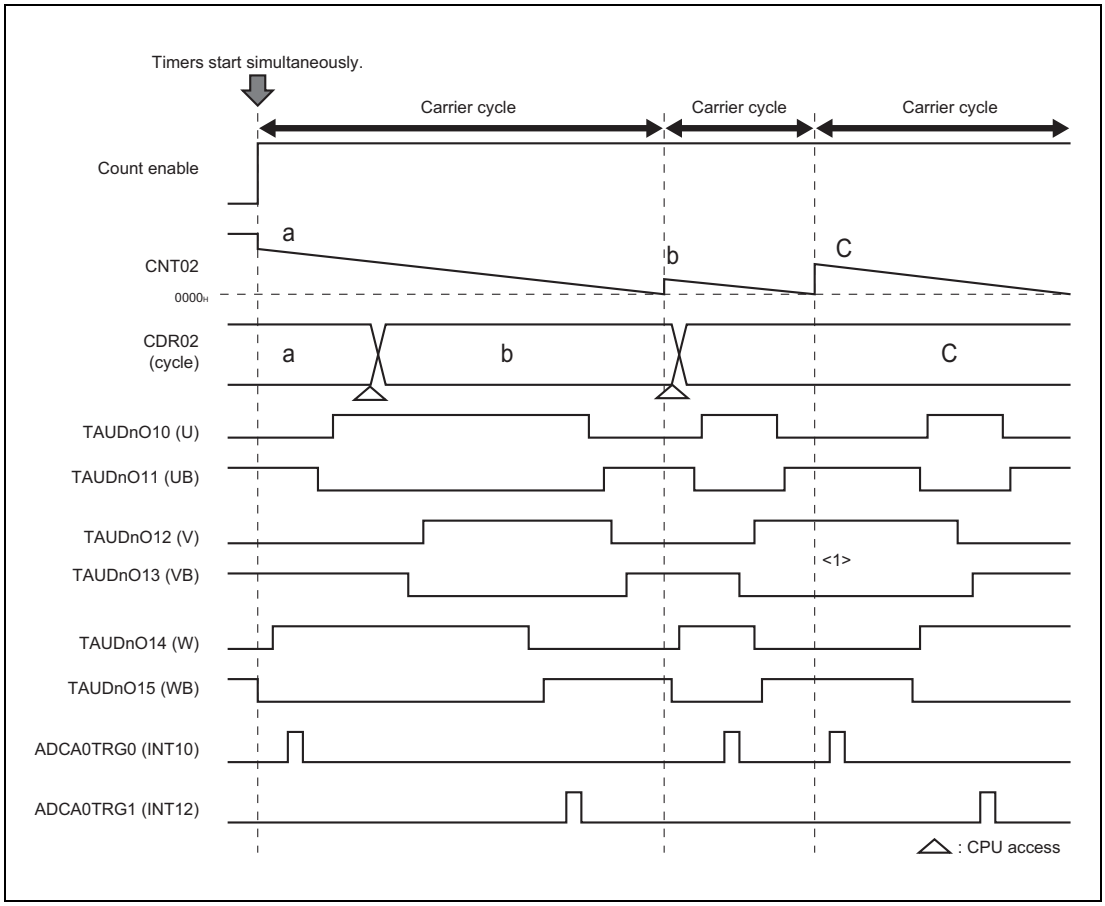


Figure 27.28 Example of Three-Phase PWM (U/UB, V/VB, W/WB) Output with Dead Time

Figure 27.28 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.

In this way, because only the TOUTm signal that performs signal output for the channel performing positive phase output is used during one-phase PWM output, any feature that uses CNTm, CDRm, or INTm can be specified. For details, see **Section 23, 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.

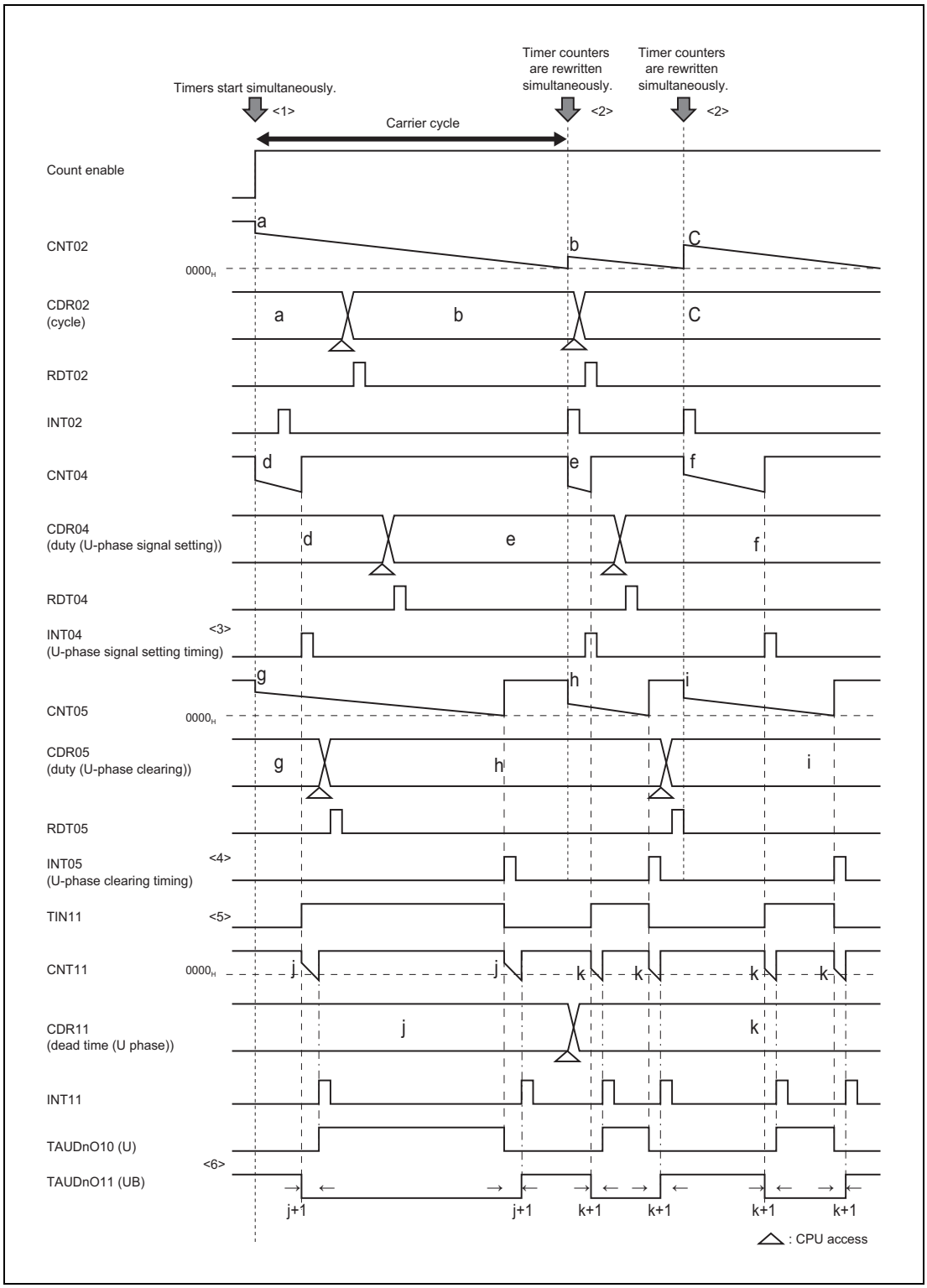


Figure 27.29 Example of One-Phase PWM (U phase, UB phase) Output with Dead Time

An operation example of the timer configuration for performing the U-phase PWM output in **Figure 27.29** 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 timer is also enabled, 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 output by detecting a TIN11 edge.

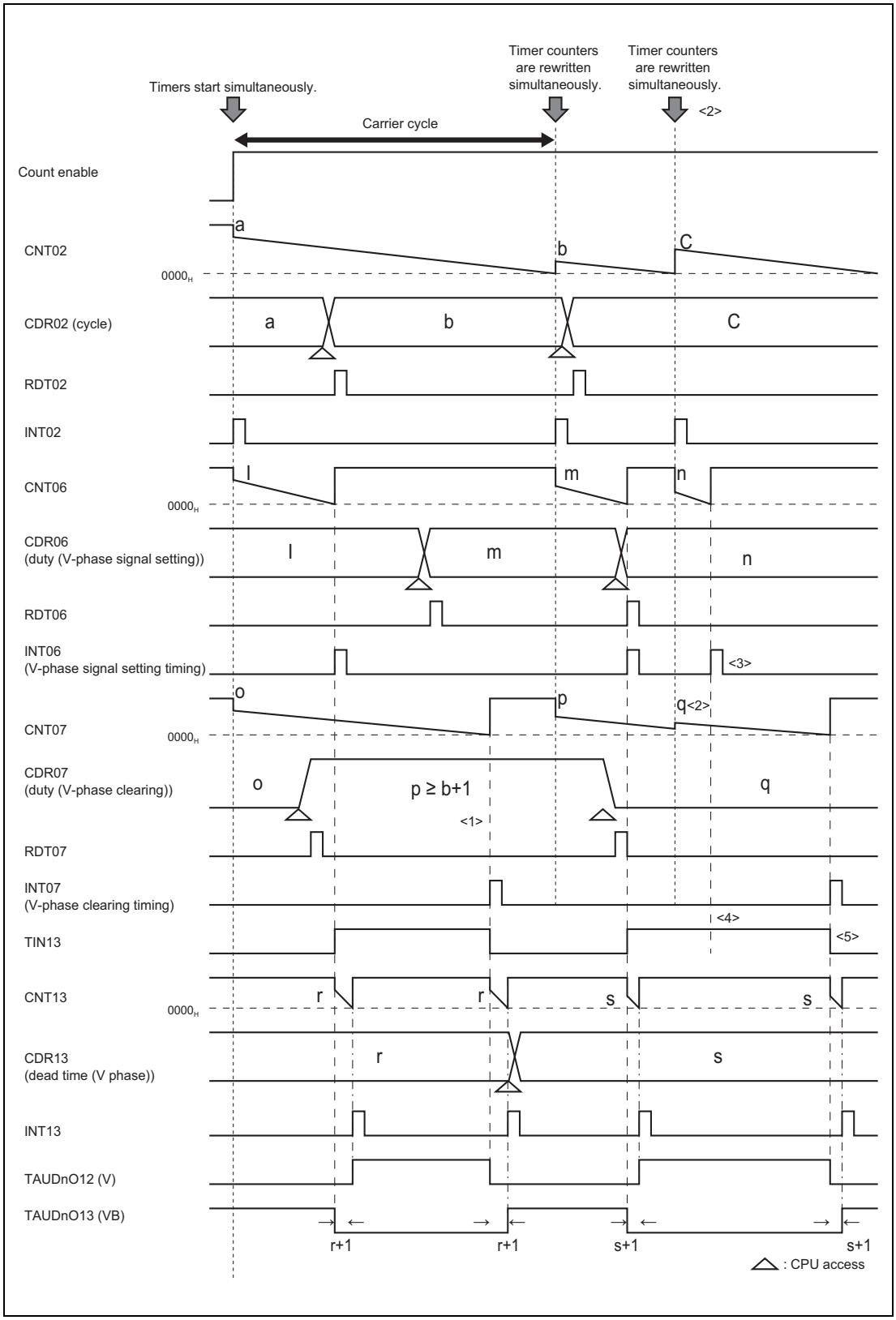


Figure 27.30 Example of One-Phase PWM (V phase, VB phase) Output with Dead Time

An operation example of the timer configuration for performing the V-phase PWM output in **Figure 27.30** 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 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) This results in consecutive V-phase set timing signals (INT06) being generated instead of the V-phase clear timing signal (INT07) that is supposed to be generated.

(4) In this case, because the V-phase set timing signal (INT06) is ignored by the RS 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).

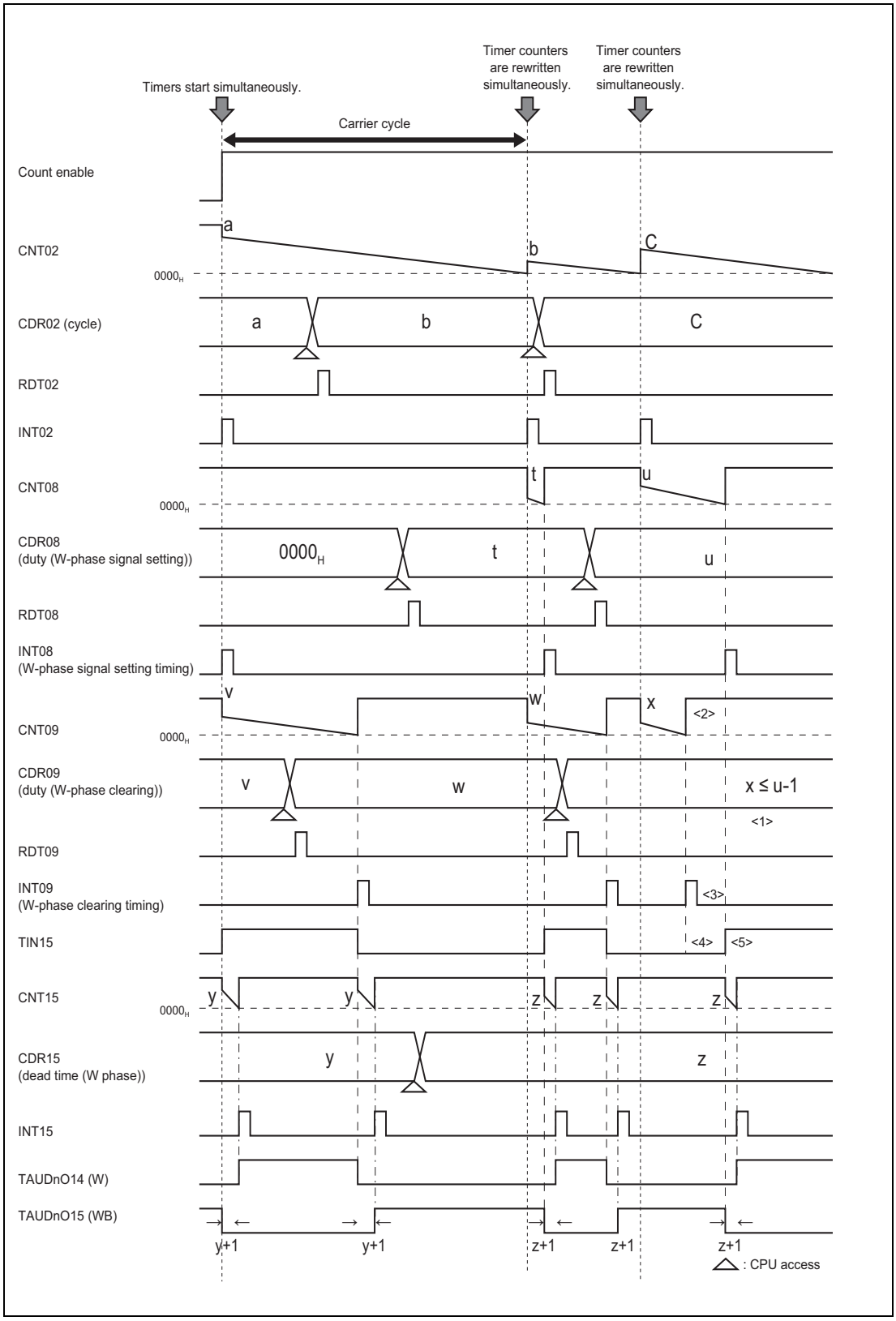


Figure 27.31 Example of One-Phase PWM (W phase, WB phase) Output with Dead Time

An operation example of the timer configuration for performing the W-phase PWM output in **Figure 27.31** 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.

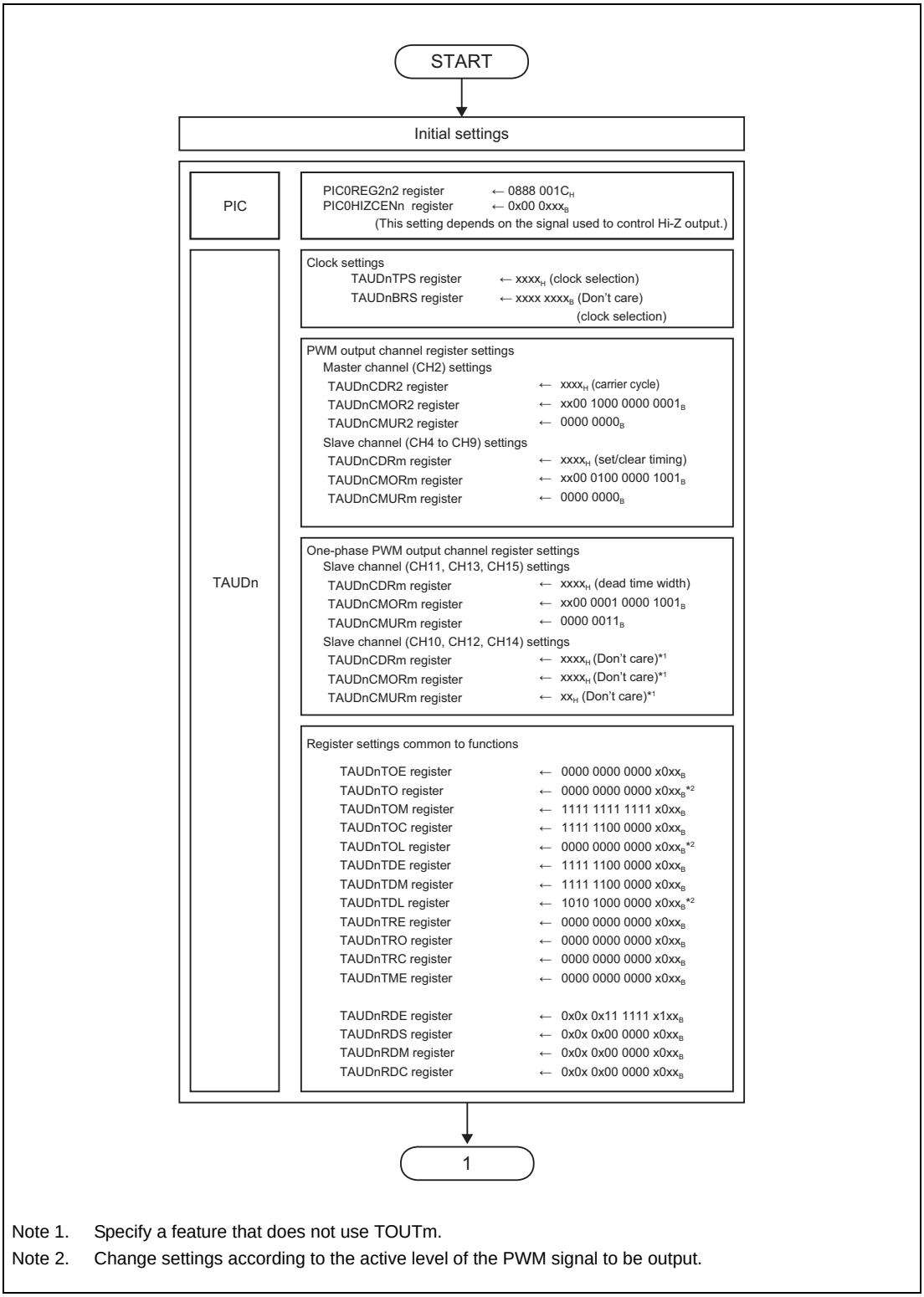
(2) Before a W-phase set timing signal (INT08) is generated by a CH8 underflow, a CH9 (W-phase clear signal output timing timer) underflow occurs, and the W-phase clear timing signal (INT09) is generated.

(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 RS flip-flop circuit, there is no effect on the PWM output waveform.

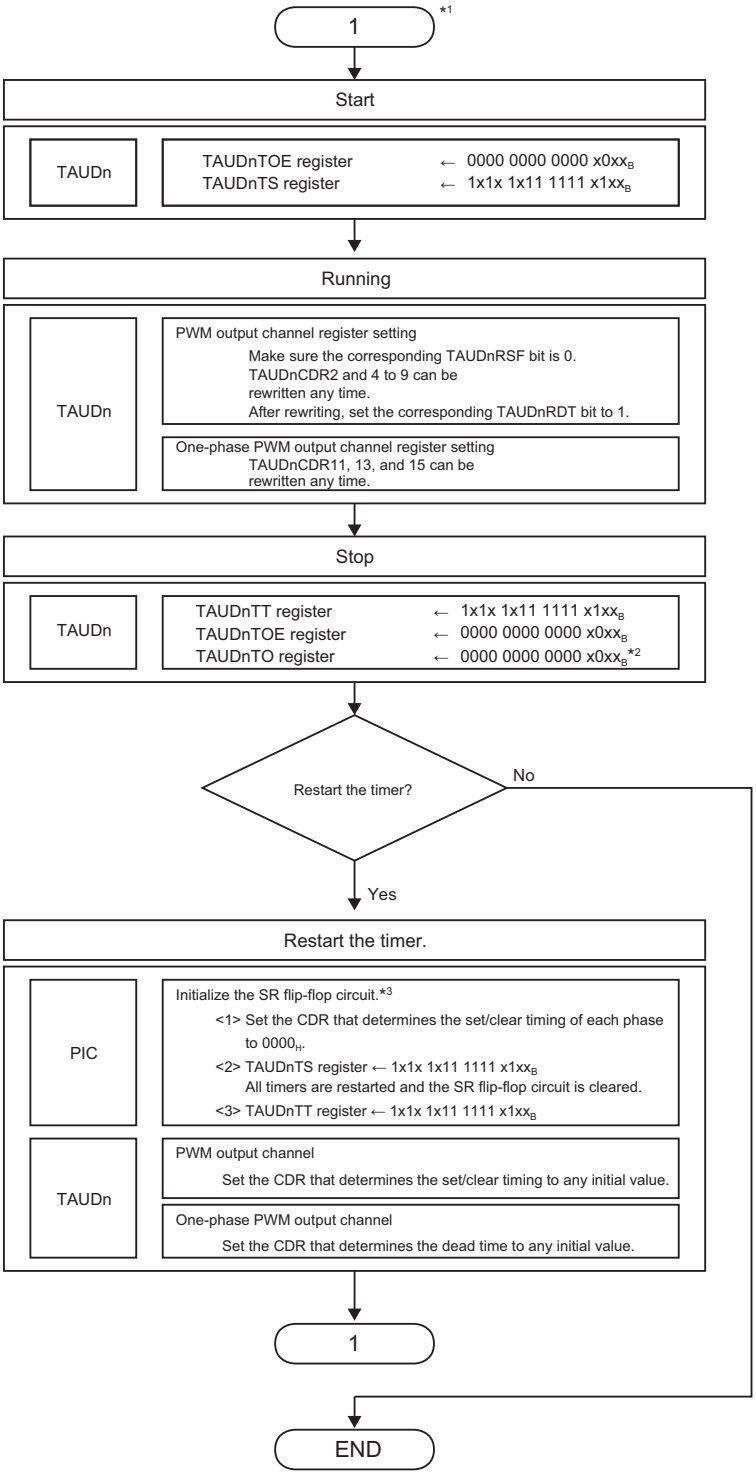
(5) The PWM output is changed at the timing of the next W-phase set timing signal (INT08).

27.11.5 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 27.32 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 27.33 Setup Flow (Active High Example) (continued)

27.11.6 Example of Setting Up Operation Functions

This section provides example settings for each register.

27.11.6.1 TAUDn Settings (Active High Example)

Table 27.45 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 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 27.46 TAUDn: CH4 to CH9-related (PWM Output Slave Channel*1) (m = 4 to 9)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnCMOR2	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	
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 23, 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 used during 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 23, Timer Array Unit D (TAUD)**.

Table 27.47 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 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. 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 23, Timer Array Unit D (TAUD)**.

Table 27.48 Common TAUDn Channel Settings (1/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTOE	15 to 10	TAUDnTOE15 to TAUDnTOE10	0 1	Disable the timer. Enable 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	Output a low-level signal to TOUT15 to TOUT10.
	9 to 4	TAUDnTO09 to TAUDnTO04	0	Output a low-level signal to TOUT09 to TOUT04.
	3	TAUDnTO03	Don't care	
	2	TAUDnTO02	0	Output 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 27.48 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[a]	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	Enable dead time control.*2
	9 to 4	TAUDnTDE09 to TAUDnTDE04	0	Disable dead time control.
	3	TAUDnTDE03	Don't care	
	2	TAUDnTDE02	0	Disable dead time control.
	1, 0	TAUDnTDE01 TAUDnTDE00	Don't care	
TAUDnTDM	15 to 10	TAUDnTDM15 to TAUDnTDM10	1	Output dead time upon detecting a TINm input edge at a lower odd channel.
	9 to 4	TAUDnTDM09 to TAUDnTDM04	0	Invalid because dead time control is disabled.
	3	TAUDnTDM03	Don't care	
	2	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 positive 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 positive 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 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 27.48 Common TAUDn Channel Settings (3/4)

Register	Bit Position	Bit Name	Setting	Remark
TAUDnTRE	15 to 4	TAUDnTRE15 to TAUDnTRE04	0	Stop real-time output.
	3	TAUDnTRE03	Don't care	
	2	TAUDnTRE02	0	Stop 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	
TAUDnTME	15 to 4	TAUDnTME15 to TAUDnTME04	0	Disable modulation output for timer output and real-time output.
	3	TAUDnTME03	Don't Care	
	2	TAUDnTME02	0	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don't Care	
TAUDnRDE	15	TAUDnRDE15	0	Disable simultaneous rewriting.
	14	TAUDnRDE14	Don't Care	
	13	TAUDnRDE13	0	Disable simultaneous rewriting.
	12	TAUDnRDE12	Don't Care	
	11	TAUDnRDE11	0	Disable simultaneous rewriting.
	10	TAUDnRDE10	Don't Care	
	9 to 4	TAUDnRDE09 to TAUDnRDE04	1	Enable simultaneous rewriting.
	3	TAUDnRDE03	Don't Care	
	2	TAUDnRDE02	1	Enable simultaneous rewriting.
	1, 0	TAUDnRDE01 TAUDnRDE00	Don't Care	

Table 27.48 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	Enable simultaneous rewriting by using a master channel.
	3	TAUDnRDS03	Don't Care	
	2	TAUDnRDS02	0	Enable 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	Load the signal when the master channel starts counting.
	3	TAUDnRDM03	Don't Care	
	2	TAUDnRDM02	0	Load 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 23, Timer Array Unit D (TAUD)**.

27.11.6.2 PIC Settings

Table 27.49 PIC Settings

Register	Bit Position	Bit Name	Setting	Remark
PIC0REG2n2	27	PIC0REG2n227	1	Select the input selected by the PIC0REG2n204 bit.
	23	PIC0REG2n223	1	Select the input selected by the PIC0REG2n203 bit.
	19	PIC0REG2n219	1	Select the input selected by the PIC0REG2n202 bit.
	4	PIC0REG2n204	1	Select the set/clear output according to INTTAUDnI8 and INTTAUDnI9.
	3	PIC0REG2n203	1	Select the set/clear output according to INTTAUDnI6 and INTTAUDnI7.
	2	PIC0REG2n202	1	Select the set/clear output according to INTTAUDnI4 and INTTAUDnI5.

27.12 High-accuracy Triangle PWM Output with Dead Time

27.12.1 Functional Overview

Compared to the triangle PWM output with dead time feature 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.

For 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 27.34**)

For this feature, a pulse is generated in combination with the TAUD timer output, and a pseudo dead time pulse is added.

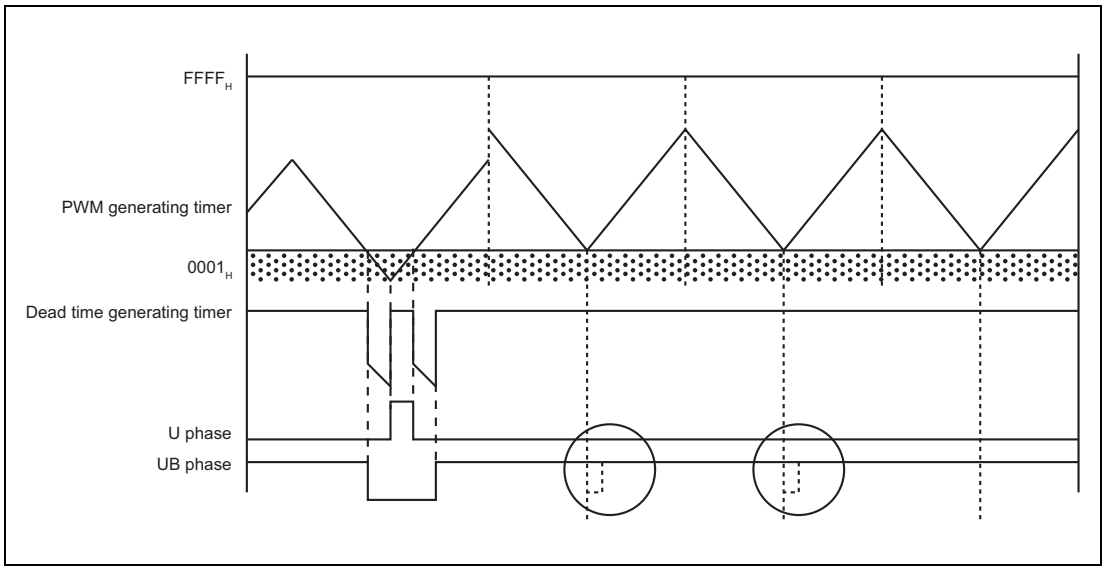


Figure 27.34 Timing of Dead Time Output by the TAUD Feature for Outputting a Triangle PWM Signal with Dead Time

27.12.2 Configuration

The unit and channel configuration for this feature are shown below. (n = 0)

Table 27.50 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)

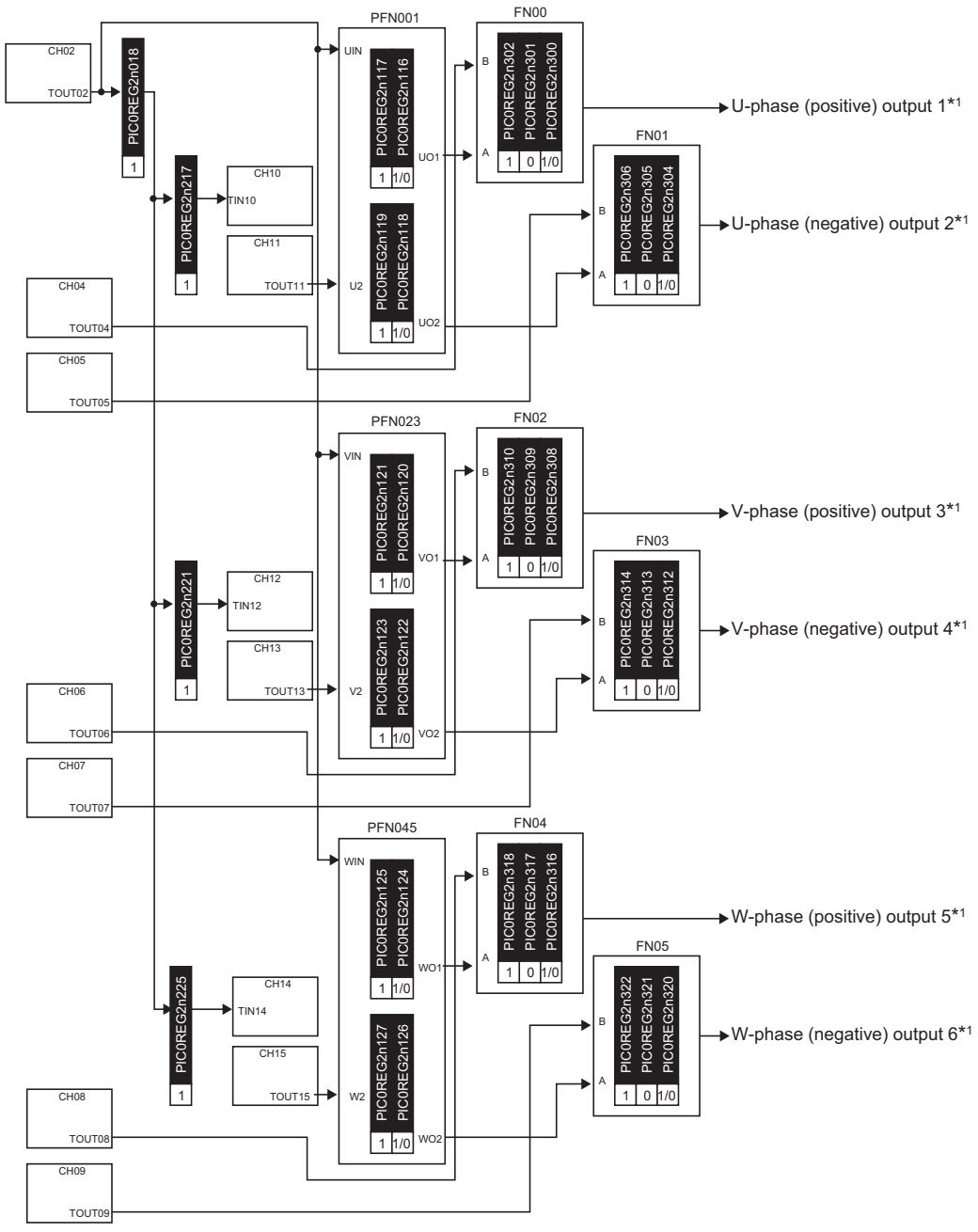
(1) TAUDn configuration

Table 27.51 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 27.26, Block Diagram: Motor Output Buffer Control**.

Figure 27.35 Block Diagram: High-Accuracy Triangle PWM Output with Dead Time

27.12.3 Registers

27.12.3.1 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 27.52 PIC0REG2n0 Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	*1
18	PIC0REG2n018	Select the TIN input signal to TAUDnTTIN10, TAUDnTTIN12, and TAUDnTTIN14. 0: Setting prohibited 1: Select TAUDnTTOUT2.
17 to 0	Reserved	*1

Note 1. Some of the bits defined as 0 in the PIC0REG2n0 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

27.12.3.2 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 27.53 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 27.53 PIC0REG2n1 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>21, 20</td><td>PIC0REG2n121
PIC0REG2n120</td><td>Select the FN02 A input signal according to the output logic specified for CH6 of TAUDn.
<table>
<tr><th>PIC0REG2n121</th><th>PIC0REG2n120</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Combination circuit output
(Select this when the active high setting is specified (TAUDnTOL06 = 0).)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted combination circuit output
(Select this when the active low setting is specified (TAUDnTOL06 = 1).)</td></tr>
<tr><td colspan="2">Other than the above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>19, 18</td><td>PIC0REG2n119
PIC0REG2n118</td><td>Select the FN01 A input signal according to the output logic specified for CH5 of TAUDn.
<table>
<tr><th>PIC0REG2n119</th><th>PIC0REG2n118</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Combination circuit output
(Select this when the active high setting is specified (TAUDnTOL05 = 0).)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted combination circuit output
(Select this when the active low setting is specified (TAUDnTOL05 = 1).)</td></tr>
<tr><td colspan="2">Other than the above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>17, 16</td><td>PIC0REG2n117
PIC0REG2n116</td><td>Select the FN00 A input signal according to the output logic specified for CH4 of TAUDn.
<table>
<tr><th>PIC0REG2n117</th><th>PIC0REG2n116</th><th>Input Signal</th></tr>
<tr><td>1</td><td>0</td><td>Combination circuit output
(Select this when the active high setting is specified (TAUDnTOL04 = 0).)</td></tr>
<tr><td>1</td><td>1</td><td>Inverted combination circuit output
(Select this when the active low setting is specified (TAUDnTOL04 = 1).)</td></tr>
<tr><td colspan="2">Other than the above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>15 to 0</td><td>Reserved</td><td>*1</td></tr>
</table>

Note 1. Some of the bits defined as 0 in the PIC0REG2n1 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

27.12.3.3 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 27.54 PIC0REG2n2 Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	*1
25	PIC0REG2n225	Select the TIN input signal to TAUDnTTIN14.

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 TAUDnTTIN12.

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 TAUDnTTIN10.

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. Some of the bits defined as 0 in the PIC0REG2n2 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

27.12.3.4 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 27.55 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 27.55 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. Some of the bits defined as 0 in the PIC0REG2n3 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

27.12.3.5 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 27.56 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, 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 to these bits, 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 to this bit, 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

27.12.4 Operation Example

This is achieved by combining the following TAUD features:

- Triangle PWM output with dead time
- One-shot pulse output

In addition, the following 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 peripheral interconnections are also used because the pulse to be inserted into the variable dead time area is combined 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 between phases is the assigned channel, only one phase (the U phase) is described below.

27.12.4.1 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.

27.12.4.2 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 23, Timer Array Unit D (TAUD)**.

27.12.4.3 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.

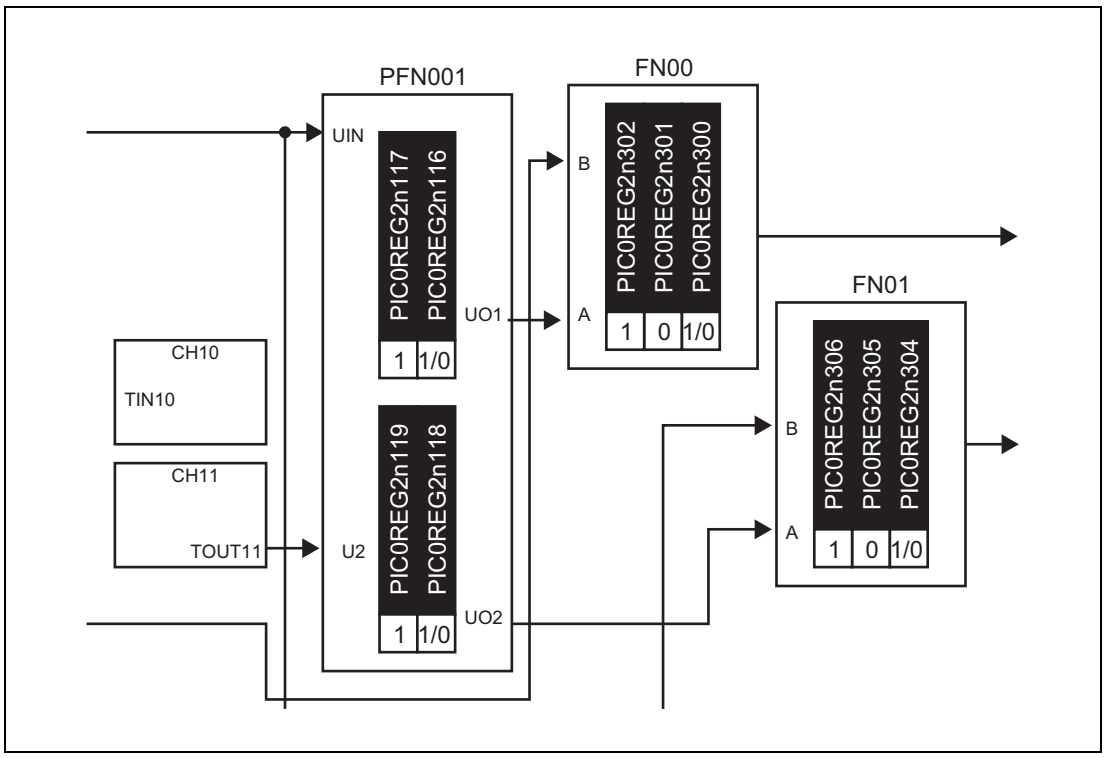


Figure 27.36 Block Diagram Excerpt (PFN001, FN00, and FN01)

The table below shows the relationships between combination circuit input (UIN, U2) and output (UO1, UO2).

Table 27.57 U and UB Phase Combination Circuit (PFN001) I/O Table

- **UO1 (U-phase variable dead time area pulse) output**

		UO1	
UIN (TOUT02)	**U2 (TOUT11)**	**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**

		UO2	
UIN (TOUT02)	**U2 (TOUT11)**	**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.

27.12.4.4 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 27.58 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)	100B	AND of FN00 B (TOUT04) and FN00 A (UO1)
Active low (TAUDnTOL04 = 1)	101B	OR of FN00 B (TOUT04) and FN00 A (UO1)

- **UB-phase output (TOUT05)**

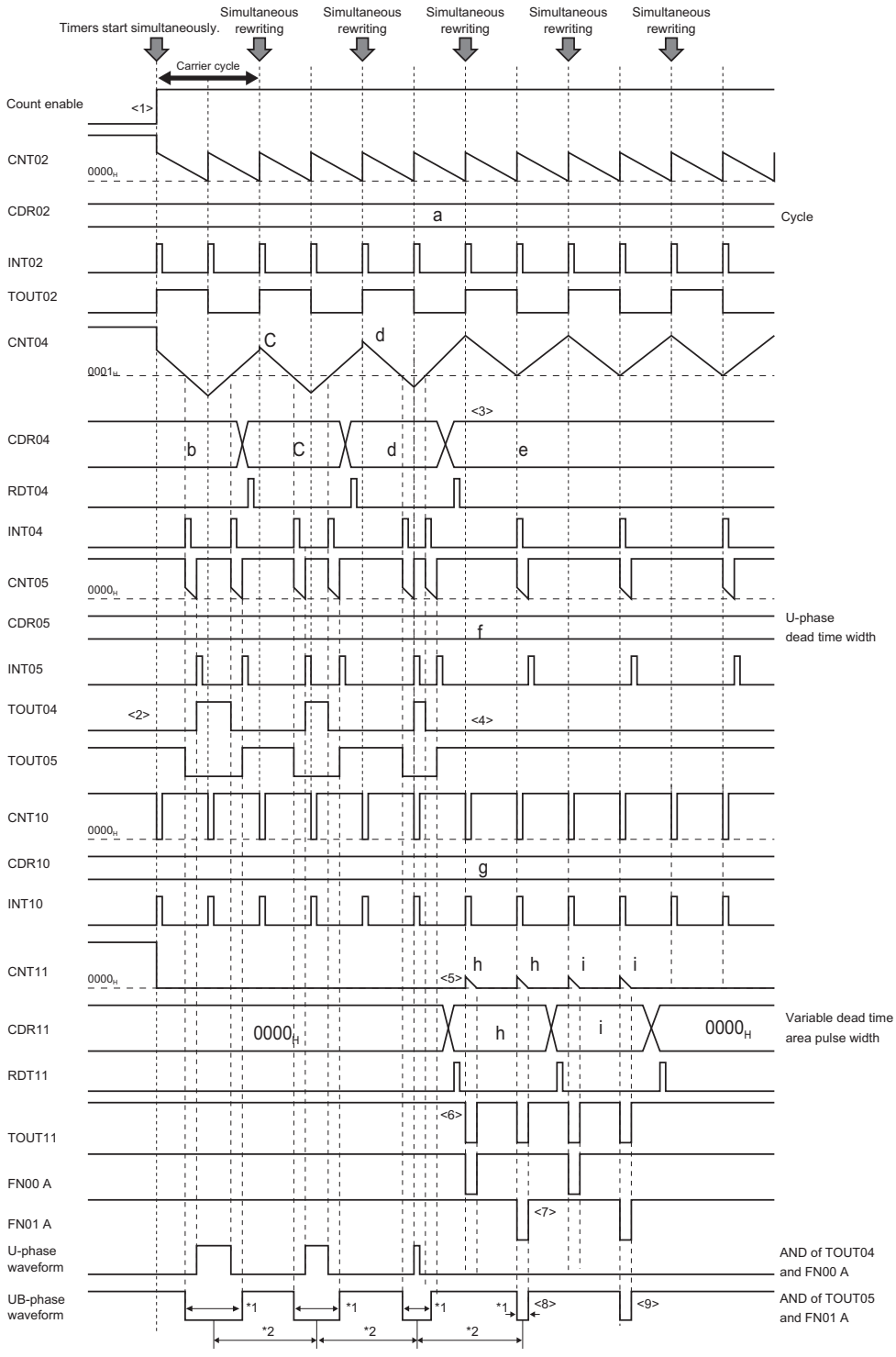
Active level	PIC0REG2n306 to 04	TAPAnUM pin output
Active high (TAUDnTOL05 = 0)	100B	AND of FN01 B (TOUT05) and FN01 A (UO2)
Active low (TAUDnTOL05 = 1)	101B	OR of FN01 B (TOUT05) and FN01 A (UO2)

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 27.35, 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 27.37 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 27.37** 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 set to the inactive level, and TOUT05 output is set to 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.

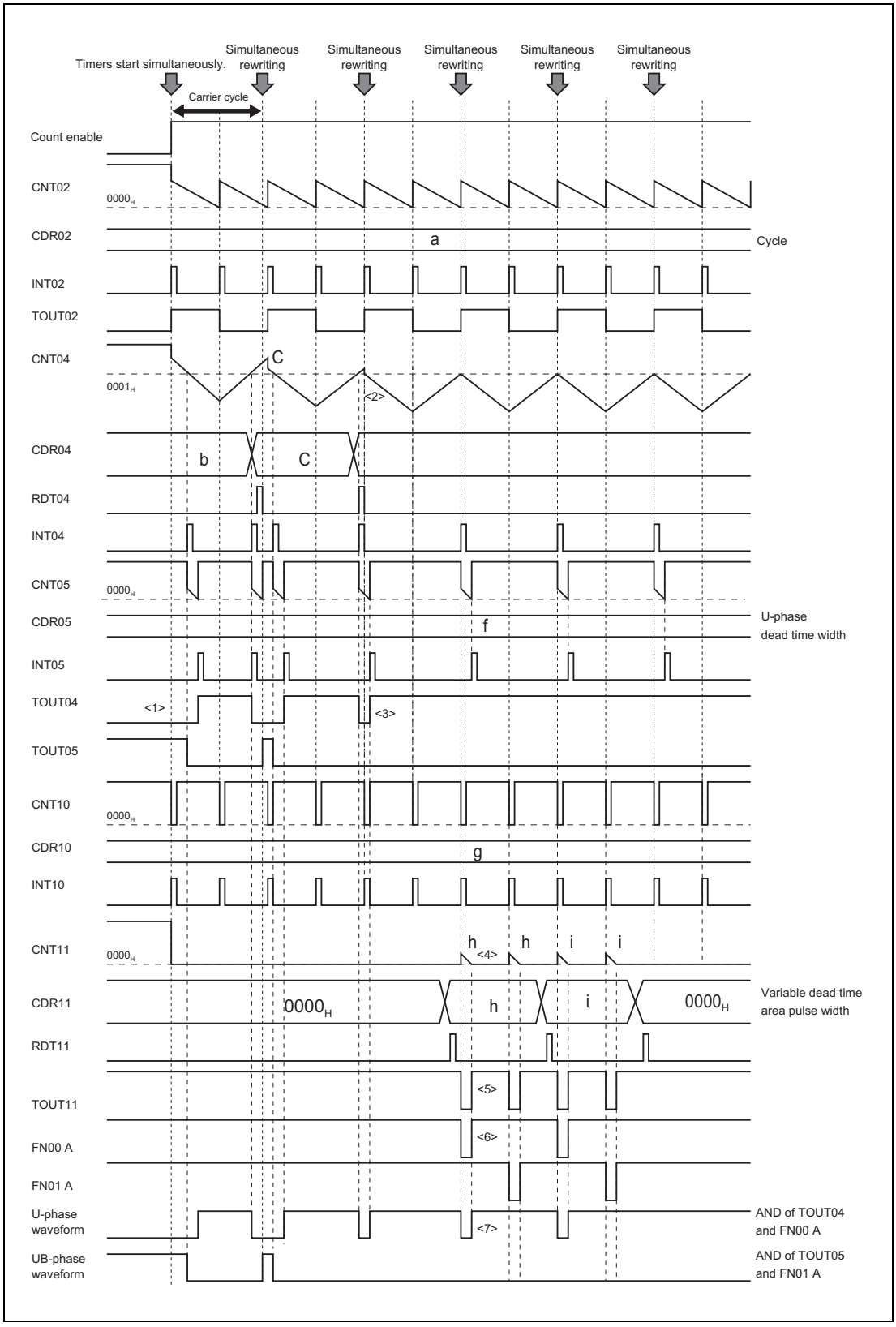


Figure 27.38 Example of a High-Accuracy PWM Signal Output with Dead Time (U-Phase: 100%, UB-Phase: 0%) (when TAUDnTOL04 = 0 (Active High) and TAUDnTOL05 = 0 (Active High))

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 27.38** 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 set to the active level, and TOUT05 output is set to 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 is affected by the amount shown by <2> for the last PWM signal output from TOUT04 and shown by feature specification <1>, as shown in Figure 27.38.
To cancel this effect, the CDR11 setting is delayed one cycle.

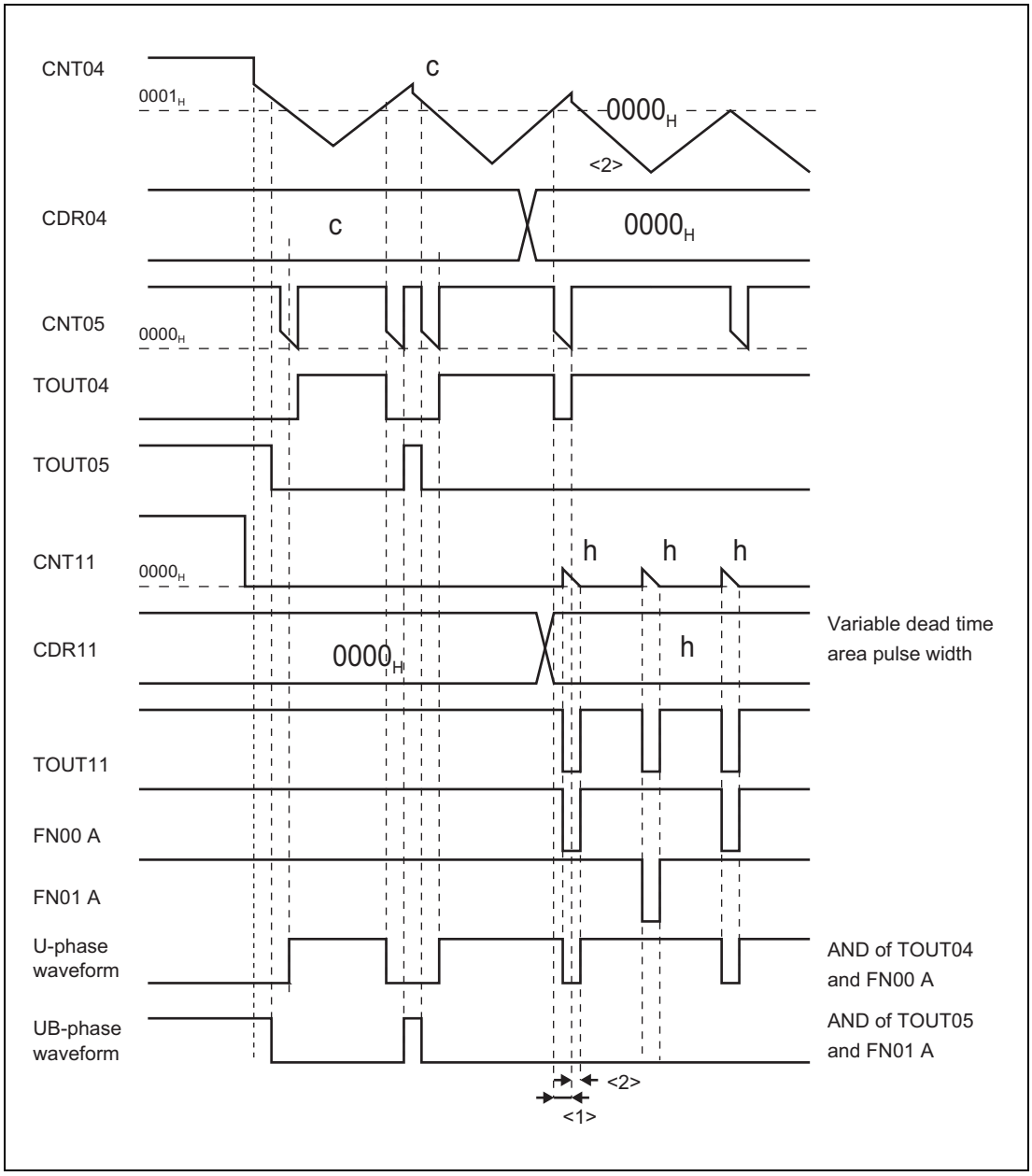


Figure 27.39 Effect on the Output Triangle PWM Wave with Dead Time by the Variable Dead Time Area Pulse

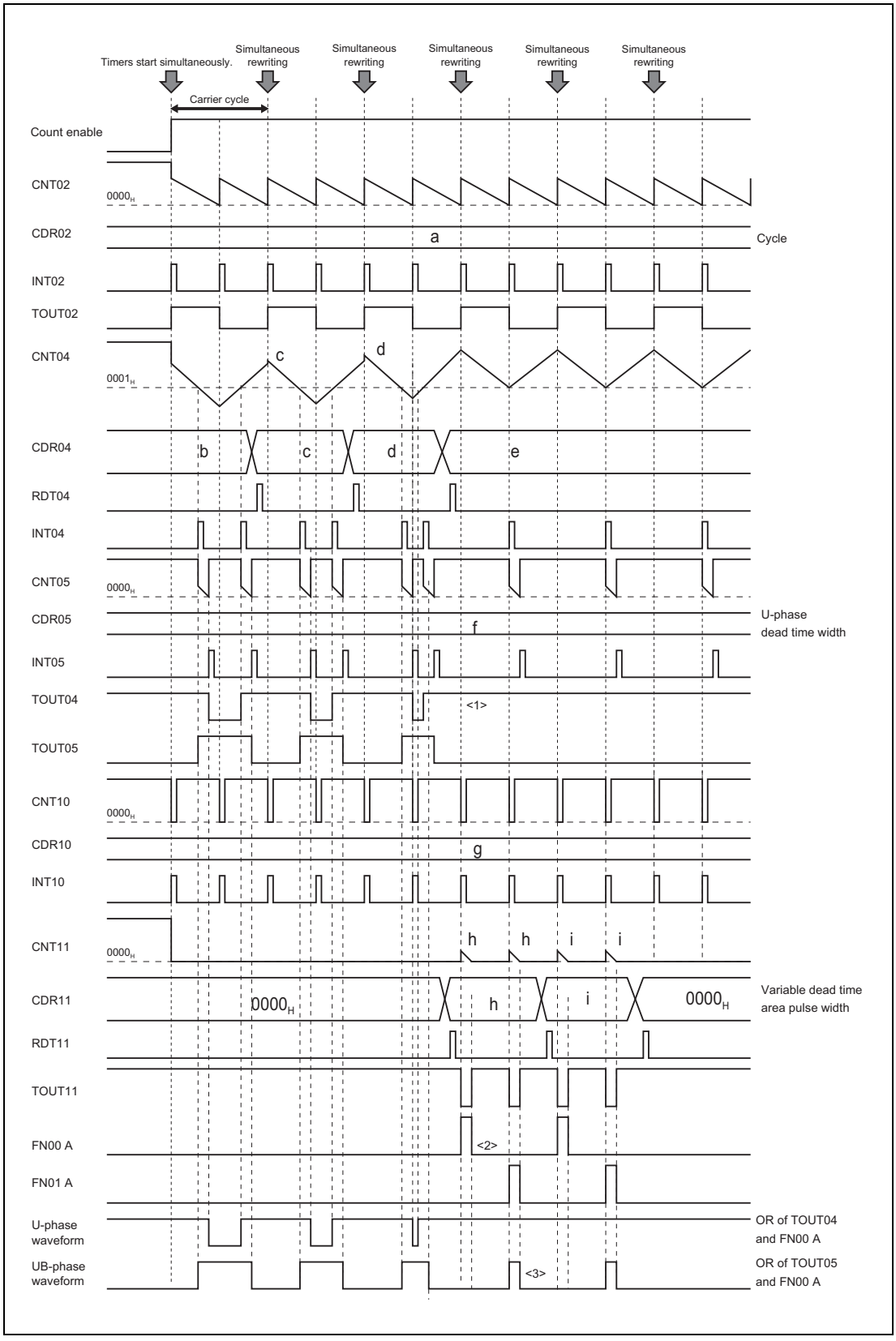


Figure 27.40 Example of a High-Accuracy PWM Signal Output with Dead Time (U-Phase: 100%, UB-Phase: 0%) (TAUDnTOL04 = 1 (Active Low), TAUDnTOL05 = 1 (Active Low))

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 27.40** 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 27.37, 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.

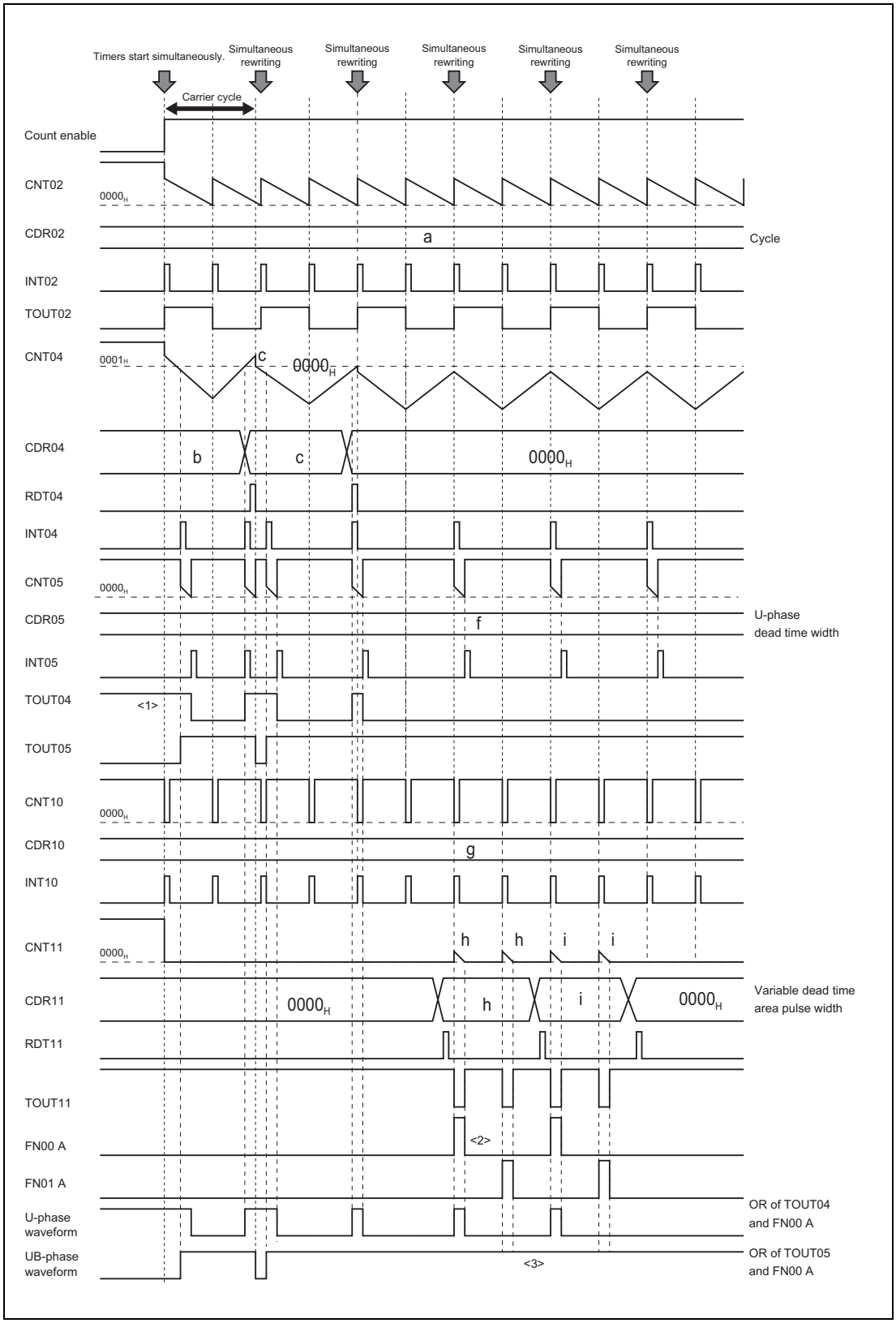


Figure 27.41 Example of a High-Accuracy PWM Signal Output with Dead Time (U-Phase: 0%, UB-Phase: 100%) (when TAUDnTOL04 = 0 (Active Low) and TAUDnTOL05 = 0 (Active Low))

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 27.41** 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 27.38, 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.

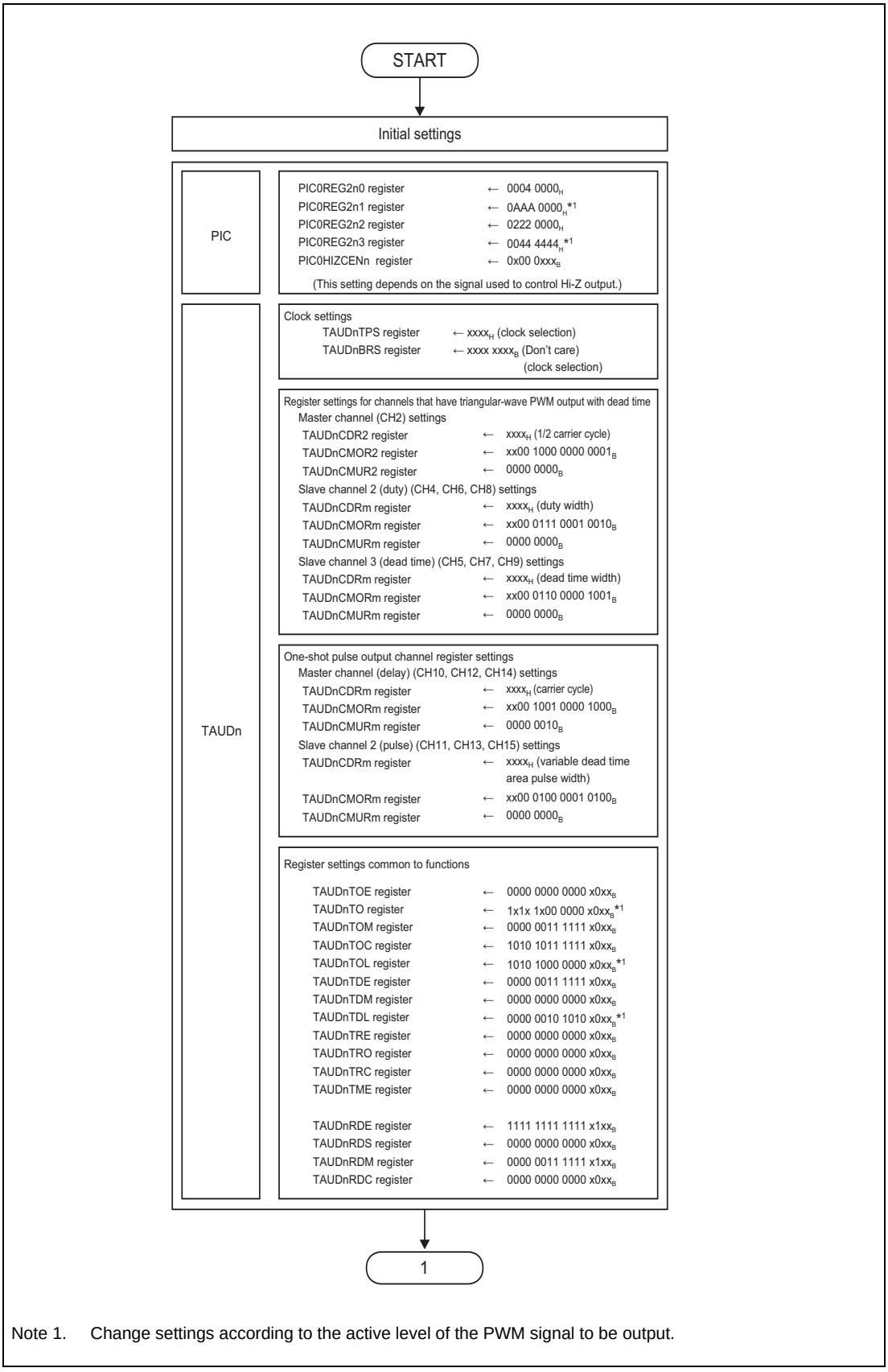
(2) Therefore, active low output that corresponds with PWM output is specified as the combination circuit setting (PIC0REG2n116 and PIC0REG2n117, and PIC0REG2n118 and PIC0REG2n119). This results in the output of an active low variable dead time area pulse for the U phase (FN00 A) and UB phase (FN01 A).

(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

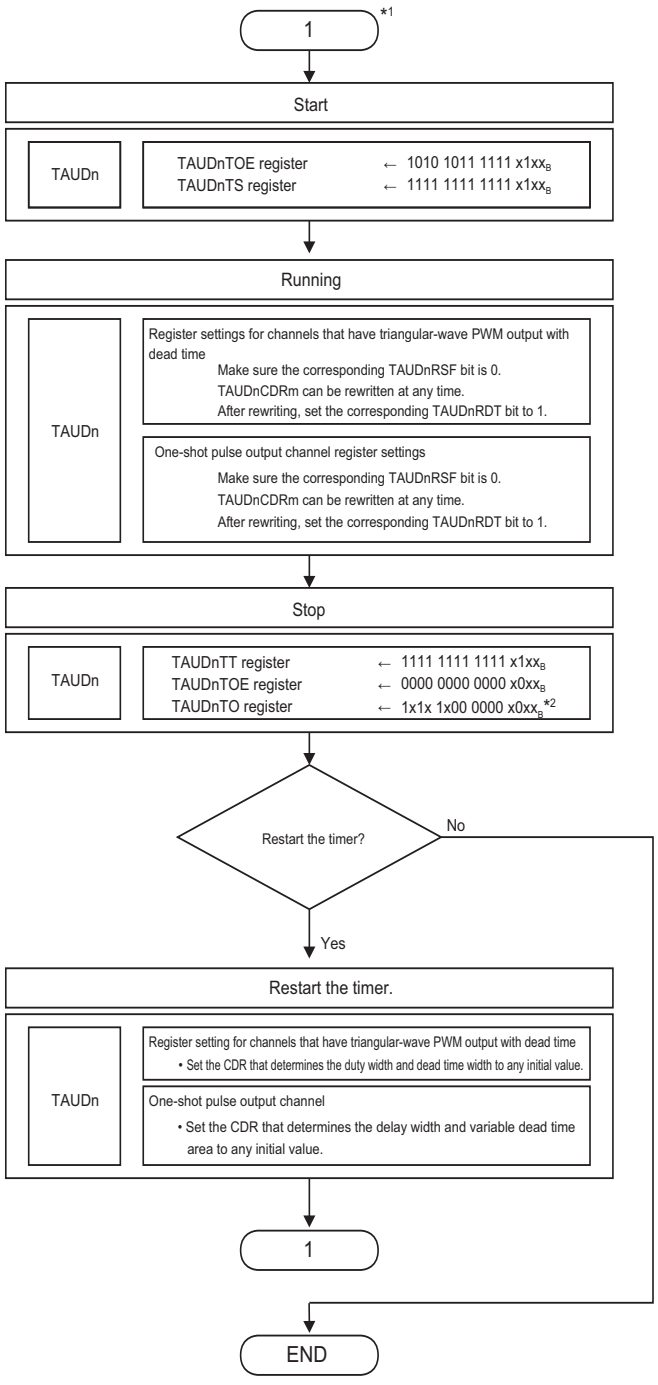
If the 100% U-phase duty setting for CDR04 and the variable dead time area pulse width for CDR11 are specified at the same time, the last PWM signal output from TOUT04 is adversely affected due to the feature specifications.
To cancel this effect, the CDR11 setting is delayed one cycle.
For details, see Figure 27.39, Effect on the Output Triangle PWM Wave with Dead Time by the Variable Dead Time Area Pulse.

27.12.5 Setup Flow



Note 1. Change settings according to the active level of the PWM signal to be output.

Figure 27.42 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 27.43 Setup Flow (Active High Example) (continued)

27.12.6 Example of Setting Up Operation Functions

This section provides example settings for each register.

27.12.6.1 TAUDn settings (active high example)

Table 27.59 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 23, 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 23, Timer Array Unit D (TAUD)**.
For this feature, set TAUDnMD0 to 1.

Table 27.60 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 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 27.61 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 23, 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 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 23, Timer Array Unit D (TAUD)**.

Table 27.62 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	Disable start triggers during counting.
TAUDnCMURm	1, 0	TAUDnTIS[1:0]	10	Detect 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 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 27.63 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	Disable 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 23, 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 23, Timer Array Unit D (TAUD)**.
For this feature clear TAUDnMD0 to 0.

Table 27.64 Common TAUDn Channel Settings (1/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTOE	15	TAUDnTOE15	0 1	Disable the timer. Enable the timer.
	14	TAUDnTOE14	0	
	13	TAUDnTOE13	0 1	Disable the timer. Enable the timer.
	12	TAUDnTOE12	0	
	11	TAUDnTOE11	0 1	Disable the timer. Enable the timer.
	10	TAUDnTOE10	0	
	9 to 4	TAUDnTOE09 to TAUDnTOE04	0 1	Disable the timer. Enable the timer.
	3	TAUDnTOE03	Don't care	
	2	TAUDnTOE02	0 1	Disable the timer. Enable the timer.
	1, 0	TAUDnTOE01 TAUDnTOE00	Don't care	

Table 27.64 Common TAUDn Channel Settings (2/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTO	15	TAUDnTO15	1*[1]	Output a high-level signal to TOUT15.
	14	TAUDnTO14	Don't Care	
	13	TAUDnTO13	1*[1]	Output a high-level signal to TOUT13.
	12	TAUDnTO12	Don't Care	
	11	TAUDnTO11	1*[1]	Output a high-level signal to TOUT11.
	10	TAUDnTO10	Don't Care	
	9 to 4	TAUDnTO09 to TAUDnTO04	0*[1]	Output a low-level signal to TOUT09 to TOUT04.
	3	TAUDnTO03	Don't Care	
	2	TAUDnTO02	0	Output 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 27.64 Common TAUDn Channel Settings (3/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTDE	15 to 10	TAUDnTDE15 to TAUDnTDE10	0	Disable dead time control.
	9 to 4	TAUDnTDE09 to TAUDnTDE04	1	Enable dead time control.*2
	3	TAUDnTDE03	Don't Care	
	2	TAUDnTDE02	0	Disable 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	Disable real-time output.
	3	TAUDnTRE03	Don't Care	
	2	TAUDnTER02	0	Disable 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 27.64 Common TAUDn Channel Settings (4/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTME	15 to 4	TAUDnTME15 to TAUDnTME04	0	Disable modulation output for timer output and real-time output.
	3	TAUDnTME03	Don't Care	
	2	TAUDnTME02	0	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don't Care	
TAUDnRDE	15 to 4	TAUDnRDE15 to TAUDnRDE04	1	Enable simultaneous rewriting.
	3	TAUDnRDE03	Don't Care	
	2	TAUDnRDE02	1	Enable 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	Perform simultaneous rewriting when the master channel starts counting.
	9 to 4	TAUDnRDM09 to TAUDnRDM04	1	Perform 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	Perform 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 23, Timer Array Unit D (TAUD)**.

27.12.6.2 PIC Settings (Active High Example)

Table 27.65 PIC Settings

Register	Bit position	Bit name	Setting	Remark
PIC0REG2n0	18	PIC0REG2n018	1	Select 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	Select the input selected by the PIC0REG2n018 bit.
	21	PIC0REG2n221	1	Select the input selected by the PIC0REG2n018 bit.
	17	PIC0REG2n217	1	Select 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

27.13 Delay Pulse Output with Dead Time

27.13.1 Functional Overview

This feature outputs a three-phase PWM signal with dead time that is later than the cycle timing by an amount equal to the delay amount.

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.

27.13.2 Configuration

The unit and channel configuration for this feature are shown below. (n = 0, m = 0 to 15)

Table 27.66 Configuration of Delay Pulse Output with Dead Time

Timer	Timer motor control function
TAUD0 CH2 to CH15 (used channels fixed)	TAPA0

NOTE

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)

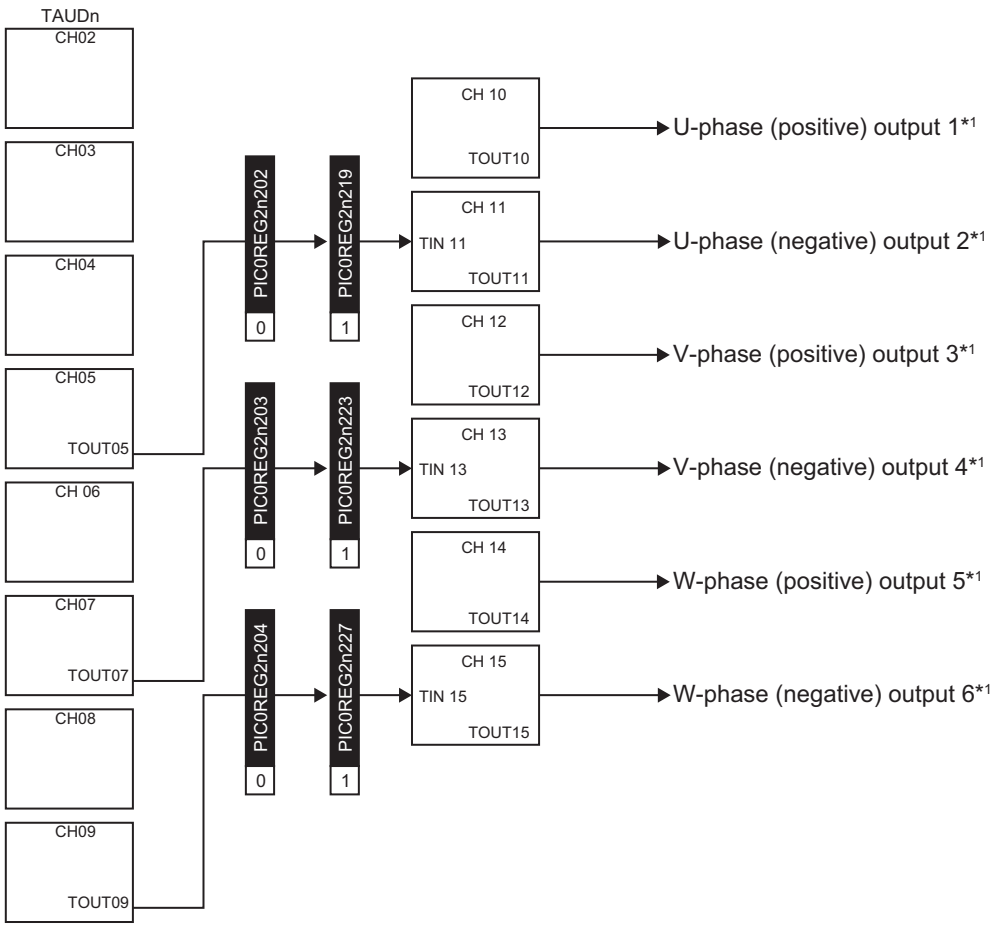
27.13.2.1 TAUDn configuration

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 27.67 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 27.26, Block Diagram: Motor Output Buffer Control**.

Figure 27.44 Block Diagram: Delay Pulse Output with Dead Time

27.13.3 Registers

27.13.3.1 PIC0REG2n2 — Timer I/O Control Register 2n2 (n = 0)

This register selects CHm input signals of the TAUDn timer. 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 27.68 PIC0REG2n2 Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	*1
27	PIC0REG2n227	Select the TIN input signal to TAUDnTTIN15. **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 TAUDnTTIN13. **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 TAUDnTTIN11. **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 TAUDnTTIN15. 0: Select TAUDnTTOUT9. 1: Setting prohibited
3	PIC0REG2n203	Select the signal supplied to TAUDnTTIN13. 0: Select TAUDnTTOUT7. 1: Setting prohibited
2	PIC0REG2n202	Select the signal supplied to TAUDnTTIN11. 0: Select TAUDnTTOUT5. 1: Setting prohibited
1, 0	Reserved	*1

Note 1. Some of the bits defined as 0 in the PIC0REG2n2 register are defined for the other timer connection functions. For such bits, use the bit definition of those timer connection functions.

27.13.3.2 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 27.69 PIC0HIZCENn Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, 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 to these bits, write the value after reset.
2	PIC0HIZCENn2	Select whether to enable or disable Hi-Z output control by the WDTA0TNMI interrupt signal. 0: Disable 1: Enable
1	Reserved	When read, the value after reset is returned. When writing to this bit, 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

27.13.4 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.

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.

27.13.4.1 Delay pulse output function

A basic PWM signal for the one-phase PWM output delayed by the amount specified on CH4 for the cycle specified by TOUT05 on CH2 is output by using CH2, CH4, and CH5 in combination.

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.

27.13.4.2 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 23, 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.

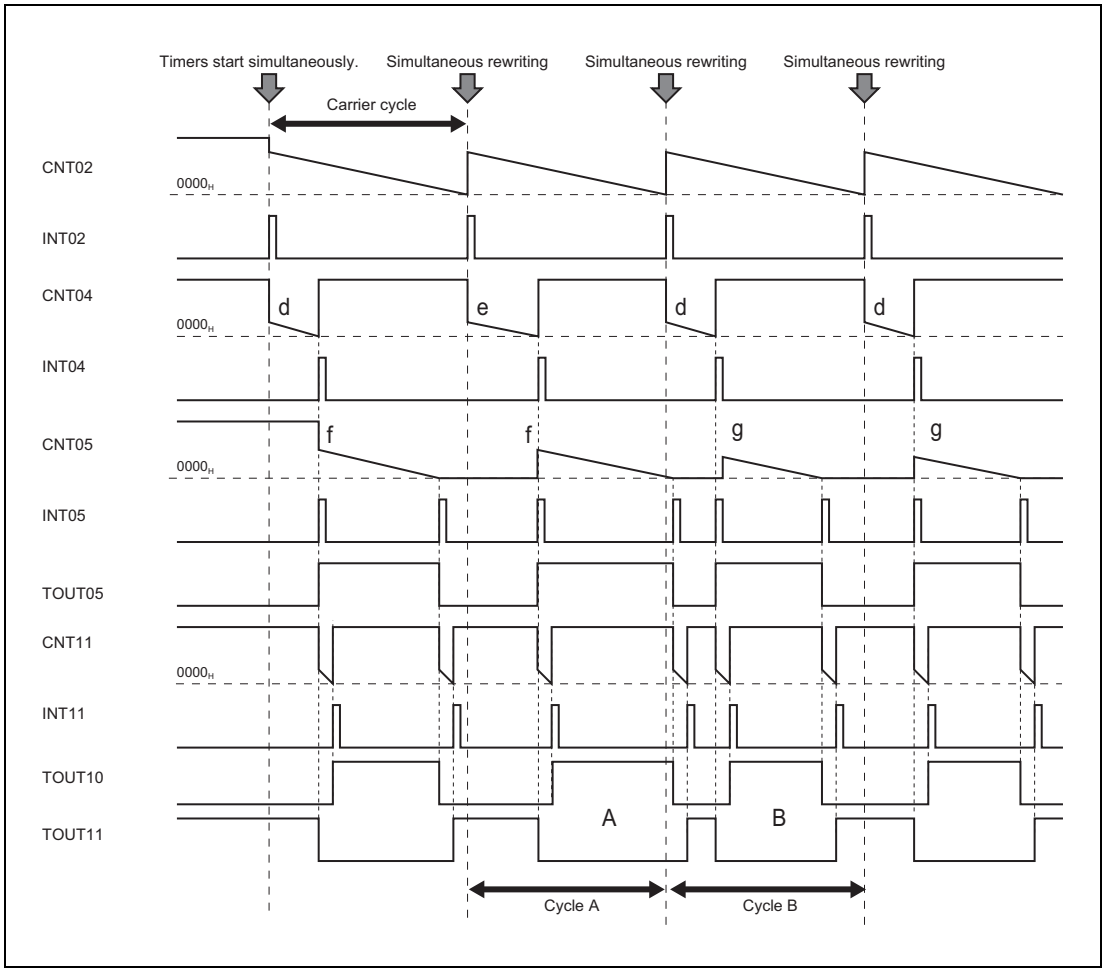


Figure 27.45 PWM Output by Outputting a Delay Pulse with Dead Time

In **Figure 27.45**, PWM waveform A is supposed to be output before cycle A ends, but, because the delay timing is too long, the PWM clear position is after the end of cycle A. Next, PWM waveform B, which is for cycle B, is output.

The operations shown below occur when an attempt is made to achieve the operations shown in **Figure 27.45, PWM Output by Outputting a Delay Pulse with Dead Time** by the three-phase PWM output with dead time.

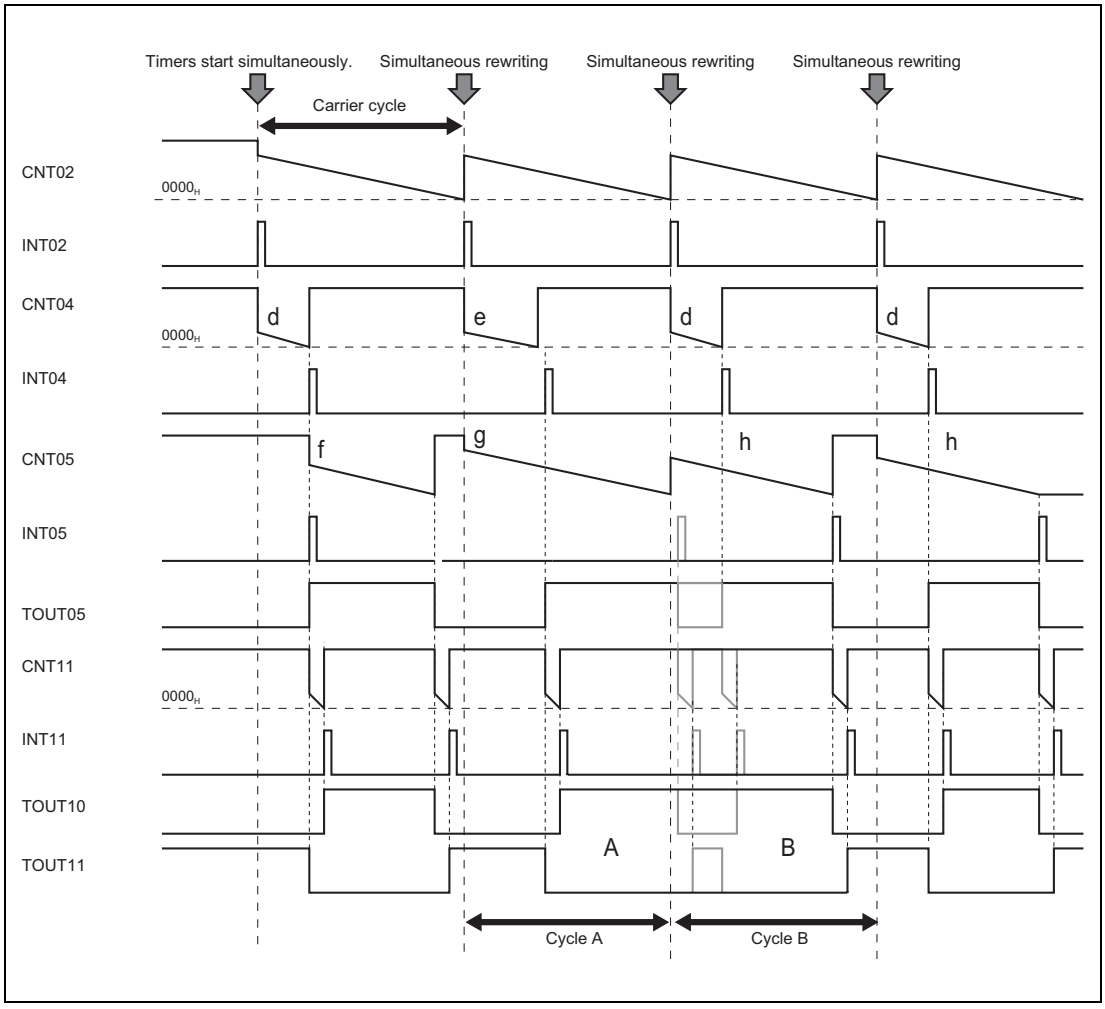


Figure 27.46 Output of a Three-Phase PWM Signal with Dead Time (1)

Figure 27.46 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 that combines PWM waveform A and PWM waveform B is output.

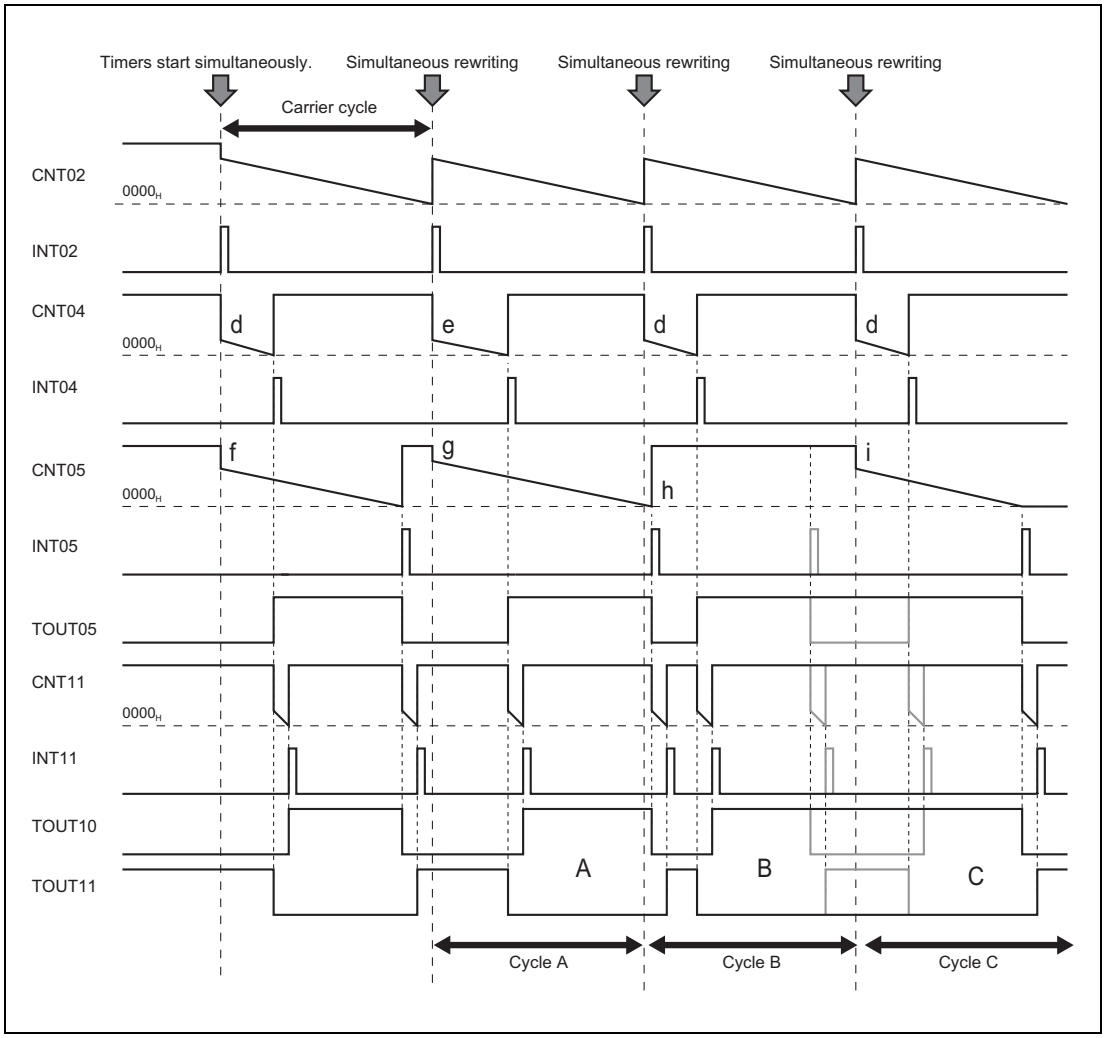


Figure 27.47 Output of a Three-Phase PWM Signal with Dead Time (2)

Figure 27.47 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 that combines PWM waveform B and PWM waveform C 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 27.48 shows a timing chart for outputting a delay pulse with dead time.

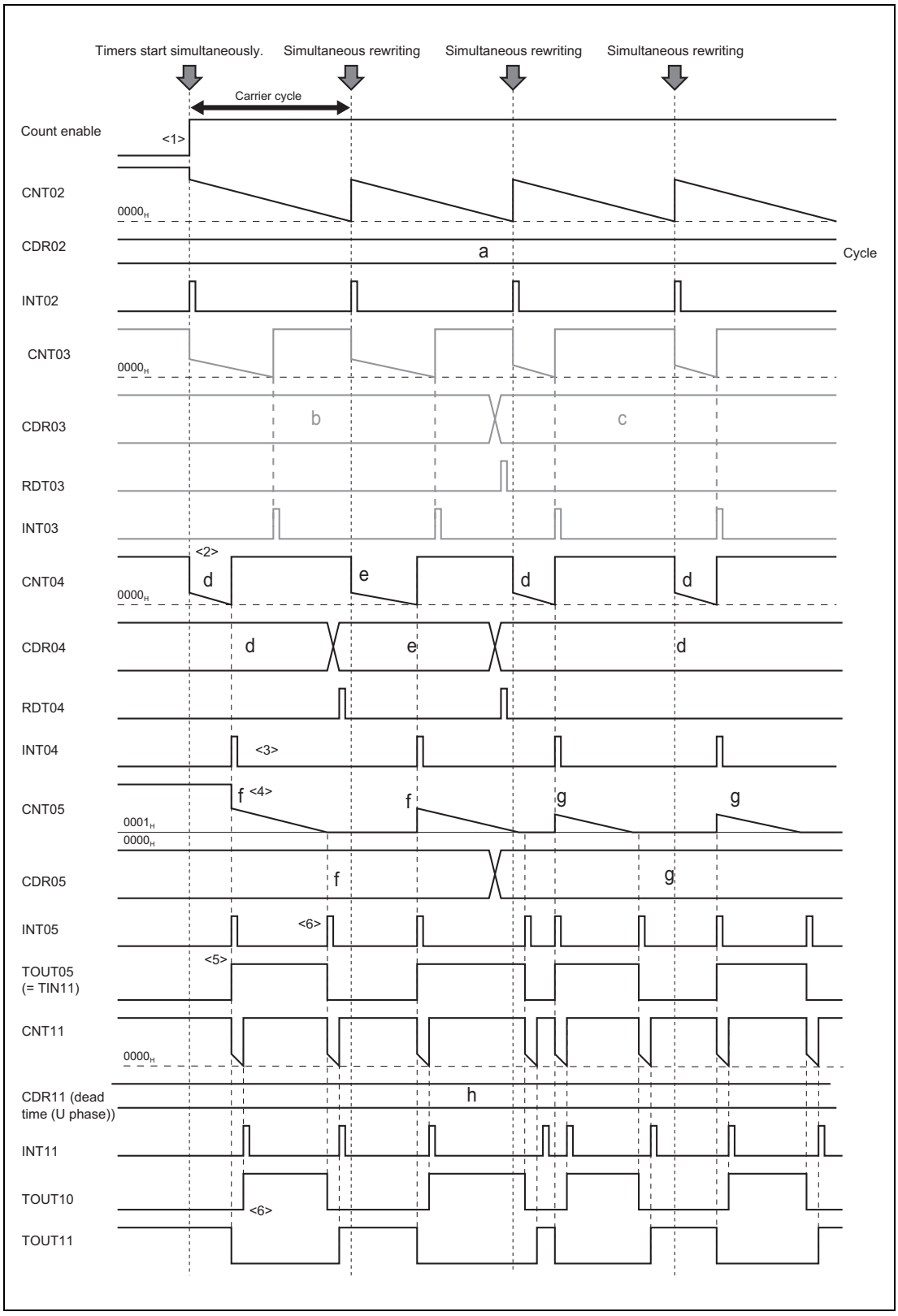


Figure 27.48 Output of a Delay Pulse with Dead Time

The output of a delay pulse with dead time shown in **Figure 27.48** 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).

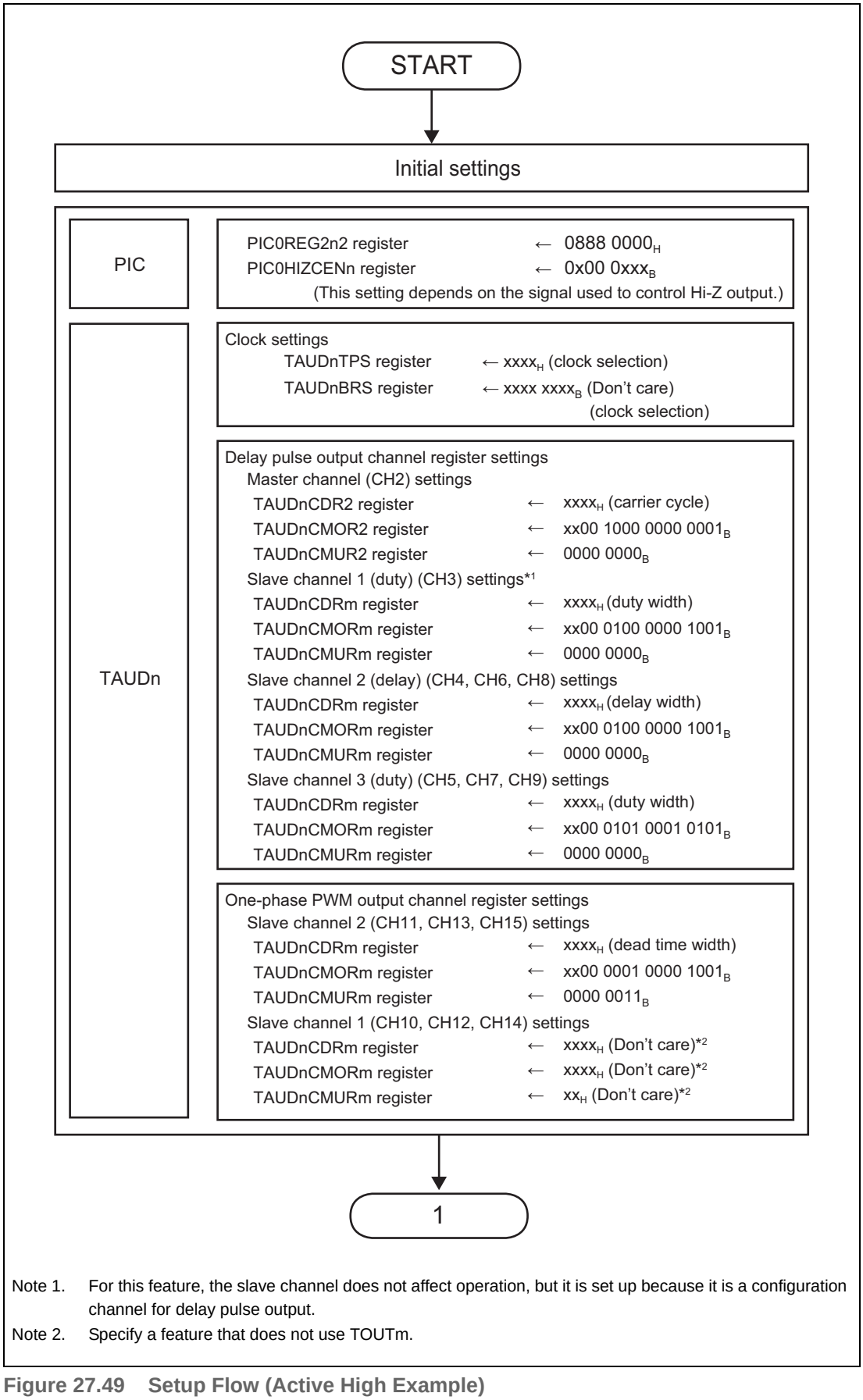
(4) When INT04 is generated, the settings from CDR05 are reloaded to CNT05, and then the CH5 (the PWM duty timer) operation starts.

(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 output by detecting a TIN11 edge.

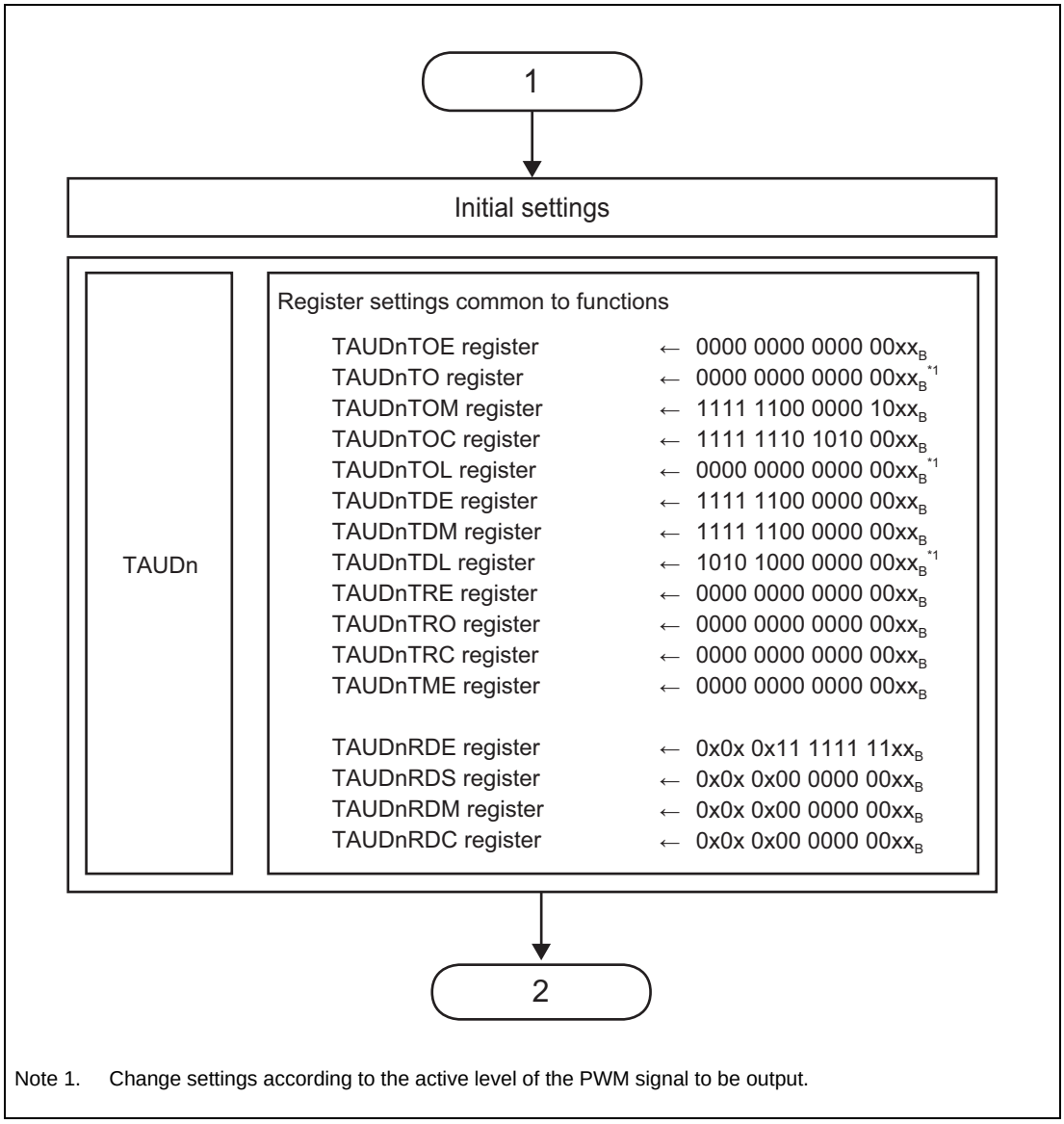
27.13.5 Setup Flow



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.

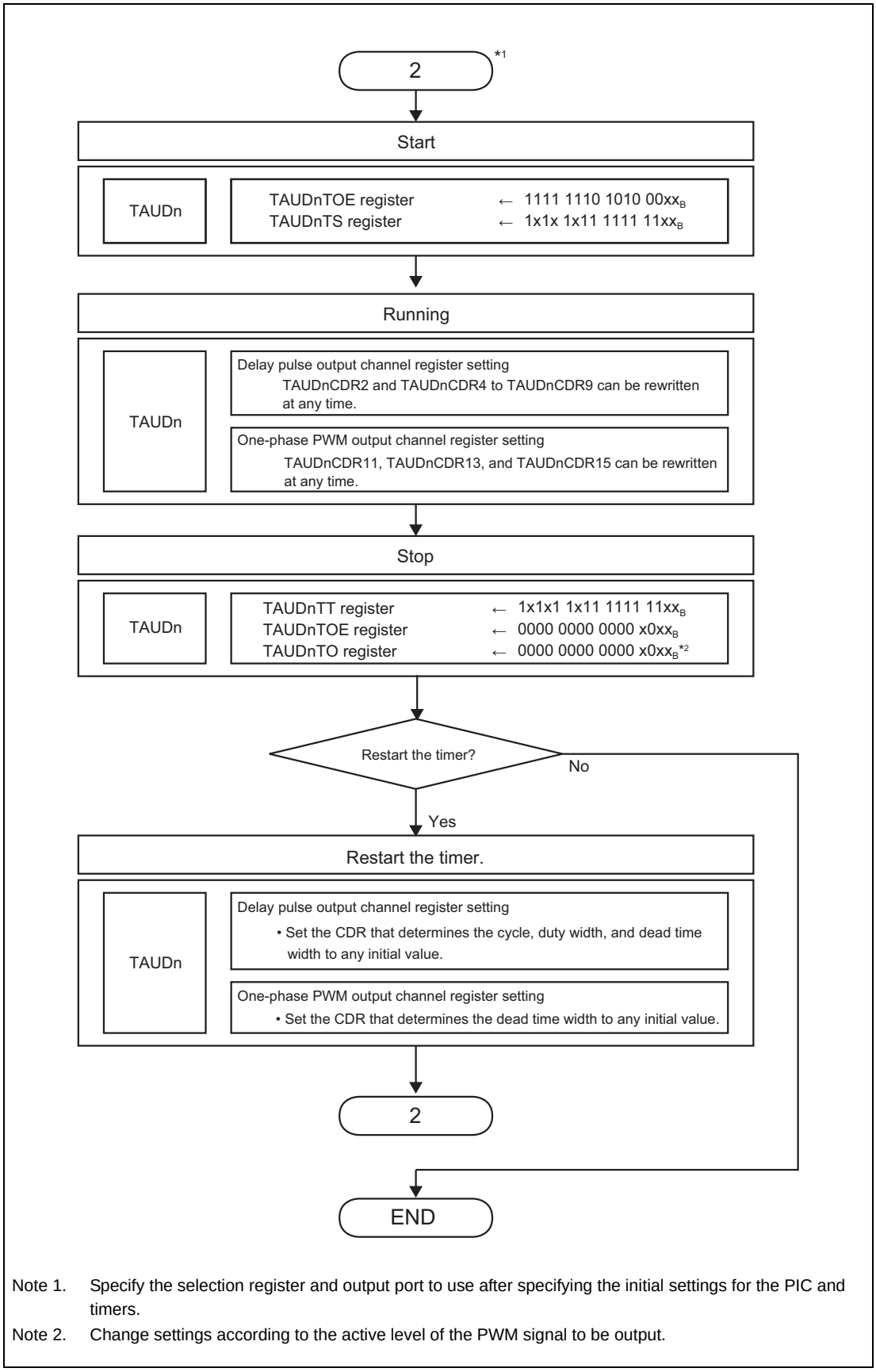
Note 2. Specify a feature that does not use TOUTm.

Figure 27.49 Setup Flow (Active High Example)



Note 1. Change settings according to the active level of the PWM signal to be output.

Figure 27.50 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 27.51 Setup Flow (Active High Example) (continued)

27.13.6 Example of Setting Up Operation Functions

This section provides example settings for each register.

27.13.6.1 TAUDn Settings

Table 27.70 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 23, Timer Array Unit D (TAUD)**.

Note 2. The same operation clock must be specified for the master channel and slave channel.

Table 27.71 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	Enable 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 23, 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 23, Timer Array Unit D (TAUD)**.

Table 27.72 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	Enable 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 23, 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 23, Timer Array Unit D (TAUD)**.

Table 27.73 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	Enable 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 23, 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 23, Timer Array Unit D (TAUD)**.

Table 27.74 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	Enable 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 23, 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 27.75 Common TAUDn Channel Settings (1/4)

Register	Bit position	Bit name	Setting	Remark
TAUDnTOE	15 to 10	TAUDnTOE15 to TAUDnTOE10	0 1	Disable the timer. Enable the timer.
	9	TAUDnTOE09	0 1	Disable the timer. Enable the timer.
	8	TAUDnTOE08	0	This is fixed to 0 because TOUT08 is not used.
	7	TAUDnTOE07	0 1	Disable the timer. Enable the timer.
	6	TAUDnTOE06	0	This is fixed to 0 because TOUT06 is not used
	5	TAUDnTOE05	0 1	Disable the timer. Enable 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	Output a low-level signal to TOUT15 to TOUT10.
	9 to 2	TAUDnTO09 to TAUDnTO02	0	Output a low-level signal to TOUT09 to TOUT02.
	1, 0	TAUDnTO01 TAUDnTO00	Don't care	

Table 27.75 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	Enable dead time control.*2
	9 to 2	TAUDnTDE09 to TAUDnTDE02	0	Disable dead time control.
	1, 0	TAUDnTDE01 TAUDnTDE00	Don't Care	
TAUDnTDM	15 to 10	TAUDnTDM15 to TAUDnTDM10	1	Output 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	Add dead time to the negative W phase period.
	14	TAUDnTDL14	0*1	Add dead time to the positive W phase period.
	13	TAUDnTDL13	1*1	Add dead time to the negative V phase period.
	12	TAUDnTDL12	0*1	Add dead time to the positive V phase period.
	11	TAUDnTDL11	1*1	Add dead time to the negative U phase period.
	10	TAUDnTDL10	0*1	Add dead time to the positive U phase period.
	9 to 2	TAUDnTDL09 to TAUDnTDL02	0	Invalid because dead time control is disabled.
	1, 0	TAUDnTDL01 TAUDnTDL00	Don't Care	
TAUDnTRE	15 to 2	TAUDnTRE15 to TAUDnTRE02	0	Disable real-time output.
	1, 0	TAUDnTRE01 TAUDnTRE00	Don't Care	

Table 27.75 Common TAUDn Channel Settings (3/4)

Register	Bit position	Bit name	Setting	Remark
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	Disable modulation output for timer output and real-time output.
	1, 0	TAUDnTME01 TAUDnTME00	Don't Care	
TAUDnRDE	15	TAUDnRDE15	0	Disable simultaneous rewriting.
	14	TAUDnRDE14	Don't Care	
	13	TAUDnRDE13	0	Disable simultaneous rewriting.
	12	TAUDnRDE12	Don't Care	
	11	TAUDnRDE11	0	Disable simultaneous rewriting.
	10	TAUDnRDE10	Don't Care	
	9 to 2	TAUDnRDE09 to TAUDnRDE02	1	Enable 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	Enable 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 2	TAUDnRDM09 to TAUDnRDM02	0	Load the signal when the master channel starts counting.
	1, 0	TAUDnRDM01 TAUDnRDM00	Don't Care	

Table 27.75 Common TAUDn Channel Settings (4/4)

Register	Bit position	Bit name	Setting	Remark
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 23, Timer Array Unit D (TAUD)**.

27.13.6.2 Peripheral Interconnections Settings

Table 27.76 Peripheral Interconnections Settings

Register	Bit position	Bit name	Setting	Remark
PIC0REG2n2	27	PIC0REG2n227	1	Select the input selected by the PIC0REG2n204 bit.
	23	PIC0REG2n223	1	Select the input selected by the PIC0REG2n203 bit.
	19	PIC0REG2n219	1	Select the input selected by the PIC0REG2n202 bit.
	4	PIC0REG2n204	0	Select TAUDnTTOUT09.
	3	PIC0REG2n203	0	Select TAUDnTTOUT07.
	2	PIC0REG2n202	0	Select TAUDnTTOUT05.

Section 28 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/F1L 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.

28.1 Features of RH850/F1L PWM-Diag

28.1.1 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.

A PWM channel can be output per PWGA unit. In this section, the number of PWGA units is used for the same meaning as the number of channels.

Table 28.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
PWBA						
Number of Units	1					
Name	PWBAn (n = 0)					
PWGA						
Number of Units	13	24	24	48	64	72
Name	PWGAn (n = 12)	PWGAn (n = 23)	PWGAn (n = 23)	PWGAn (n = 47)	PWGAn (n = 63)	PWGAn (n = 71)
PWSA						
Number of Units	1					
Name	PWSAn (n = 0)					

Table 28.2 Index

Index	Meaning
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, ..., 70_71).
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 indexes of each product.

Table 28.3 Indexes of Products

Indexes of each product				
48 pins	**64 pins, 80 pins**	**100 pins**	**144 pins**	**176 pins**
x = 00, 02, ,,, 12 y = 01, 03, ,,, 13*1	x = 00, 02, ,,, 22 y = 01, 03, ,,, 23	x = 00, 02, ,,, 46 y = 01, 03, ,,, 47	x = 00, 02, ,,, 62, y = 01, 03, ,,, 63	x = 00, 02, ,,, 70 y = 01, 03, ,,, 71
j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7
k = 0	k = 0	k = 0, 1	k = 0, 1	k = 0 to 2

Note 1. Channels 0 to 12 are provided in PWM-Diag.

28.1.2 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 in general.

Table 28.4 Register Base Address

Base Address Name	Base Address
<PWBA0_base>	FFE7 2800_H
<PWGAn_base>	FFE7 1000_H + 40_H × n
<PWSA0_base>	FFE7 0000_H

28.1.3 Clock Supply

Clock supply by and to PWM-Diag is listed in the following table.

Table 28.5 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
PWBAn	PCLK	CKSCLK_IPERI2
PWGAn	PCLK	CKSCLK_IPERI2
PWSAn	PCLK	CKSCLK_IPERI2

28.1.4 Interrupt Requests

PWM-Diag interrupt requests are listed in the following table.

Table 28.6 Interrupt Requests (1/3)

Unit Interrupt Signal	Signal Outline	Interrupt Number	DMA Trigger Number
PWGA_INT0	PWGA0 interrupt	84	—
PWGA_INT1	PWGA1 interrupt	85	—
PWGA_INT2	PWGA2 interrupt	86	—
PWGA_INT3	PWGA3 interrupt	87	—
PWGA_INT4	PWGA4 interrupt	77	—
PWGA_INT5	PWGA5 interrupt	78	—
PWGA_INT6	PWGA6 interrupt	79	—
PWGA_INT7	PWGA7 interrupt	80	—

Table 28.6 Interrupt Requests (2/3)

Unit Interrupt Signal	Signal Outline	Interrupt Number	DMA Trigger Number
PWGA_INT8	PWGA8 interrupt	88	—
PWGA_INT9	PWGA9 interrupt	89	—
PWGA_INT10	PWGA10 interrupt	90	—
PWGA_INT11	PWGA11 interrupt	91	—
PWGA_INT12	PWGA12 interrupt	92	—
PWGA_INT13	PWGA13 interrupt	93	—
PWGA_INT14	PWGA14 interrupt	94	—
PWGA_INT15	PWGA15 interrupt	95	—
PWGA_INT16	PWGA16 interrupt	137	—
PWGA_INT17	PWGA17 interrupt	139	—
PWGA_INT18	PWGA18 interrupt	141	—
PWGA_INT19	PWGA19 interrupt	143	—
PWGA_INT20	PWGA20 interrupt	116	—
PWGA_INT21	PWGA21 interrupt	117	—
PWGA_INT22	PWGA22 interrupt	118	—
PWGA_INT23	PWGA23 interrupt	119	—
PWGA_INT24	PWGA24 interrupt	176	—
PWGA_INT25	PWGA25 interrupt	177	—
PWGA_INT26	PWGA26 interrupt	145	—
PWGA_INT27	PWGA27 interrupt	178	—
PWGA_INT28	PWGA28 interrupt	179	—
PWGA_INT29	PWGA29 interrupt	180	—
PWGA_INT30	PWGA30 interrupt	147	—
PWGA_INT31	PWGA31 interrupt	149	—
PWGA_INT32	PWGA32 interrupt	181	—
PWGA_INT33	PWGA33 interrupt	182	—
PWGA_INT34	PWGA34 interrupt	183	—
PWGA_INT35	PWGA35 interrupt	184	—
PWGA_INT36	PWGA36 interrupt	185	—
PWGA_INT37	PWGA37 interrupt	186	—
PWGA_INT38	PWGA38 interrupt	187	—
PWGA_INT39	PWGA39 interrupt	188	—
PWGA_INT40	PWGA40 interrupt	189	—
PWGA_INT41	PWGA41 interrupt	190	—
PWGA_INT42	PWGA42 interrupt	191	—
PWGA_INT43	PWGA43 interrupt	192	—
PWGA_INT44	PWGA44 interrupt	193	—
PWGA_INT45	PWGA45 interrupt	194	—
PWGA_INT46	PWGA46 interrupt	195	—
PWGA_INT47	PWGA47 interrupt	196	—
PWGA_INT48	PWGA48 interrupt	232	—
PWGA_INT49	PWGA49 interrupt	233	—
PWGA_INT50	PWGA50 interrupt	234	—

Table 28.6 Interrupt Requests (3/3)

Unit Interrupt Signal	Signal Outline	Interrupt Number	DMA Trigger Number
PWGA_INT51	PWGA51 interrupt	235	—
PWGA_INT52	PWGA52 interrupt	236	—
PWGA_INT53	PWGA53 interrupt	237	—
PWGA_INT54	PWGA54 interrupt	238	—
PWGA_INT55	PWGA55 interrupt	239	—
PWGA_INT56	PWGA56 interrupt	240	—
PWGA_INT57	PWGA57 interrupt	241	—
PWGA_INT58	PWGA58 interrupt	242	—
PWGA_INT59	PWGA59 interrupt	243	—
PWGA_INT60	PWGA60 interrupt	244	—
PWGA_INT61	PWGA61 interrupt	245	—
PWGA_INT62	PWGA62 interrupt	246	—
PWGA_INT63	PWGA63 interrupt	247	—
PWGA_INT64	PWGA64 interrupt	269	—
PWGA_INT65	PWGA65 interrupt	270	—
PWGA_INT66	PWGA66 interrupt	271	—
PWGA_INT67	PWGA67 interrupt	272	—
PWGA_INT68	PWGA68 interrupt	273	—
PWGA_INT69	PWGA69 interrupt	274	—
PWGA_INT70	PWGA70 interrupt	275	—
PWGA_INT71	PWGA71 interrupt	276	—
PWSA_INT_QFULL	PWSA queue full interrupt	83	—

28.1.5 Reset Sources

PWM-Diag reset sources are listed in the following table. The individual PWM-Diag units are initialized by these reset sources.

Table 28.7 Reset Sources

Unit Name	Reset Source
PWBAn PWGAn PWSAn	All reset sources (ISORES)

28.1.6 External Input/Output Signals

External input/output signals of the PWM-Diag are listed below.

Table 28.8 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
PWGA_TOUTn (Unit: PWGA)	PWGA unit n output	PWGAnO

28.1.7 Internal Signals

The I/O signals for connecting two PWM-Diag channels or a PWM-Diag and another function are listed below.

Table 28.9 Internal Output Signals

Unit Signal Name	Outline	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

28.1.8 Outline of Functions

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 PWBA0EMU register.

PWGA

PWGA outputs PWM waveforms and A/D conversion trigger to PWSA by using the input clock PWMCLKm from PWBA.

- PWM waveform output PWGA_TOUTn
  This generator outputs PWM waveforms from the PWGA_TOUTn pin. The PWM cycle is controlled by the timing of underflows or overflows 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 PWGA0CRDR[11:0] > PWGA0CSDR[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 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, it is generated with the timing of underflows or overflows of the PWGAnCNT register (PWM cycle).

PWSA

PWSA transmits the required settings 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) is output to the A/D converter.
  At the same timing, A/D conversion trigger is output to ADCA0 or ADCA1. (A maximum of eight input trigger signal PWGA_TRGOUTn information 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 stored in the PWSAnQUE register.
- Queuing of A/D conversion trigger 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, where 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 29, A/D Converter (ADCA)**.

For the A/D converter operation with the PWM-Diag function, see **Section 29.4.7.1, A/D Conversion with PWM-Diag Enabled**.

28.1.9 Block Diagram

The following figure shows an example of connecting the LED control circuit combining the PWM-Diag and the A/D converter.

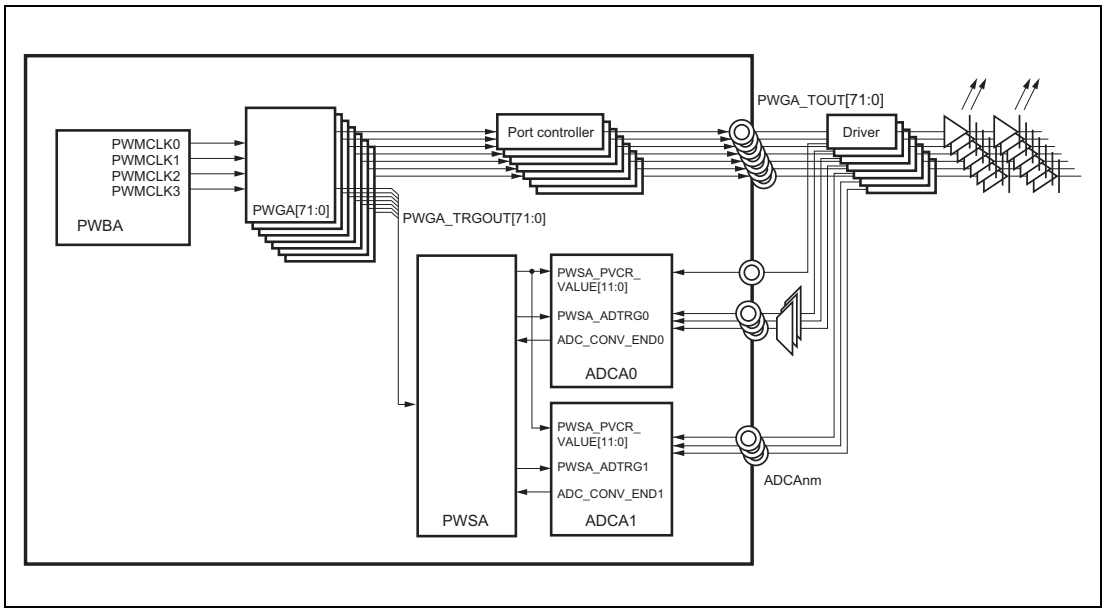


Figure 28.1 Example of Connecting the LED Control Circuit using the PWM-Diag and the A/D Converter

28.2 Registers

28.2.1 List of Registers

PWM-Diag registers are listed in the following table.

<PWBAn_base>, <PWSAn_base>, and <PWGAn_base> are defined in **Section 28.1.2, Register Base Address**.

Table 28.10 List of PMW-Diag 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
	PWGAnCRDR buffer register	PWGAnCTBR	<PWGAn_base> + $002C_H$
—	PWGA synchronous trigger register	SLPWGAk	FFBC 1000_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> + $000C_H$

28.2.1.1 PWBAnBRSm Register

This register sets the clock cycle of PWMCLKm.

Access: This register can be read/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 28.11 PWBAnBRSm Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing to these bits, 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).

28.2.1.2 PWBAnTE Register

This is a status register that indicates the output status of PWMCLKm (m = 0 to 3).

Access: This register 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 28.12 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

28.2.1.3 PWBAnTS Register

This register is a start trigger register for PWMCLKm (n = 0 to 3).

Access: This register can be witten 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	W	W	W	W	W	W	W	W

Table 28.13 PWBAnTS Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing to these bits, write the value after reset.
3	PWBAnTS3	Start trigger bit for PWMCLK3 0: Writing 0 does not work as a function. 1: Starts the output of PWMCLK3.
2	PWBAnTS2	Start trigger bit for PWMCLK2 0: Writing 0 does not work as a function. 1: Starts the output of PWMCLK2.
1	PWBAnTS1	Start trigger bit for PWMCLK1 0: Writing 0 does not work as a function. 1: Starts the output of PWMCLK1.
0	PWBAnTS0	Start trigger bit for PWMCLK0 0: Writing 0 does not work as a function. 1: Starts the output of PWMCLK0.

28.2.1.4 PWBAnTT Register

This register is a stop trigger register for PWMCLKm (m = 0 to 3).

Access: This register 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	W	W	W	W	W	W	W	W

Table 28.14 PWBAnTT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing to these bits, write the value after reset.
3	PWBAnTT3	Stop trigger bit for PWMCLK3 0: Writing 0 does not work as a function. 1: Stops the output of PWMCLK3.
2	PWBAnTT2	Stop trigger bit for PWMCLK2 0: Writing 0 does not work as a function. 1: Stops the output of PWMCLK2.
1	PWBAnTT1	Stop trigger bit for PWMCLK1 0: Writing 0 does not work as a function. 1: Stops the output of PWMCLK1.
0	PWBAnTT0	Stop trigger bit for PWMCLK0 0: Writing 0 does not work as a function. 1: Stops the output of PWMCLK0.

28.2.1.5 PWBA0EMU Register

This register sets the operation at the emulation.

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

Address: <PWBAn_base> + $001C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PWBA0SVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 28.15 PWBA0EMU Register Contents

Bit Position	Bit Name	Function
7	PWBA0SVSDIS	(When the EPC.SVSTOP bit = 0) The count clock is provided continuously when the debugger has control of the microcontroller (by 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 has control of the microcontroller (by break points, etc.). 1: The count clock is provided continuously when the debugger has control of the microcontroller (by 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 to these bits, write the value after reset.

28.2.1.6 PWGAnCTL — PWGA Control Register

PWGAnCTL is used to select the count clock from PWBA.

Access: This register can be read/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 28.16 PWGAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, 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).

28.2.1.7 PWGAnCNT — PWM Cycle Count Register

This is a count register.

Access: This register 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 28.17 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

28.2.1.8 PWGAnCSDR — PWM Output Set Condition Register

This register sets the setting condition for PWGA_TOUTn output.

Access: This register can be read/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 28.18 PWGAnCSDR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
11 to 0	PWGAnCSDR [11:0]	These bits set the setting condition for PWM output. The setting value is reflected to the PWGAnCSBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresonding CH = 1) or simultaneous rewrite (PWGAnRDT.PWGAnRDT = 1).

28.2.1.9 PWGAnCRDR — PWM Output Reset Condition Register

This register sets the reset condition for PWGA_TOUTn output.

Access: This register can be read/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 28.19 PWGAnCRDR Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 0	PWGAnCRDR [12:0]	These bits set the reset condition for PWM output. The setting value is reflected to the PWGAnCRBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresonding CH = 1) or simultaneous rewrite (PWGAnRDT.PWGAnRDT = 1).

28.2.1.10 PWGAnCTDR — PWGA_TRGOUTn Generation Condition Register

This register sets the generation condition for PWGA_TRGOUTn.

Access: This register can be read/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 28.20 PWGAnCTDR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
11 to 0	PWGAnCTDR [11:0]	These bits set the A/D conversion trigger generation condition for PWSAn. The setting value is reflected to the PWGAnCTBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresonding CH = 1) or simultaneous rewrite (PWGAnRDT.PWGAnRDT = 1).

28.2.1.11 PWGAnCSBR — PWGAnCSDR Buffer Register

This is a buffer register for the PWGAnCSDR register.

Access: This register 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 28.21 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 setting value is reflected to the PWGAnCSDR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresonding CH = 1) or simultaneous rewrite (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, the pin output is driven high.

28.2.1.12 PWGAnCRBR — PWGAnCRDR Buffer Register

This is a buffer register for the PWGA_TOUTn reset condition.

Access: This register 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 28.22 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 setting value is reflected to the PWGAnCRDR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresonding CH = 1) or simultaneous rewrite (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, the pin output is driven low.

28.2.1.13 PWGAnCTBR — PWGAnCTDR Buffer Register

This is a buffer register for the PWGA_TRGOUTn generation condition.

Access: This register 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 28.23 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 setting value is reflected to the PWGAnCTDR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresonding CH = 1) or simultaneous rewrite (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, a trigger is transmitted to PWSAn.

28.2.1.14 PWGAnRSF — Buffer Register Reload Status Register

This register is a status register for simultaneous rewrite control.

Access: This register 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 28.24 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.

28.2.1.15 PWGAnRDT — Buffer Register Reload Trigger Register

This is a simultaneous rewrite request trigger register.

Access: This register 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 28.25 PWGAnRDT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	PWGAnRDT	Simultaneous Rewrite Request Trigger 0: Writing 0 does not work as a function. 1: Triggers the simultaneous rewrite request for the compare registers (PWGA0CSDR, PWGA0CRDR, and PWGA0CTDR0), and sets PWGA0RSF.PWGA0RSF to 1.

28.2.1.16 SLPWGAk — PWGA Simultaneous Trigger Register (k = 0 to 2)

This register triggers start and stop for respective channels simultaneously.

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

Address: FFBC 1000_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 28.26 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[7:0]: PWGA71 - PWGA64

28.2.1.17 PWSAnCTL Register

This register is used to control operations of PWSA.

Access: This register can be read/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 28.27 PWSAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, 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.

28.2.1.18 PWSAnSTR Register

This is a status register that indicates the storage condition of the trigger generated channel number in the PWSAnQUEj register.

Access: This register 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 28.28 PWGAnSTR 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 PWSAQUEj registers do not store the channel number. 1: All of the PWSAnQUEj registers store the channel number.
0	PWSAnQNE	Bit indicating that there is a trigger in the trigger queue 0: The channel number is not stored in the PWSAnQUEj register, or the A/D conversion is in process while only PWSAnQUE0 stores the channel number. 1: The channel waiting for the conversion is stored in the PWSAnQUEj register after j = 1.

28.2.1.19 PWSAnSTC Register

This register clears the status of the PWSAnSTR register.

Access: This register 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	W	W	W	W	W	W	W	W

Table 28.29 PWGAnSTC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing to these bits, write the value after reset.
1	PWSAnCLFL	PWSAnQFL Clear 0: PWSAnQFL retains the status (Writing 0 does not work as a function). 1: PWSAnQFL is cleared to 0.
0	PWSAnCLNE	PWSAnQNE Clear 0: PWSAnQNE retains the status (Writing 0 does not work as a function). 1: PWSAnQNE is cleared to 0.

28.2.1.20 PWSAnQUEj (j = 0 to 7) Register

This register stores the channel number that received the trigger from PWGAn.

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

Address: <PWSAn_base> + $0020_H + j \times 4_H$

Value after reset: RH850/F1L 48-, 64-, 80-, and 100-pin products: $3F_H$
RH850/F1L 144- and 176-pin products: $7F_H$

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 28.30 PWSAnQUEj Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	PWSAnQUEj [6:0]	Stores the channel number (0 to 71) of the trigger-generated PWGA in order from PWSAnQUE0 to PWSAnQUE7. After the A/D conversion of PWSAnQUE0 is completed, the values in PWSAnQUE1 to PWSAnQUE07 shift to PWSAnQUE0 to PWSAnQUE6.

NOTE

If a trigger occurs simultaneously in multiple channels, the trigger with the smaller channel number has priority.

28.2.1.21 PWSAnPVCRx_y (x = 00, 02, 04 ... 70, y = 01, 03, 05 ... 71) Register

This register is used to set the corresponding A/D converter for each channel.

Consecutive two 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 setting value is transmitted to the ADCAnPWDVCR register of the A/D converter.

For the ADCAnPWDVCR register, see **Section 29, A/D Converter (ADCA)**.

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

Address: <PWSAn_base> + 0040_H + $xx2_H$

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 28.31 PWSAnPVCRx_y Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 29	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
28	PWSAnSLADy	RH850/F1L 48-, 64-, 80-, and 100-pin products: When writing to this bit, write the value after reset. RH850/F1L 144- and 176-pin products: ADCA Select (odd-numbered channel) 0: Output to ADCA0. 1: Output to ADCA1.
27	PWSAnVRDTy[27]	This bit indicates the setting value of the ADCAnPWDVCR.MPXE bit (odd-numbered channel).
26 to 24	PWSAnVRDTy[26:24]	This bit indicates the setting value of the ADCAnPWDVCR.MPXV[2:0] bits (odd-numbered channel).
23, 22	PWSAnVRDTy[23:22]	This bit indicates the setting value of the ADCAnPWDVCR.ULS[1:0] bits (odd-numbered channel).
21 to 16	PWSAnVRDTy[21:16]	This bit indicates the setting 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 to these bits, write the value after reset.
12	PWSAnSLADx	RH850/F1L 48-, 64-, 80-, and 100-pin products: When writing to this bit, write the value after reset. RH850/F1L 144- and 176-pin products: ADCA Select (even-numbered channel) 0: Output to ADCA0. 1: Output to ADCA1.

Table 28.31 PWSAnPVCRx_y Register Contents (2/2)

Bit Position	Bit Name	Function
11	PWSAnVRDTx [11]	This bit indicates the setting value of the ADCAnPWDVCR.MPXE bit. (even-numbered channel)
10 to 8	PWSAnVRDTx [10:8]	These bits indicate the setting value of the ADCAnPWDVCR.MPXV[2:0] bits. (even-numbered channel)
7 to 6	PWSAnVRDTx [7:6]	These bits indicate the setting value of the ADCAnPWDVCR.ULS[1:0] bits. (even-numbered channel)
5 to 0	PWSAnVRDTx [5:0]	These bits indicate the setting value of the ADCAnPWDVCR.GCTRL[5:0] bits. (even-numbered channel)

CAUTION

For the RH850/F1L 48-pin products, set the initial value to the 16 higher-order bits of PWSA0PVCR12_13.

28.2.1.22 PWSAnEMU — Emulation Control Register

This register is used to set the operation for emulation.

Access: This register can be read/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 28.32 PWSAnEMU Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	PWSAnSVSDIS	(When the EPC.SVSTOP bit = 0) The operation continues when the debugger has control of the microcontroller (by break points, etc.), regardless of the value of this bit (1 or 0). (When the EPC.SVSTOP bit = 1) 0: When the debugger has control of the microcontroller (by break points, etc.); – The output state to A/D is retained, the ADC_CONV_ENDn input at a break point is internally retained, and PWSA0QUEn is updated after break release. – The PWGA_TRGOUT input is accepted even at a break, and PWSA_INT_QFULL is even output. – Reading and writing to the register is possible. 1: The operation continues when the debugger has control of the microcontroller (by 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 (PWSB0QUE0 is the value after reset).

28.3 Operating Procedure

Procedures for setting when starting and stopping operation of PWM-Diag are illustrated below.

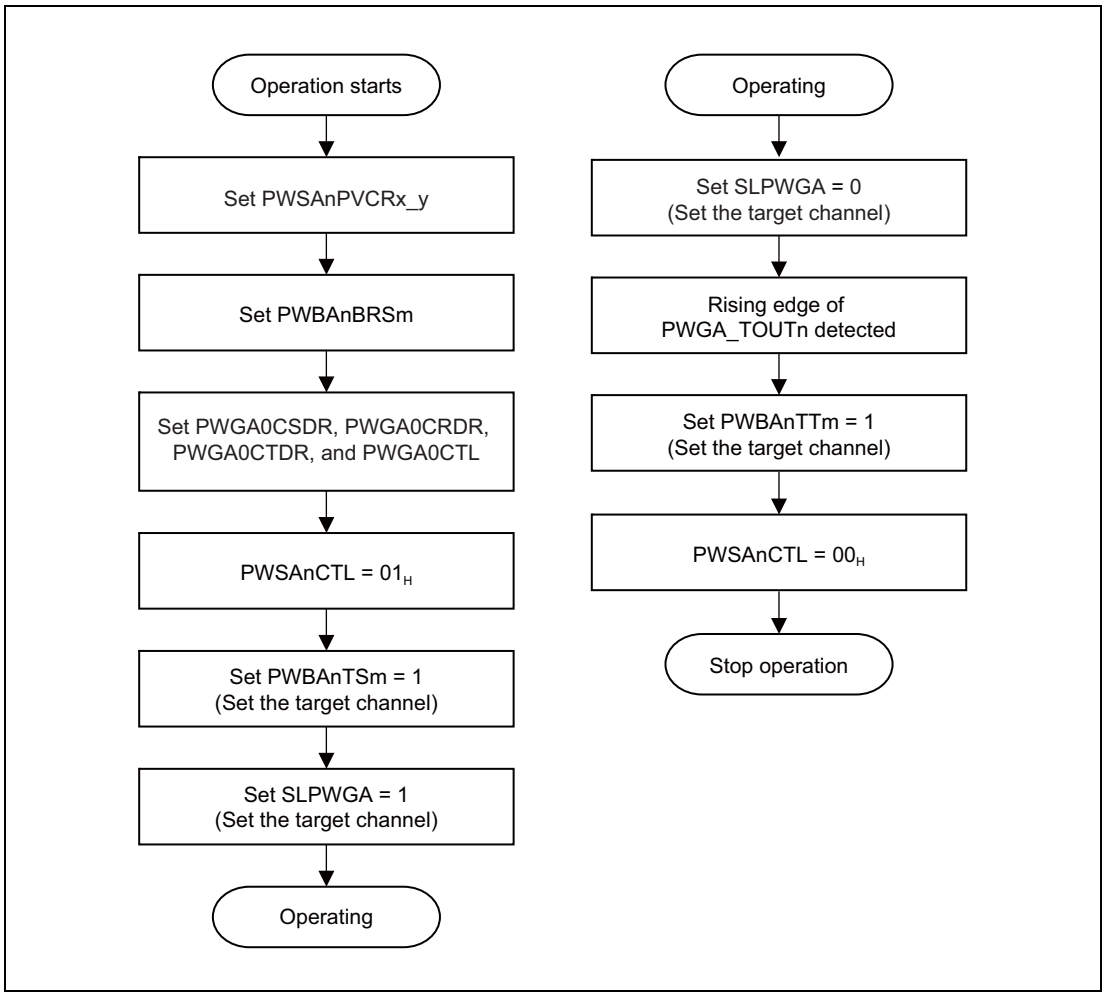
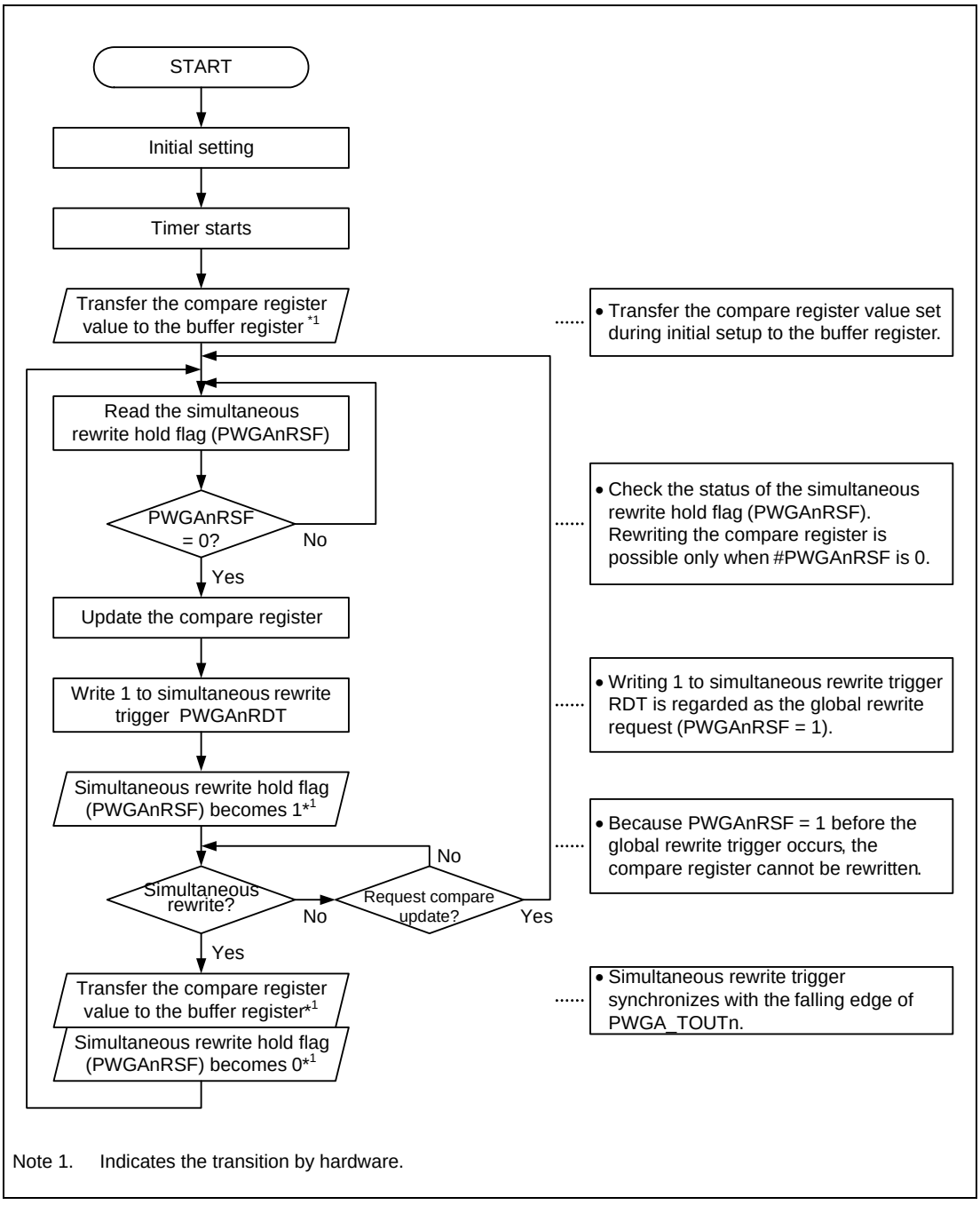


Figure 28.2 PWM-Diag Operating Procedure

Procedures for simultaneous rewrite of PWGA are illustrated below.

The described term “compare register” indicates PWGA0CSDR, PWGA0CRDR, or PWGA0CTDR0.

In addition, the described term “buffer register” indicates PWGA0CSBR, PWGA0CRBR, or PWGA0CTBR0.



Note 1. Indicates the transition by hardware.

Figure 28.3 Simultaneous Rewrite Procedure

28.4 Operation Waveform of PWM-Diag

This section describes the PWM generator (PWGA).

28.4.1 PWM Waveform Output by PWGA and Operation Waveform for A/D Conversion Trigger Output

28.4.1.1 Basic Operation Waveform of PWGA

The basic operation waveforms of PWGA are illustrated below.

The PWM waveform cycle and the duty can be calculated by the formula below.

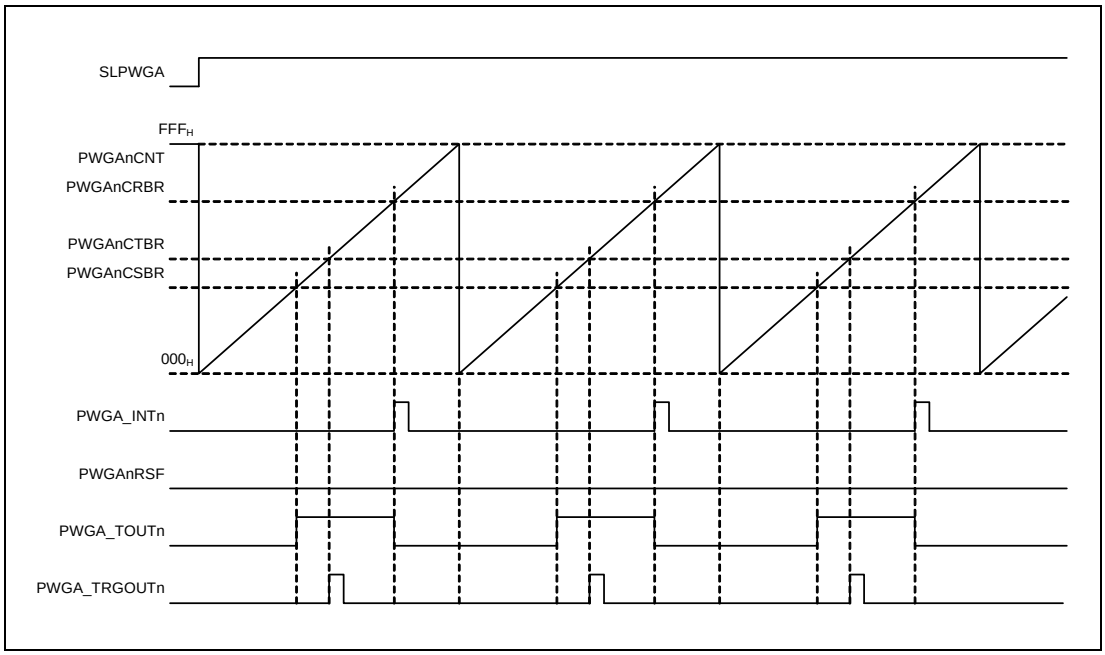


Figure 28.4 Basic Waveform

28.4.1.2 Operation Waveform when Simultaneous Rewrite for PWGA is Executed

The following figure illustrates the operation waveforms when simultaneous rewrite for PWGA is executed.

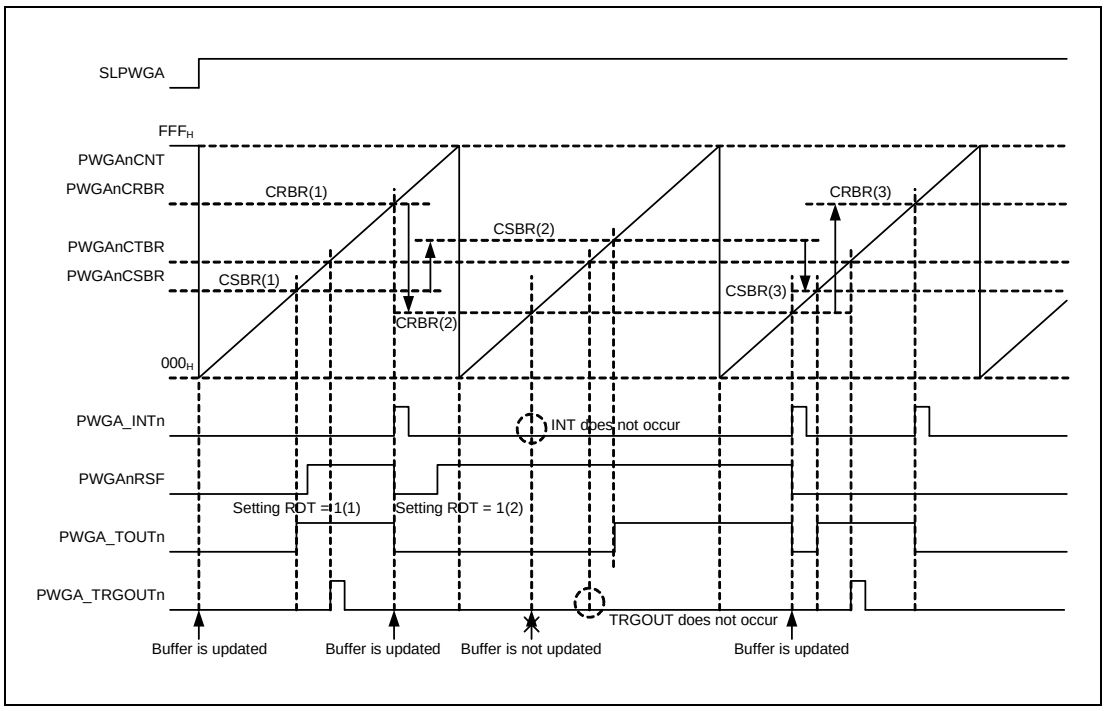


Figure 28.5 Waveform when Simultaneous Rewrite is Executed

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.

28.4.1.3 Operation Waveform when Stopping and Restarting PWGA Operation

The following figure illustrates the operation waveforms when stopping and restarting PWGA operation.

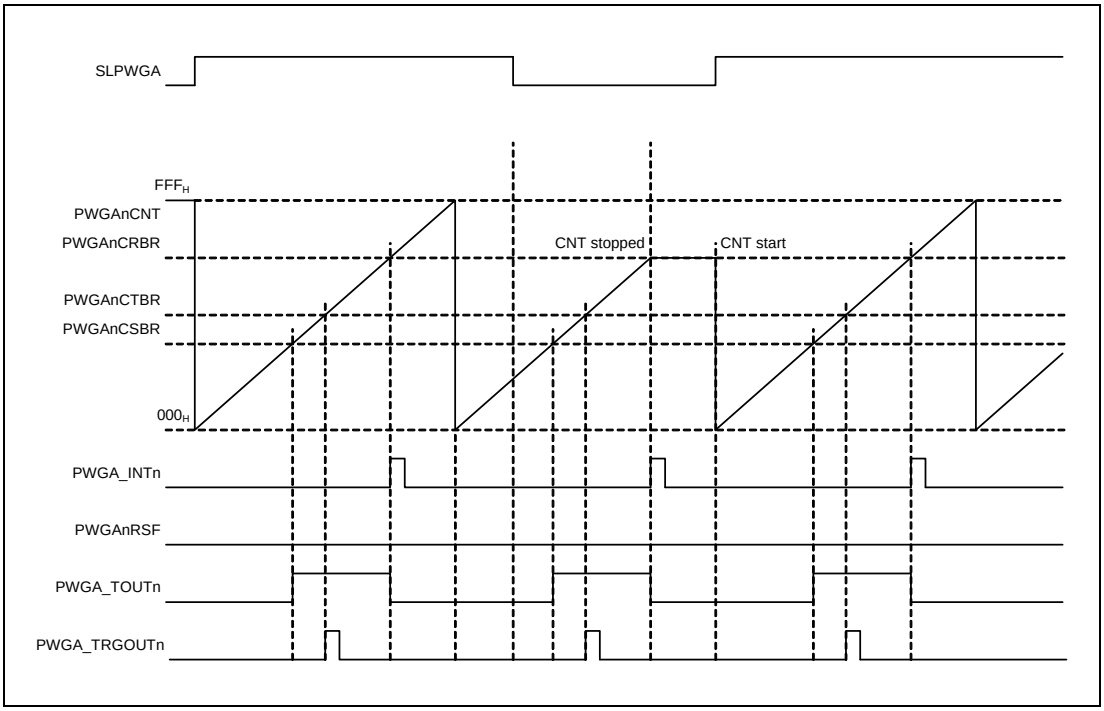


Figure 28.6 Stopping and Resuming Operation (1)

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.

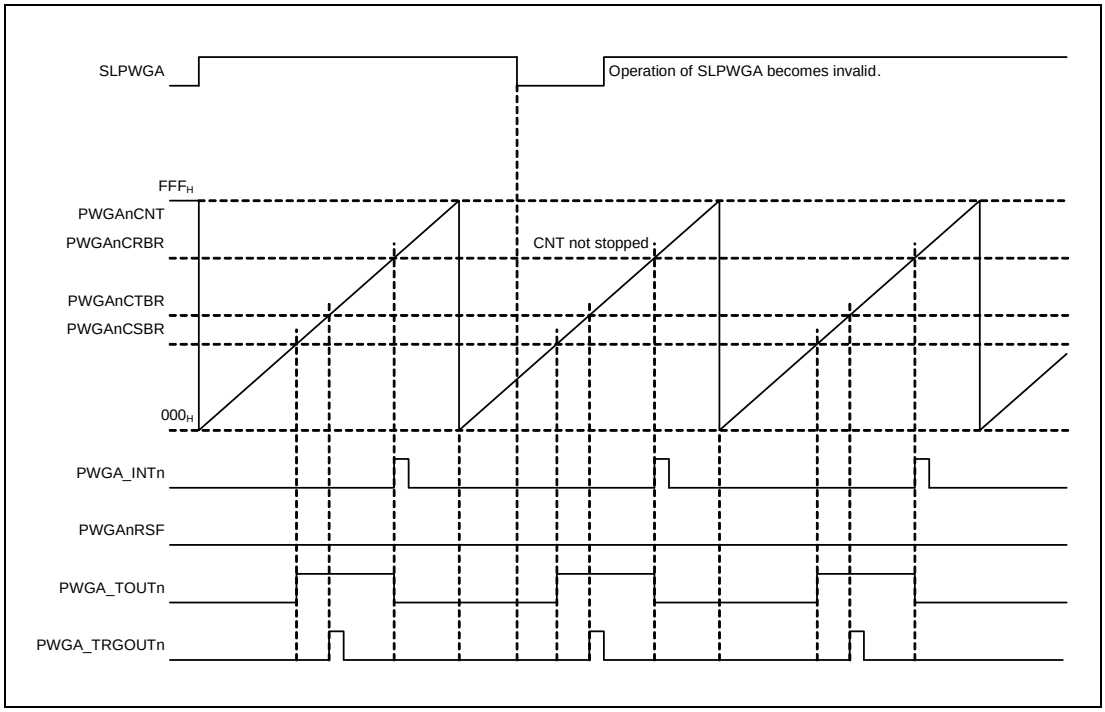


Figure 28.7 Stopping and Resuming Operation (2)

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.

28.4.1.4 Waveforms of PWGA Operation with Specific Settings

The following figures illustrate the waveforms of PWGA operation with specific settings.

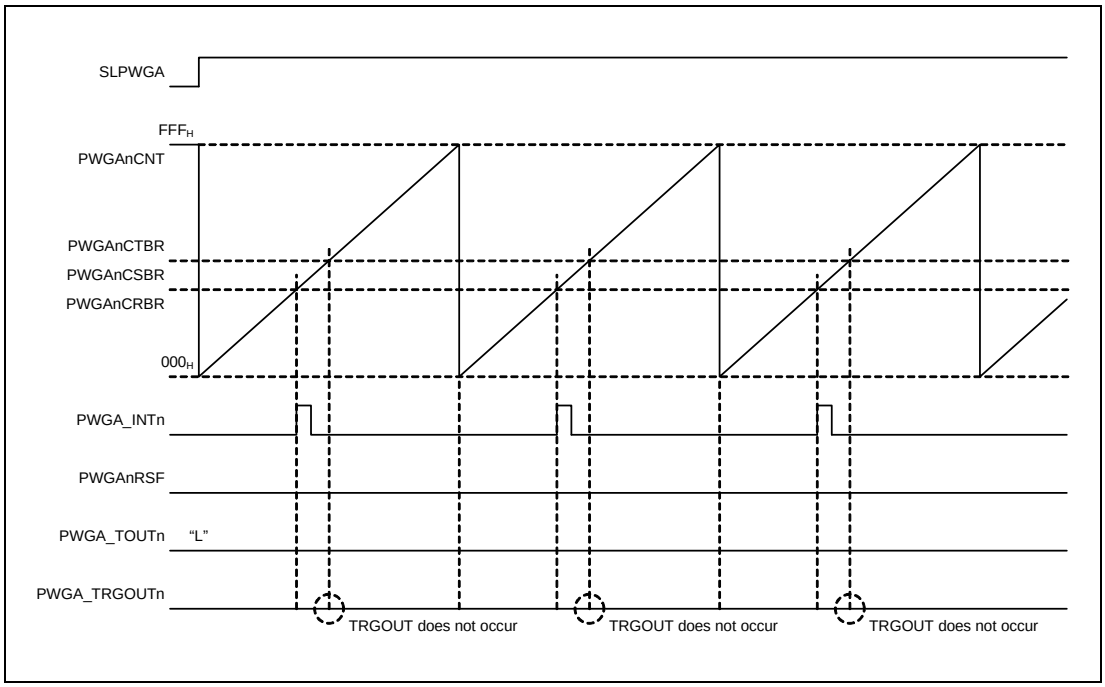


Figure 28.8 PWGA_TOUTn = 0% Output Waveform

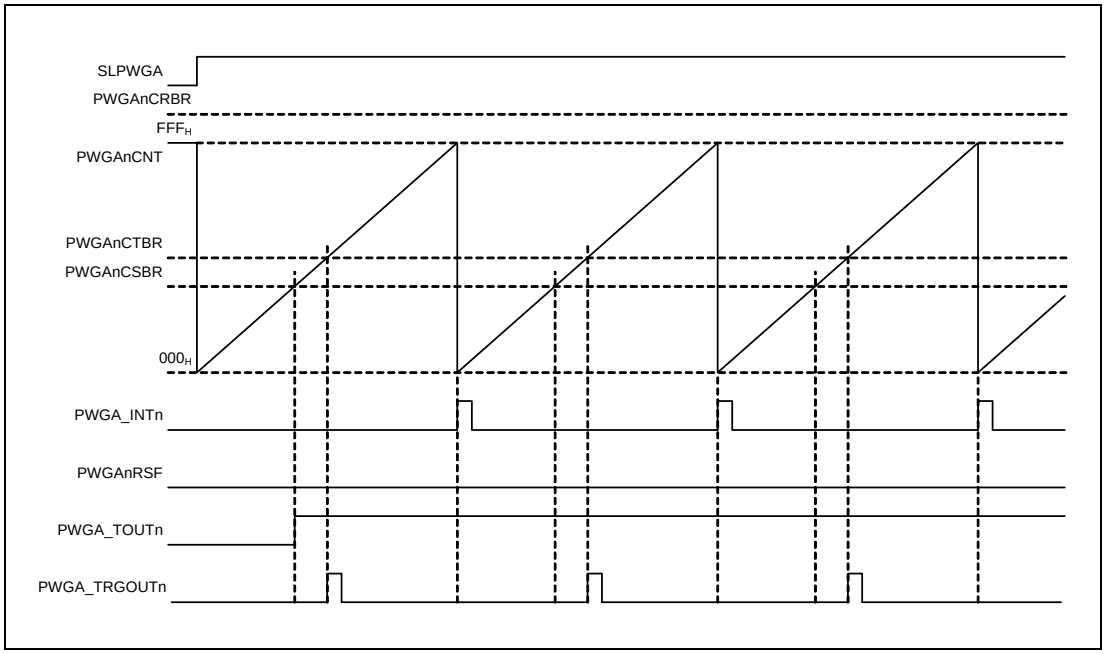
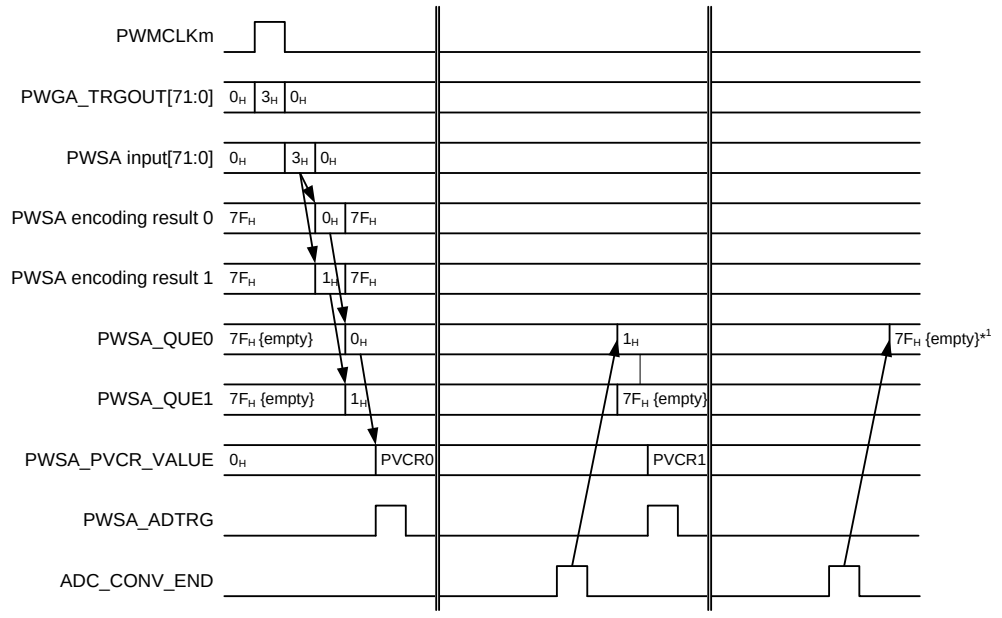


Figure 28.9 PWGA_TOUTn = 100% Output Waveform

28.4.2 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 28.10 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 bits information of PWSAnPVCR00_01 corresponding 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 information 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.

28.5 PWM-Diag Related Function in A/D Converter (ADCA)

This section describes the A/D converter used in the PWM-Diag function.

28.5.1 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 setting value corresponding to the channel under conversion is set in the following registers 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 29 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/F1L 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.

29.1 Features of RH850/F1L ADCA

29.1.1 Number of Units and Channels

This microcontroller has the following number of ADCA units.

Table 29.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	1	1	2	2
Name	ADCAn (n = 0)	ADCAn (n = 0)	ADCAn (n = 0)	ADCAn (n = 0)	ADCAn (n = 0, 1)	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 29.2 Unit Configurations and Physical Channels

Unit Name (Number of Channels) ADCAn		RH850/F1L 48 pins (12 ch)	RH850/F1L 64 pins (21 ch)	RH850/F1L 80 pins (25 ch)	RH850/F1L 100 pins (36 ch)	RH850/F1L 144 pins (48 ch)	RH850/F1L 176 pins (60 ch)
ADCA0	10/12 bit pin for conversion*1	8	10	11	16	16	16
	10 bit pin for conversion*2	4	11	14	20	20	20
ADCA1	10/12 bit pin for conversion*1					8	16
	10 bit pin for conversion*2					4	8

Note 1. When 12 bit mode is selected, 10 bit conversion cannot be performed.

Note 2. When 12 bit mode is selected, 10 bit pin for conversion cannot be used.

Table 29.3 Unit Configurations and Virtual Channels

Unit Name (Number of Channels) ADCAn	RH850/F1L 48 pins (20 ch)	RH850/F1L 64 pins (29 ch)	RH850/F1L 80 pins (37 ch)	RH850/F1L 100 pins (48 ch)	RH850/F1L 144 pins (62 ch)	RH850/F1L 176 pins (74 ch)
ADCA0	20	29	37	48	50	50
ADCA1	—	—	—	—	12	24

Table 29.4 Index

Index	Meaning
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 indexes of each product.

Table 29.5 Indexes of Products

Indexes of Each Product					
48 pins	**64 pins**	**80 pins**	**100 pins**	**144 pins**	**176 pins**
m = 0 to 11 (ADCA0)	m = 0 to 20 (ADCA0)	m = 0 to 24(ADCA0)	m = 0 to 35(ADCA0)	m = 0 to 35 (ADCA0) m = 0 to 11 (ADCA1)	m = 0 to 35 (ADCA0) m = 0 to 23 (ADCA1)
j = 00 to 19 (ADCA0)	j = 00 to 28 (ADCA0)	j = 00 to 36 (ADCA0)	j = 00 to 47 (ADCA0)	j = 00 to 49 (ADCA0) j = 00 to 11 (ADCA1)	j = 00 to 49 (ADCA0) j = 00 to 23 (ADCA1)
x = 1 to 3	x = 1 to 3	x = 1 to 3	x = 1 to 3	x = 1 to 3	x = 1 to 3
k = 0, 2, 4	k = 0, 2, 4	k = 0, 2, 4	k = 0 to 5	k = 0 to 5	k = 0 to 5

29.1.2 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 29.6 Register Base Address

Base Address Name	Base Address
<ADCA0_base>	FFF2 0000_H
<ADCA1_base>	FFF2 1000_H

29.1.3 Clock Supply

The ADCAn clock supply is shown in the following table.

Table 29.7 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
ADCA0	PCLK	CKSCLK_AADCA
ADCA1	PCLK	CKSCLK_IADCA

29.1.4 Interrupt Requests

ADCAn interrupt requests are listed in the following table.

Table 29.8 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number	Other Trigger Signals
ADCA0				
INT_ADE	A/D error interrupt request	47	—	Motor control
INT_SG1	Scan group 1 (SG1) end interrupt	10	4 (Channels 0 to 7)	LPS
INT_SG2	Scan group 2 (SG2) end interrupt	11	5 (Channels 0 to 7)	LPS
INT_SG3	Scan group 3 (SG3) end interrupt	12	6 (Channels 0 to 7)	LPS
ADC_CONV_END0	Scan group 4 (SG4) A/D conversion end signal	—	7 (Channels 0 to 7)	—
ADCA1				
INT_ADE	A/D error interrupt request	204	—	—
INT_SG1	Scan group 1 (SG1) end interrupt	205	39 (Channels 8 to 15)	—
INT_SG2	Scan group 2 (SG2) end interrupt	206	40 (Channels 8 to 15)	—
INT_SG3	Scan group 3 (SG3) end interrupt	207	41 (Channels 8 to 15)	—
ADC_CONV_END1	Scan group 4 (SG4) A/D conversion end signal	—	42 (Channels 8 to 15)	—

29.1.5 Reset Sources

ADCAn reset sources are listed in the following table. ADCAn is initialized by these reset sources.

Table 29.9 Reset Sources

Unit Name	Reset Source
ADCA0	Reset sources other than transition to DEEPSTOP mode (AWORES)
ADCA1	All reset sources (ISORES)

29.1.6 External Input/Output Signals

External input/output signals of ADCAn are listed below.

Table 29.10 ADCA0 External Input/Output Signals (1/2)

Unit Signal Name	Outline	Alternative Port Pin Signal
ADCA0		
ANI000	12-bit resolution analog input pin (for T&H)	ADCA0I0
ANI001	12-bit resolution analog input pin (for T&H)*1	ADCA0I1
ANI002	12-bit resolution analog input pin (for T&H)	ADCA0I2
ANI003	12-bit resolution analog input pin (for T&H)*1	ADCA0I3
ANI004	12-bit resolution analog input pin (for T&H)	ADCA0I4
ANI005	12-bit resolution analog input pin (for T&H)*1	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

Table 29.10 ADCA0 External Input/Output Signals (2/2)

Unit Signal Name	Outline	Alternative Port Pin Signal
ANI009	12-bit resolution analog input pin	ADCA0I9
ANI010	12-bit resolution analog input pin	ADCA0I10
ANI011	12-bit resolution analog input pin	ADCA0I1
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	ADCA0I12S
ANI029	10-bit resolution analog input pin	ADCA0I13S
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
ADCA0TRG0	External trigger pin (scan group 1)	ADCA0TRG0
ADCA0TRG1	External trigger pin (scan group 2)	ADCA0TRG1
ADCA0TRG2	External trigger pin (scan group 3)	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. The track and hold circuit (T&H) for ANI0m (m = 1, 3, 5) is only included on F1L devices with 100, 144 or 176 pins.

Table 29.11 ADCA1 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
ADCA1		
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
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
ADCA1TRG0	External trigger pin (scan group 1)	ADCA1TRG0
ADCA1TRG1	External trigger pin (scan group 2)	ADCA1TRG1
ADCA1TRG2	External trigger pin (scan group 3)	ADCA1TRG2

29.2 Overview

29.2.1 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 36 ADCA0 channels and 24 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 with no limit of numbers.
- Asynchronous/synchronous suspend and resume function
  A processing for a scan group can interrupt a running 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 twice 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 (MPX is not used)
  2.30 μs (MPX is used)

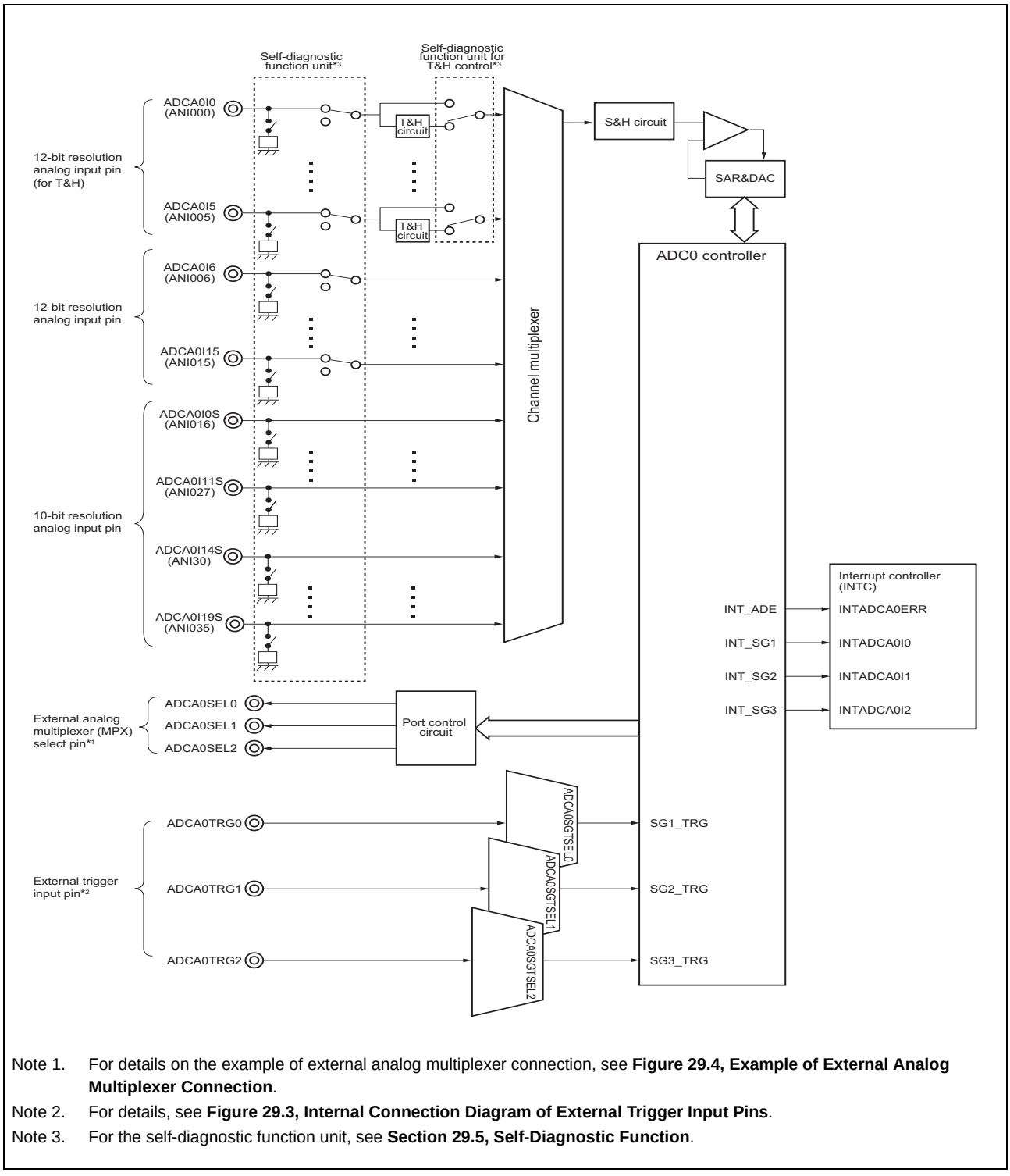
NOTE

- 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 35) for ADCA0 and ANI1m (m = 0 to 23) for ADCA1.
  In RH850/F1L, 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 24 virtual channels. The virtual channel specifies the physical channel to be scanned.
  Scans are executed in the order from the small virtual channel number. The scan order can be arbitrarily-specified with 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). AD conversion is executed in group unit. Each group can select the channel to be scanned by specifying the scan range, that is, the conversion start virtual channel and the conversion end virtual channel.

29.2.2 Block Diagram

The block diagram of ADCA0 is shown in **Figure 29.1**. The block diagram of ADCA1 is shown in **Figure 29.2**.

(1) Configuration of ADCA0



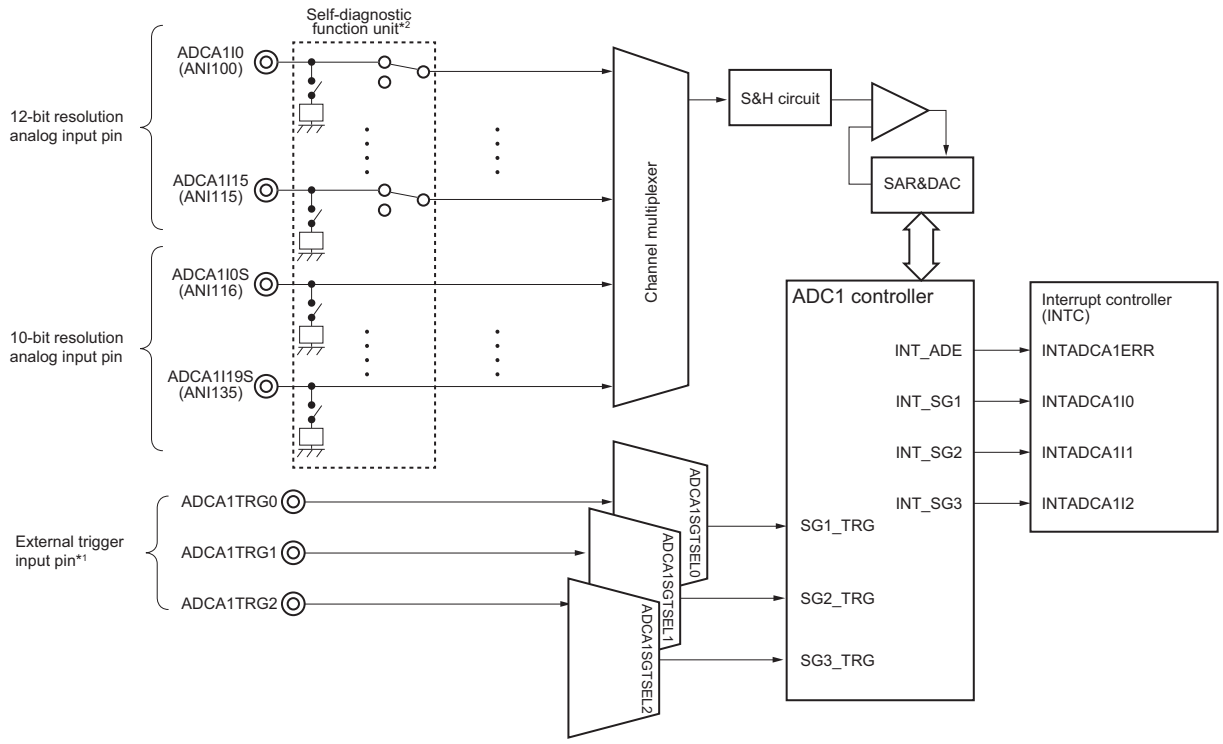
Note 1. For details on the example of external analog multiplexer connection, see **Figure 29.4, Example of External Analog Multiplexer Connection**.

Note 2. For details, see **Figure 29.3, Internal Connection Diagram of External Trigger Input Pins**.

Note 3. For the self-diagnostic function unit, see **Section 29.5, Self-Diagnostic Function**.

Figure 29.1 ADCA0 Block Diagram (RH850/F1L with 176 Pins)

(2) Configuration of ADCA1



Note 1. For details, see **Figure 29.3, Internal Connection Diagram of External Trigger Input Pins**.

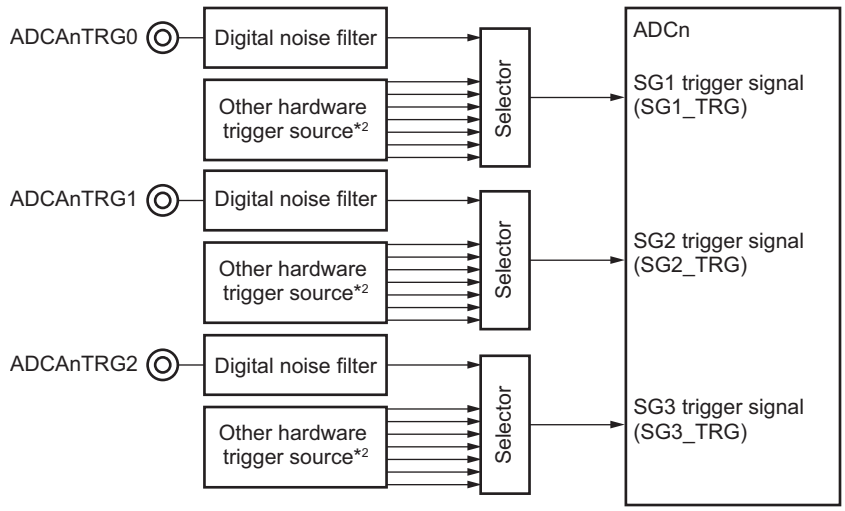
Note 2. For the self-diagnostic function unit, see **Section 29.5, Self-Diagnostic Function**.

Figure 29.2 ADCA1 Block Diagram (RH850/F1L with 176 Pins)

(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.



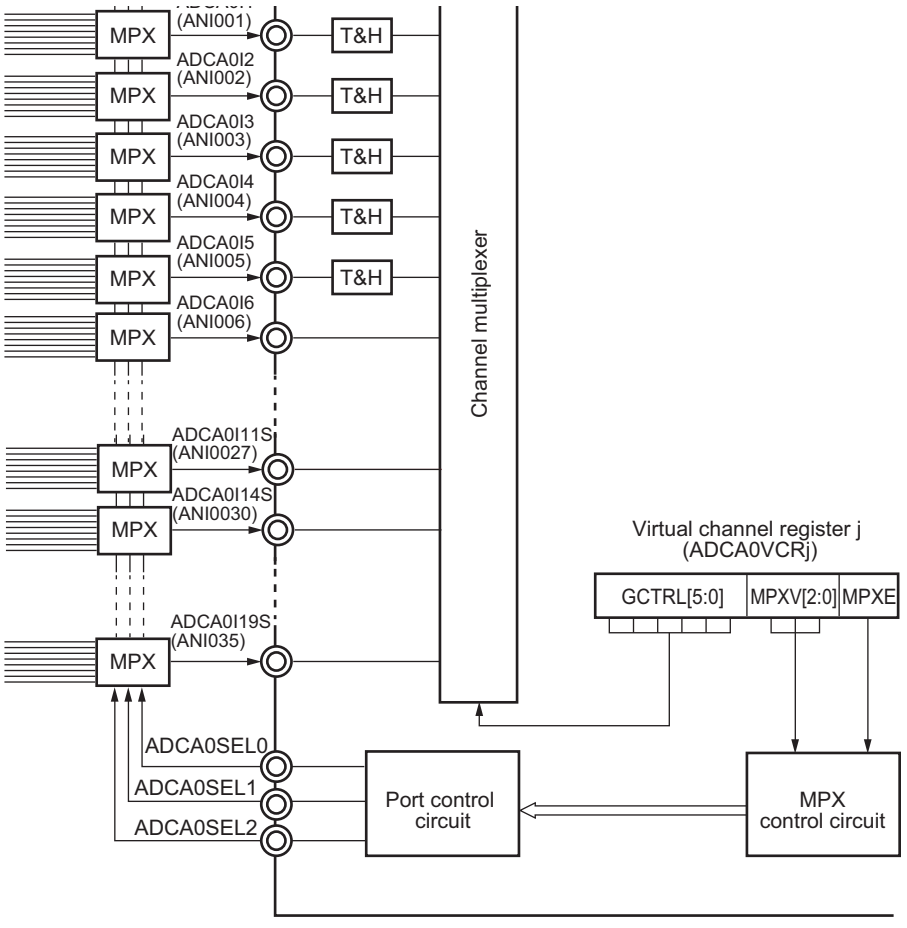
Note 1. For the condition setting to accept a trigger from external trigger input pin, see **Section 2.12, Noise Filter & Edge/Level Detector**.

Note 2. As for the hardware trigger sources, see **Table 29.47, List of A/D Conversion Hardware Triggers**.

Figure 29.3 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.

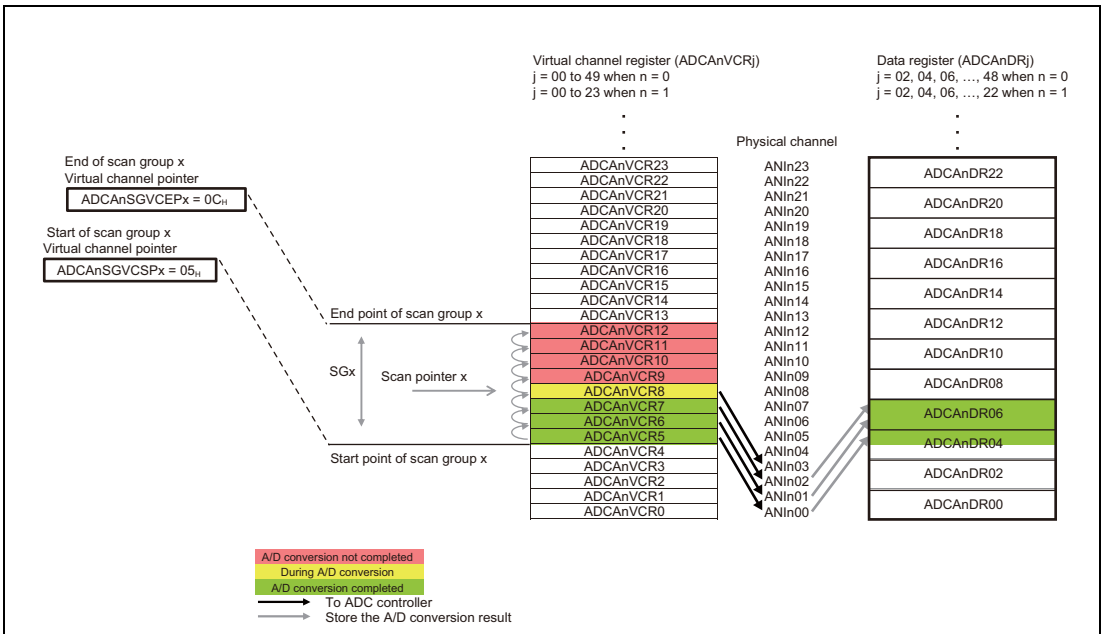


Note: The T&H function is disabled when the external analog multiplexer (MPX) is used.

Figure 29.4 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 m.

Figure 29.5 Usage Example of Virtual Register

29.3 Registers

29.3.1 List of Registers

ADCA registers are listed in the following table.

For details about <ADCAn_base>, see **Section 29.1.2, Register Base Address**.

Table 29.12 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 Unique 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 x 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 29.12 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

29.3.2 ADCA Specific Registers

This section describes the registers that are equipped in each of ADCA0 and ADCA1.

29.3.2.1 ADCAnVCRj — Virtual Channel Register j

This register is used to control the virtual channel.

Access: ADCAnVCRj can be read/written in 32-bit units.
ADCAnVCRjL can be read/written in 16-bit units.
ADCAnVCRjLL and ADCAnVCRjLH can be read/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 29.13 ADCAnVCRj Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, 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 written, 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 MPXE is set, however, a wait is inserted.
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 29.13 ADCAnVCRj Register Contents (2/2)

Bit Position	Bit Name	Function
5 to 0	GCTRL[5:0]	Physical Channel Select These bits are used to specify a physical channel to be assigned to virtual channel j (see **Table 29.14, Selection of Physical Channels**).

Note 1. These bits are only supported by ADCA0. With ADCA1, a value to be written to these bits should be the value after a reset.

Note 2. This bit is only supported when j = 33 to 35. Otherwise, when writing to this bit, write the value after a reset.

CAUTION

To prevent malfunction, ADCAnVCRj should be set when SGACT of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

Table 29.14 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)*1
0	1	1	0	0	1	ADCAnI9S (Physical channel ANIn25)*1
0	1	1	0	1	0	ADCAnI10S (Physical channel ANIn26)*1
0	1	1	0	1	1	ADCAnI11S (Physical channel ANIn27)*1
0	1	1	1	0	0	ADCAnI12S (Physical channel ANIn28)*1
0	1	1	1	0	1	ADCAnI13S (Physical channel ANIn29)*1
0	1	1	1	1	0	ADCAnI14S (Physical channel ANIn30)*1
0	1	1	1	1	1	ADCAnI15S (Physical channel ANIn31)*1

Table 29.14 Selection of Physical Channels (2/2)

GCTRL5	GCTRL4	GCTRL3	GCTRL2	GCTRL1	GCTRL0	Analog Input Pin to be Selected
1	0	0	0	0	0	ADCAnI16S (Physical channel ANIn32)*1
1	0	0	0	0	1	ADCAnI17S (Physical channel ANIn33)*1
1	0	0	0	1	0	ADCAnI18S (Physical channel ANIn34)*1
1	0	0	0	1	1	ADCAnI19S (Physical channel ANIn35)*1
1	0	0	1	0	0	Diagnosis channel for A/D converter
Other than above						Setting prohibited

Note 1. This setting only applies to ADCA0. Setting is prohibited in ADCA1 (RH850/F1L with 176 pins).

29.3.2.2 ADCAnPWDVCR — PWM-Diag Virtual Channel Register

This register controls the PWM-Diag virtual channel.

Access: ADCAnPWDVCR can only be read in 32-bit units.
ADCAnPWDVCRL can only be read in 16-bit units.
ADCAnPWDVCRLL can only be read in 8-bit units.
ADCAnPWDVCRLH can only 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 29.15 ADCAnPWDVCR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15	MPXE*1	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.
11 to 8	Reserved	When read, the value after reset is returned.
7 to 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 29.15 ADCAnPWDVCR Register Contents

Bit Position	Bit Name	Function
5 to 0	GCTRL[5:0]	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 29.14, Selection of Physical Channels**.

Note 1. ADCA0 only.

29.3.2.3 ADCAnDRj — Data Register j

This register is a 32-bit read-only register, which 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 can only be read in 32-bit units.
ADCAnDRjL and ADCAnDRjH can only 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 29.16 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) are 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 are 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 of the RH850/F1L with 176 pins)
   j = 00, 02, …, 20, 22 (for ADCA1 of the RH850/F1L with 176 pins)
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.

29.3.2.4 ADCAnDIRj — Data Supplementary Information Register j

This register is a 32-bit read-only register, which 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 the one for the ADCAnDRj register.

Access: This register can only 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 29.17 ADCAnDIRm 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 previous 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]	Holds the physical channel number (GCTRL) corresponding to the conversion result. The physical channel number to be stored is the number corresponding to the previous 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.
With ADCA1, reading these bits returns the value after a reset.

29.3.2.5 ADCAnPWDTSNDR — PWM-Diag Data Register

This register is a 32-bit read-only register, which 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 can be read only in 32-bit units.
ADCAnPWDTSNDRL can only be read in 16-bit units.

Address: ADCAnPWDTSNDR: <ADCAn_base> + 178_H
ADCAnPWDTSNDRL: <ADCAn_base> + 178_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 29.18 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.

29.3.2.6 ADCAnPWDDIR — PWM-Diag Data Supplementary Information Register

This register is a 32-bit read-only register, which 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 can be read only 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 29.19 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.
27, 26	Reserved	When read, the value after reset is returned.
25	WFLG	Write Flag 0: PWDDR or ADCAnPWDDIR is read. 1: The A/D converted value is stored in PWDDR.
24 to 22	Reserved	When read, the value after reset is returned.
21 to 16	ID[5:0]	Holds the physical channel number (GCTRL) corresponding to the conversion result.
15 to 0	PWDDR[15:0]	These bits are used to store the A/D conversion result for PWM-Diag.

Note 1. ADCA0 only.
With ADCA1, reading these bits returns the value after a reset.

29.3.2.7 ADCAnADHALTR — A/D Force Halt Register

This register is used to halt conversion for all SGs of ADCAn. The read value is always 0.

Access: ADCAnADHALTR can only be written in 32-bit units.
ADCAnADHALTRL can only be written in 16-bit units.
ADCAnADHALTRLL can only 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 29.20 ADCAnADHALTR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, 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.

29.3.2.8 ADCAnADCR — A/D Control Register

This register is used for ADCAn common control.

Access: This register can be read/written in 32/16/ 8-bit units.
ADCAnADCR can be read/written in 32-bit units.
ADCAnADCRL can be read/written in 16-bit units.
ADCAnADCRLL can be read/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 29.21 ADCAnADCR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When written, 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 written, write the value after reset.
5	CRAC	Alignment Control 0: The results of conversion to WDDR and ADCAnDRj are stored right-aligned. 1: The results of conversion to WDDR 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 written, write the value after reset.
1, 0	SUSMTD [1:0]	Suspend Mode Select Selects 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 interrupt. 01: Asynchronous suspend when a higher-priority SG (SG2, SG3, SG4) interrupts SG1, and synchronous suspend when a higher-priority SG (SG3, SG4) interrupts SG2, a higher-priority SG (SG4) interrupts SG3, or SVSTOP interrupts. 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 AD channel conversion for the lower-priority SG is resumed.

For details, see **Figure 29.21, Example of Synchronous Suspend and Resume Operation** and **Figure 29.22, Example of Asynchronous Suspend and Resume Operation**.

29.3.2.9 ADCAnMPXCURR — MPX Current Register

This register is used to store the MPX value for an external analog multiplexer.

Access: This register can only be read in 32/16/ 8-bit units.
ADCAnMPXCURR can only be read in 32-bit units.
ADCAnMPXCURRL can only be read in 16-bit units.
ADCAnMPXCURRLL can only 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 29.22 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

With the RH850/F1L, only ADCA0 supports this function.

29.3.2.10 ADCAnTHSMPSTCR — T&H Sampling Start Control Register

This register is 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 can be written only in 32-bit units.
ADCAnTHSMPSTCRL can be written only in 16-bit units.
ADCAnTHSMPSTCRLL can be written only 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	R

Table 29.23 ADCAnTHSMPSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, 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

With the RH850/F1L, only ADCA0 supports this function.

29.3.2.11 ADCAnTHCR — T&H Control Register

This register controls the sampling transition after the T&H circuit is held.

Automatic start of sampling on the T&H circuit after the hold is performed 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 29.24 ADCAnTHCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When written, 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

With the RH850/F1L, only ADCA0 supports this function.

29.3.2.12 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 can be written only in 32-bit units.
ADCAnTHAHLDSTCRL can be written only in 16-bit units.
ADCAnTHAHLDSTCRLL can be written only 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	R

Table 29.25 ADCAnTHAHLDSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, 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

With the RH850/F1L, only ADCA0 supports this function.

29.3.2.13 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 can be written only in 32-bit units.
ADCAnTHBHLDSTCRL can be written only in 16-bit units.
ADCAnTHBHLDSTCRLL can be written only 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	R

Table 29.26 ADCAnTHBHLDSTCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, 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 ADCAnTHAHLDSTCR.HLDST while ADCAnTHER.THkE = 1 (k = 0 to 5) and ADCAnTHGSR.THkGS = 0 (k = 0 to 5).

NOTE

With the RH850/F1L, only ADCA0 supports this function.

29.3.2.14 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 29.27 ADCAnTHACR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When written, write the value after reset.
5	HLDCTE	This bit is used when self-diagnosis of the T&H circuit is to proceed. 0: Self-diagnosis proceeds. 1: Self-diagnosis does not proceed. 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 written, 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

With the RH850/F1L, only ADCA0 supports this function.

29.3.2.15 ADCAnTHBCR — T&H Group B Control Register

This register is a 32/16/8-bit read/write register, which controls 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 29.28 ADCAnTHBCR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When written, write the value after reset.
5	HLDCTE	This bit is used when self-diagnosis of the T&H circuit is to proceed. 0: Self-diagnosis proceeds. 1: Self-diagnosis does not proceed. 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 ADCAnTHBCR 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 written, write the value after reset.
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

With the RH850/F1L, only ADCA0 supports this function.

29.3.2.16 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 29.29 ADCAnTHER Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When written, 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

With the RH850/F1L, only ADCA0 supports this function.

29.3.2.17 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 29.30 ADCAnTHGSR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When written, 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

With the RH850/F1L, only ADCA0 supports this function.

29.3.2.18 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 29.31 ADCAnSMPCR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When written, 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 (ADCACLK = 8 MHz to 32 MHz) 18_H: 24 cycles (ADCACLK = 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).

29.3.2.19 ADCAnSFTCR — Safety Control Register

This register is a register regarding 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 29.32 ADCAnSFTCR Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When read, the value after reset is returned. When written, 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: ADCAnPWDDR/ADCAnDRj and ADCAnPWDDIR/ADCAnDIRj are not cleared when ADCAnPWDDR/ADCAnDRj or ADCAnPWDDIR/ADCAnDIRj is read. 1: ADCAnPWDDR/ADCAnDRj and ADCAnPWDDIR/ADCAnDIRj are cleared when ADCAnPWDDR/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 written, 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.

29.3.2.20 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/written in 32-bit units.
ADCAnULLMTBR0L to 2L and ADCAnULLMTBR0H to 2H can be read/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 29.33 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[15:0] < A/D converted value
19 to 16	Reserved	When read, the value after reset is returned. When written, 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 written, write the value after reset.

CAUTION

- When AD 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.

29.3.2.21 ADCAnECR — Error Clear Register

This register is used to control clearing of an error. The read value is always 0.

Access: ADCAnECR can only be read or written in 32-bit units.
ADCAnECRL can only be read or written in 16-bit units.
ADCAnECRLL can only be read or 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 29.34 ADCAnECR Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When written, write the value after reset.
3	ULEC	Upper Limit Error Flag (ADCAnULER.UE) / Lower Limit Error Flag (ADCAnULER.LE) Clear 0: No effect. 1: Clears the flag.
2	OWEC	Overwrite Error Flag (ADCAnOWER.OWE) Clear 0: No effect. 1: Clears the flag.
1, 0	Reserved	When written, write the value after reset.

29.3.2.22 ADCAnULER — Upper Limit/Lower Limit Error Register

This register is a read-only register, which indicates information regarding the upper limit/lower limit errors.

Access: ADCAnULER can only be read in 32-bit units.
ADCAnULERL can only be read in 16-bit units.

ADCAnULERLH can only be read in 8-bit units.
ADCAnULERLL can only 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 29.35 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 ADCRnULER 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 ADCRnULER 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: A scan group where an upper limit/lower limit error occurred is SG1 to SG3. 10: A scan group where an upper limit/lower limit error occurred is PWM-Diag.
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.
10 to 8	MPXV[2:0]*1	The value of MPX is stored when the errors of the upper and lower limit occurred

Table 29.35 ADCAnULER Register Contents (2/2)

Bit Position	Bit Name	Function
7	Reserved	This bit is read as an undefined value.
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: While UE = 0 and UL = 0, the A/D converted value exceeds the range of the specified upper limit/lower limit table range. Clearing condition: 1 is written to ADCAnECR.ULEC.

Note 1. These bits are only supported by ADCA0.
With ADCA1, reading these bits returns the value after a reset.

NOTE

ADCAnULER is updated when the A/D converted value is set in ADCAnDRj or ADCAnPWDTSNDR.

29.3.2.23 ADCAnOWER — Overwrite Error Register

This register is a 32/16/8-bit read-only register, which indicates an overwrite error. The target of an overwrite error is SG1 to SG3, and not PWM-Diag.

Access: ADCAnOWER can only be read in 32-bit units.
ADCAnOWERL can only be read in 16-bit units.
ADCAnOWERLL can only 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 29.36 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 ADCRnOWER 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.

29.3.3 Scan Group (SG) Unique Registers

This section describes the registers provided for each scan group.

29.3.3.1 ADCAnSGSTCRx — Scan Group x Start Control Register

This register is used to control the start of scan group x. The read value is always 0.

Access: ADCAnSGSTCRx can only be read in 32-bit units.
ADCAnSGSTCRxL can only be read in 16-bit units.
ADCAnSGSTCRxLL can only be read 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 29.37 ADCAnSGSTCRx Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, 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.

29.3.3.2 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 29.38 ADCAnSGCRx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When written, 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 written, 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.

CAUTION

To prevent malfunction, ADCAnSGCRx should be set (except clearing TRGMD upon completion of AD conversion) when SGACT of all scan groups are 0 (before the scan group is started) and TRGMD of all scan group is 0.

29.3.3.3 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 29.39 ADCAnPWDSGCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When written, write the value after reset.
0	PWDTRGMD	PWM-Diag Trigger Mode Select 0: PVCR_TRG trigger input is disabled. 1: PVCR_TRG 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).

29.3.3.4 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 29.40 ADCAnSGVCSPx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When written, 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 SGn is started next time.
- When the hardware trigger is used, writing to this register during operation is prohibited.

29.3.3.5 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 29.41 ADCAnSGVCEPx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, the value after reset is returned. When written, 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.

29.3.3.6 ADCAnSGMCYCRx — Scan Group x Multicycle Register

This register is a 32/16/8-bit read/write register, which 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 29.42 ADCAnSGMCYCRx Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned. When written, 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.

29.3.3.7 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 can be written only in 32-bit units.
ADCAnPWDSGSEFCRL can be written only in 16-bit units.
ADCAnPWDSGSEFCRLL can be written only 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	R

Table 29.43 ADCAnPWDSGSEFCR Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the value after reset.
0	PWDSEFC	PWM-Diag Scan End Flag Clear Trigger 0: No effect. 1: The PWM-Diag scan end flag (ADCAnSGSTR.SEF[4])

29.3.3.8 ADCAnSGSEFCRx — Scan Group x Scan End Flag Clear Register

This register is a write-only register, which controls the scan end flag (ADCAnSGSTR.SEFx) The read value is always 0.

Access: ADCAnSGSEFCRx can only be written in 32-bit units.
ADCAnSGSEFCRxL can only be written in 16-bit units.
ADCAnSGSEFCRxLL can only 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 29.44 ADCAnSGSEFCRx Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, 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]).

29.3.3.9 ADCAnSGSTR — Scan Group x Status Register

This register indicates the state of T&H, SVSTOP and scan group x, and PWM-Diag scan group. The SHACT bits are cleared when HALT is executed.

Access: ADCAnSGSTR can only be read in 32-bit units.
ADCAnSGSTRL can only 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 29.45 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 29.45 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.

29.3.4 Hardware Trigger Specific Register

29.3.4.1 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 a reset.

Table 29.46 ADCAnSGTSELx Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, the value after reset is returned. When written, 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.

The list below shows the hardware triggers to be selected.

Table 29.47 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 29.47 List of A/D Conversion Hardware Triggers (2/2)

	Control Register/Bit		Trigger Input Signal	
Unit	**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

CAUTION

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.

29.3.5 Self-Diagnosis Specific Registers

29.3.5.1 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 29.48 ADCAnDGCTL0 Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When written, 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}	$2/3AnV_{REF}$	$2/3AnV_{REF}$	$2/3AnV_{REF}$

29.3.5.2 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 29.49 ADCAnDGCTL1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, 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.

29.3.5.3 ADCAnPDCTL1 — Pull Down Control Register 1

This register specifies a channel which the pull down resistor is connected with.

For details, see **Section 29.5.3, 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/W	R/W	R/W	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
	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

Table 29.50 ADCAnPDCTL1 Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, 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.

29.3.5.4 ADCAnPDCTL2 — Pull Down Control Register 2

This register specifies a channel which the pull down resistor is connected with.

For details, see **Section 29.5.3, 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. ADCA1 supports only PDNB[7:0]. When writing to the other bits, write the value after a reset.

Table 29.51 ADCAnPDCTL2 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When read, the value after reset is returned. When written, 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 (ANI0[16:35], ANI1[16:23]). 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.

29.3.6 Emulation Specific Register

29.3.6.1 ADCAnEMU — Emulation Control Register

This register controls the SVSTOP disable signal.

Access: This register can be read/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 29.52 ADCAnEMU Register Contents

Bit Position	Bit Name	Function
7	SVSDIS	SVSTOP Disable 0: SVSTOP is enabled 1: SVSTOP is disabled For the AD conversion when SVSTOP is enabled, see **Section 29.4.10.3, SVSTOP Operation**.
6 to 0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

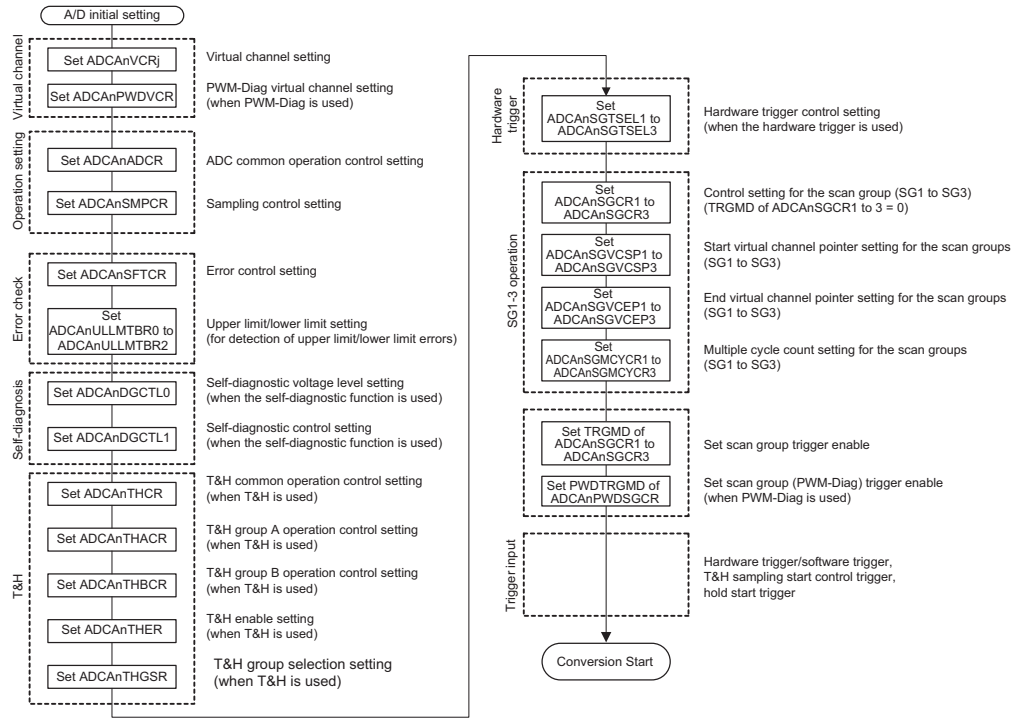
CAUTION

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.

29.4 Operation

29.4.1 Initial Setting

Figure 29.6 shows an initial setting example of the A/D conversion. For trigger input, see **Figure 29.7** in the next section. For interrupt request signals, see **Section 29.4.12, Scan End Interrupt Request**.



Note: Set these registers as necessary.

Figure 29.6 Flowchart for Initial Setting

29.4.2 Trigger Input

The following figure shows the flowchart for trigger input.

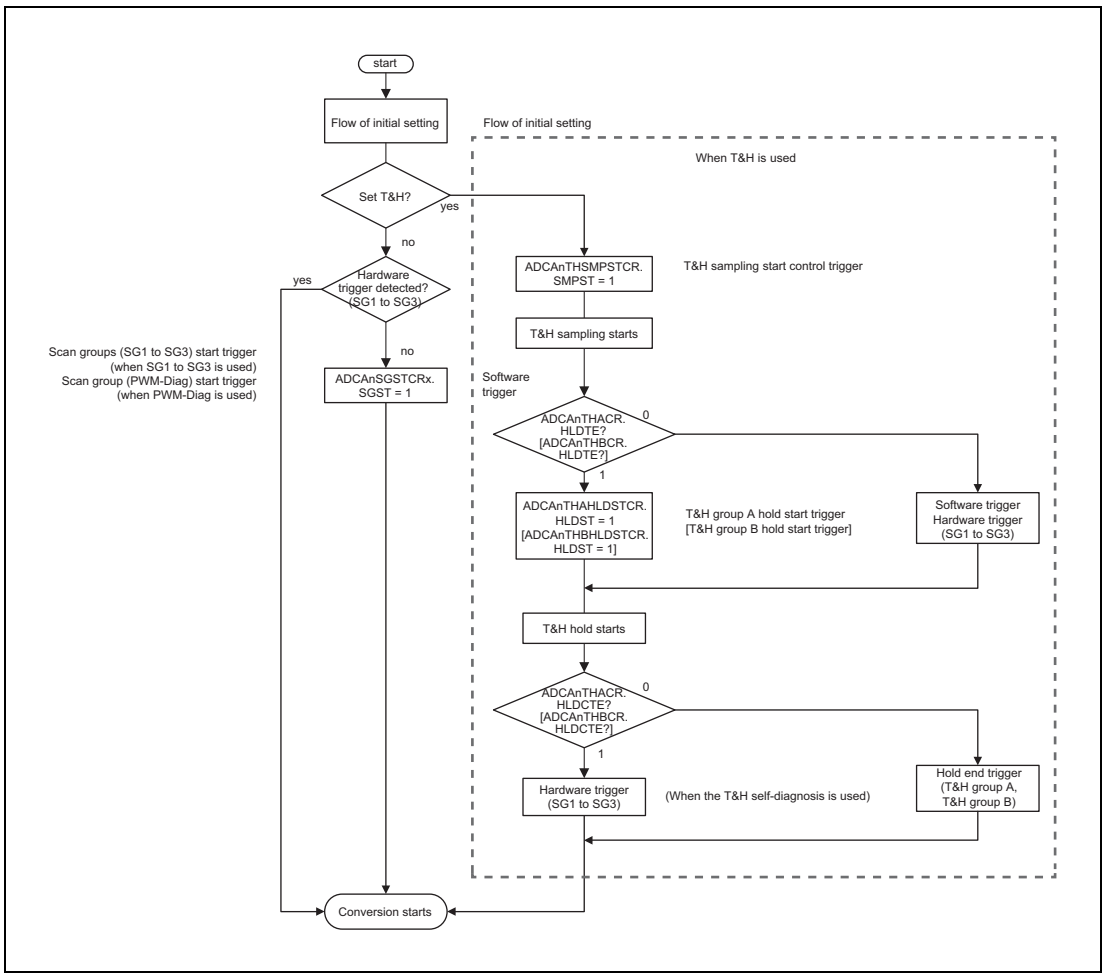


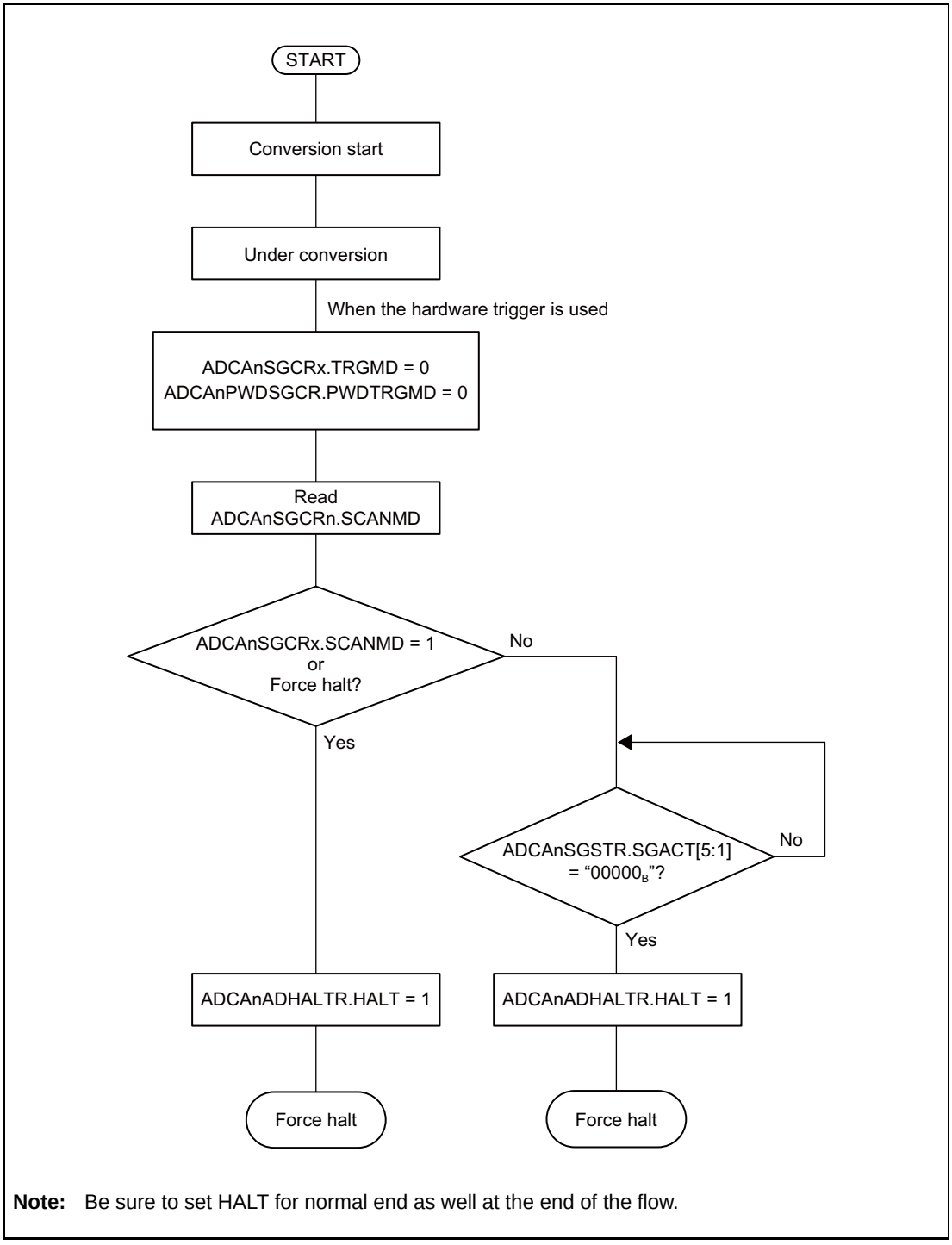
Figure 29.7 Flowchart for Trigger Input

NOTE

When an SG start trigger is generated during scanning, the SG start trigger is ignored.

29.4.3 Ending A/D Conversion

The flow for ending the A/D conversion is shown below.



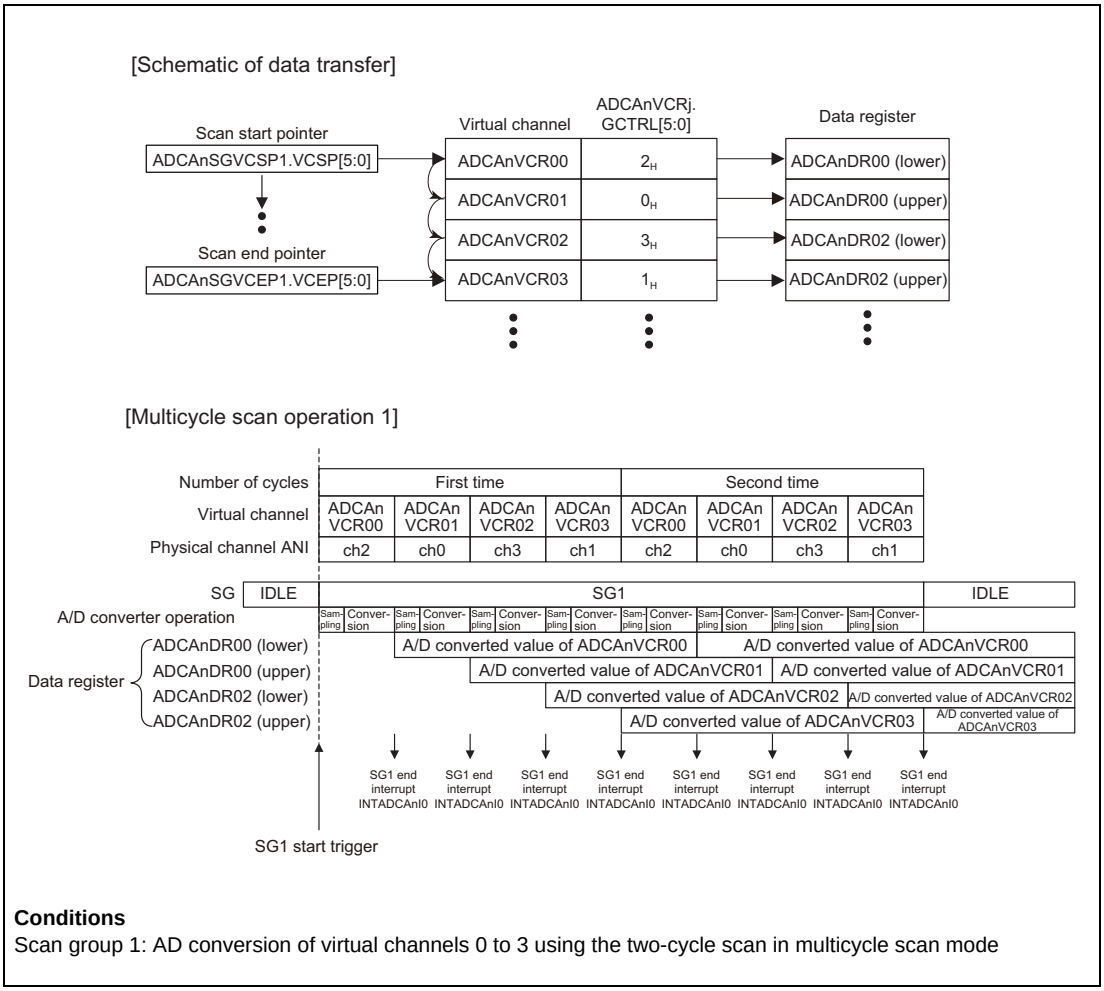
Note: Be sure to set HALT for normal end as well at the end of the flow.

Figure 29.8 Flowchart for Ending A/D Conversion

29.4.4 Example of Scan Group Operation

(1) Multicycle scan mode

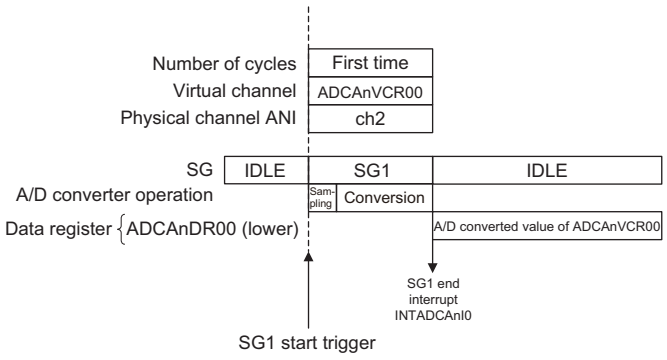
The following figure illustrates an operation example where four virtual channels of scan group1 are converted using the two-cycle scan in multicycle scan mode.



Conditions
Scan group 1: AD conversion of virtual channels 0 to 3 using the two-cycle scan in multicycle scan mode

Figure 29.9 Example of Multicycle Scan Operation 1

The following figure illustrates an operation example where a pin is scanned once, using multicycle scan mode.

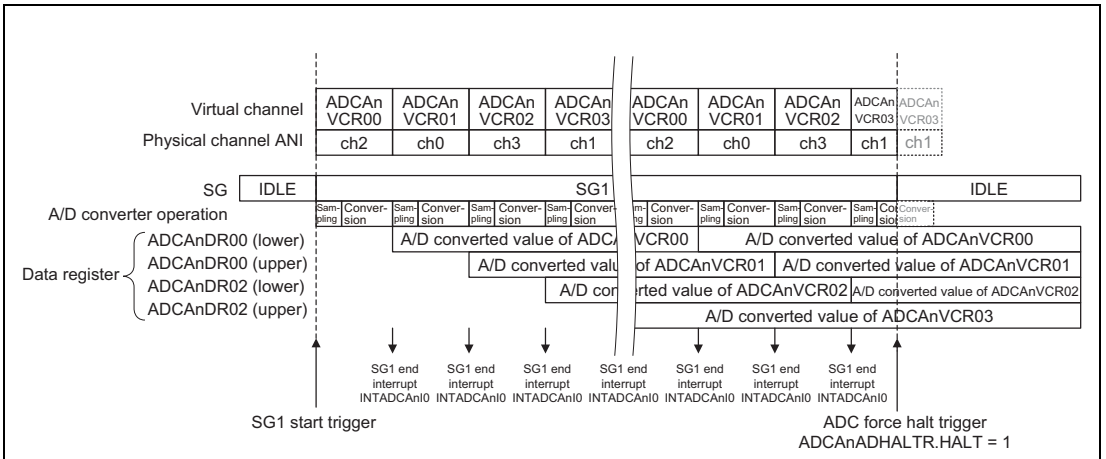


Conditions
Scan group 1: AD conversion of virtual channel 0 in which a pin is scanned once in multicycle scan mode

Figure 29.10 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 operates exclusively in each SG. The following figure shows an example of operation in continuous scan mode.



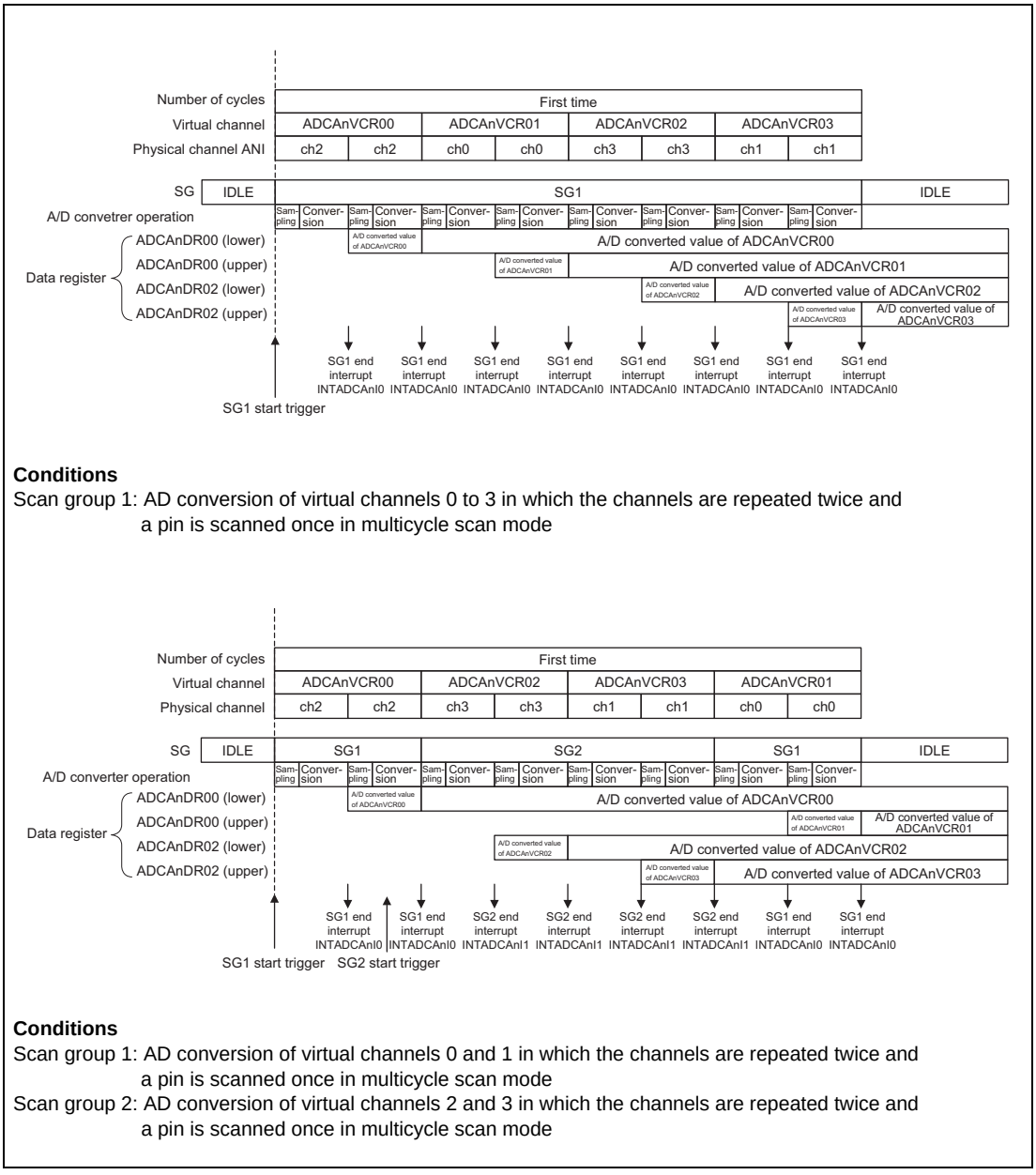
Conditions
Scan group 1: AD conversion of virtual channels 0 to 3 in continuous scan mode

Figure 29.11 Example of Continuous Scan Operation

29.4.5 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.



Conditions

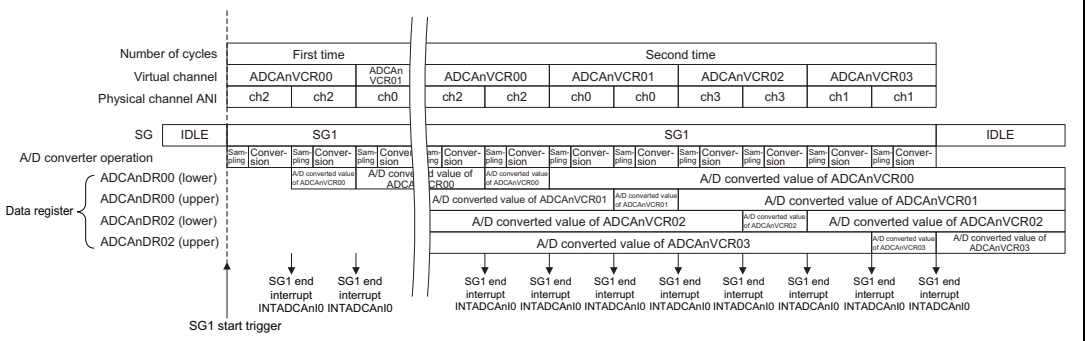
Scan group 1: AD conversion of virtual channels 0 to 3 in which the channels are repeated twice and a pin is scanned once in multicycle scan mode

Conditions

Scan group 1: AD conversion of virtual channels 0 and 1 in which the channels are repeated twice and a pin is scanned once in multicycle scan mode

Scan group 2: AD conversion of virtual channels 2 and 3 in which the channels are repeated twice and a pin is scanned once in multicycle scan mode

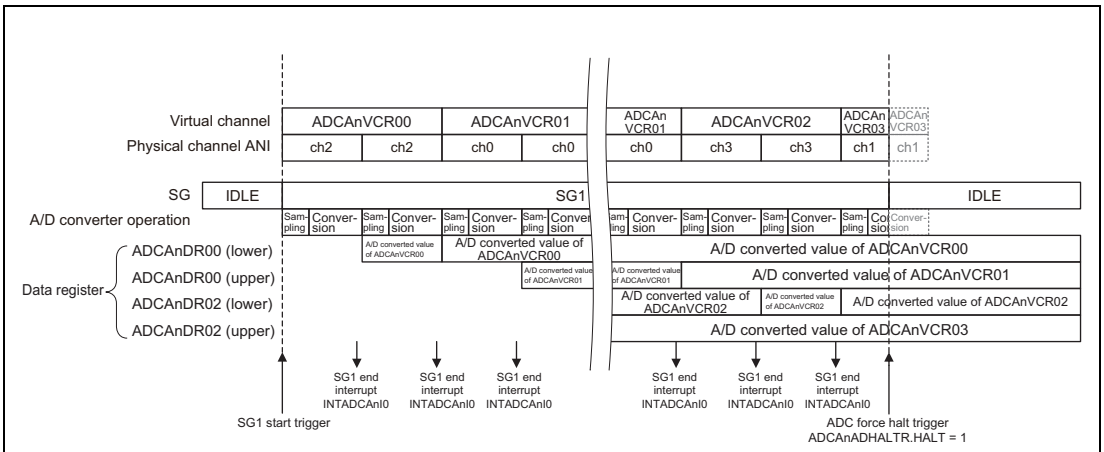
Figure 29.12 Example of Channel Repeat Operation 1



Conditions

Scan group 1: AD 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 29.13 Example of Channel Repeat Operation 2



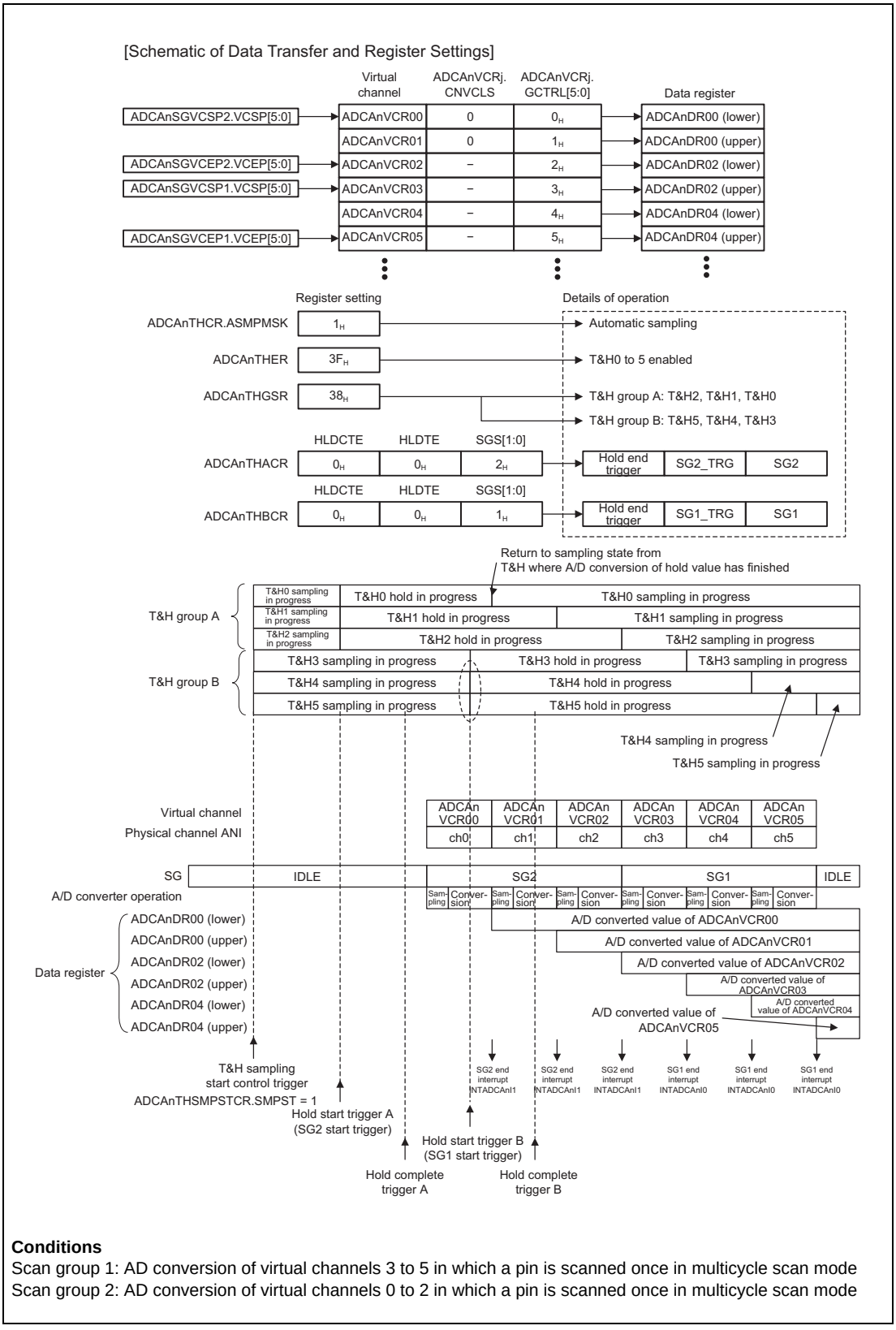
Conditions

Scan group 1: AD conversion of virtual channels 0 to 3 in which the channels are repeated twice in multicycle scan mode

Figure 29.14 Example of Channel Repeat Operation 3

29.4.6 Example of Simultaneous Track and Hold Operation

Figure 29.15 shows an operation example of simultaneous track and hold.



Conditions
Scan group 1: AD conversion of virtual channels 3 to 5 in which a pin is scanned once in multicycle scan mode
Scan group 2: AD conversion of virtual channels 0 to 2 in which a pin is scanned once in multicycle scan mode

Figure 29.15 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 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 are set and HLDST 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 before setting HLDCTE = 1 and HLDTE = 1 and 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 and completion of the group AD conversion to be 10 µs or less.**

29.4.7 A/D Conversion with External Analog Multiplexer

The following figures show examples of A/D conversion in each case.

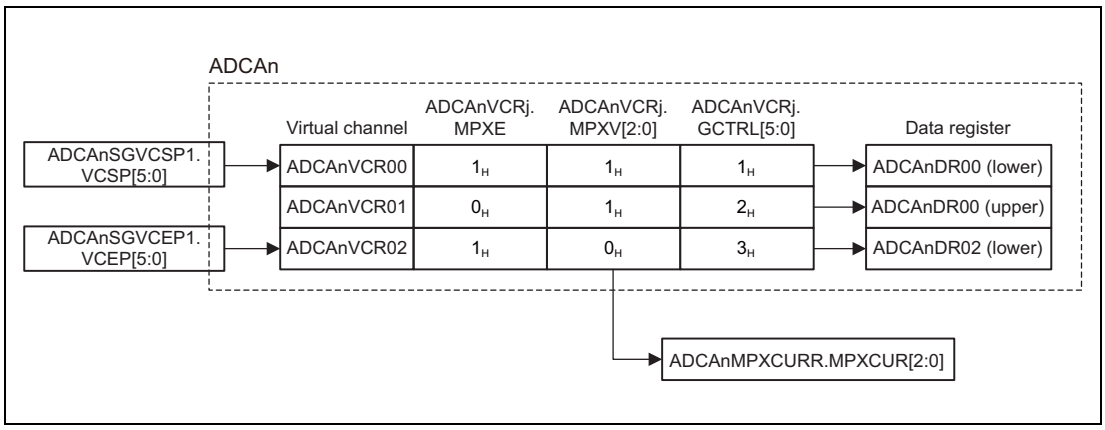
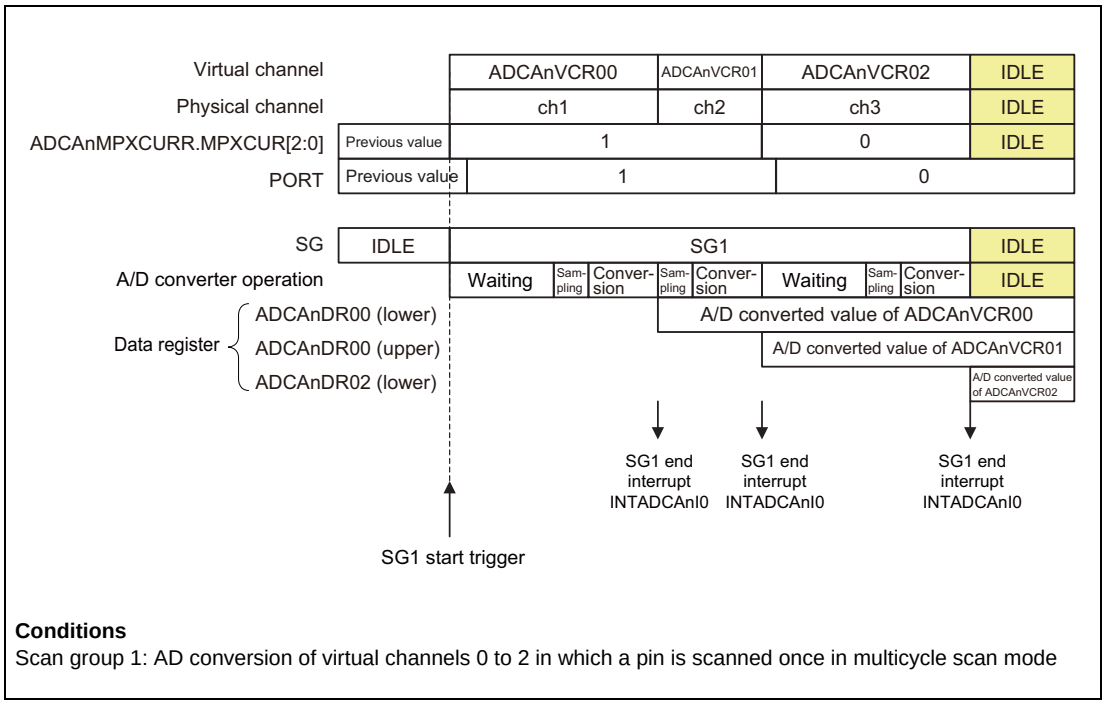
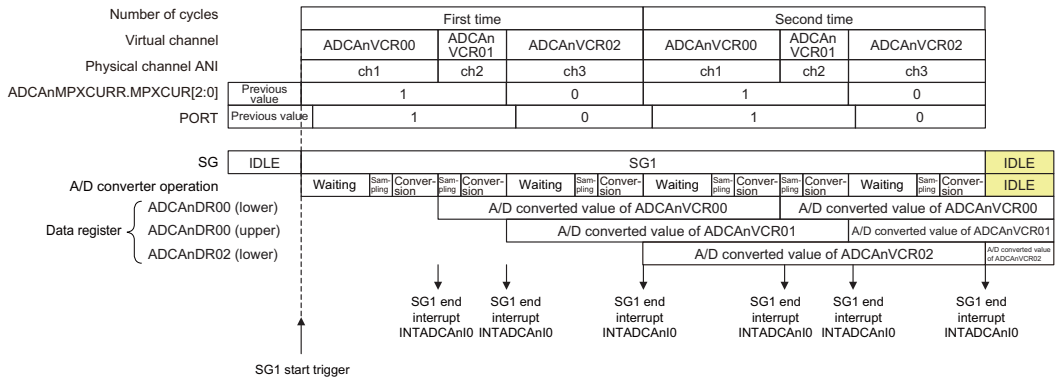


Figure 29.16 Schematic of Data Transfer and Register Settings



Conditions
Scan group 1: AD conversion of virtual channels 0 to 2 in which a pin is scanned once in multicycle scan mode

Figure 29.17 A/D Conversion 1 at an External Analog Multiplexer

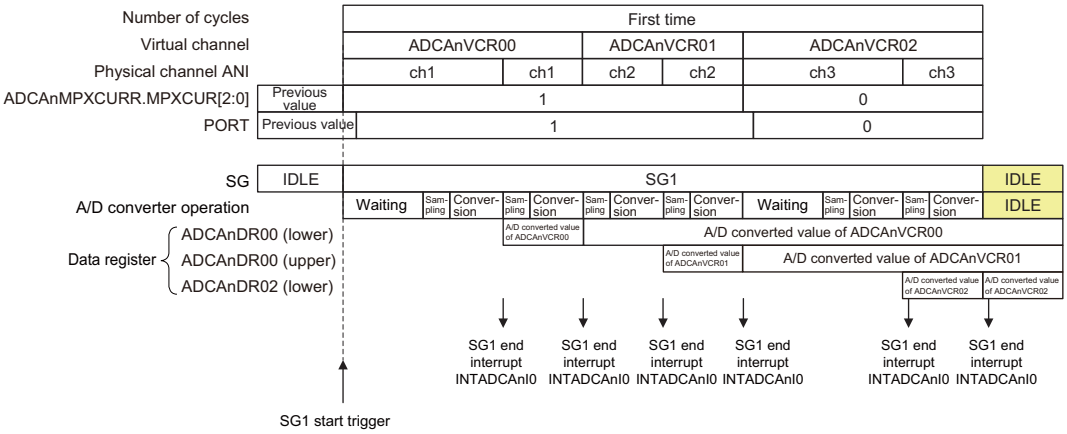


Conditions

Scan group 1: AD 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 29.18 A/D Conversion 2 at an External Analog Multiplexer



Conditions

Scan group 1: AD 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 29.19 A/D Conversion 3 at an External Analog Multiplexer

29.4.7.1 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 28, 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 PVCR_TRG 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 PVCR_TRG 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 29.20 shows an example of PWM-Diag operation using an MPX.

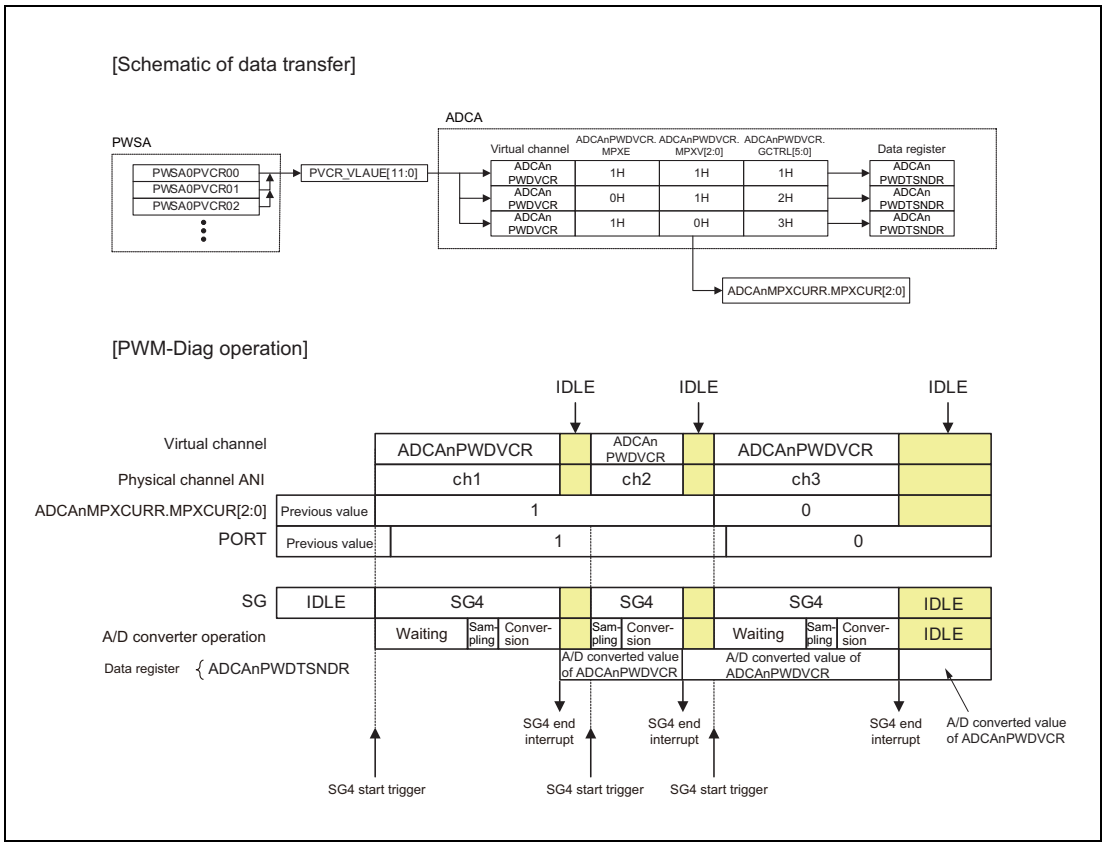


Figure 29.20 PWM-Diag Operation

29.4.8 Example of Synchronous Suspend and Resume Operation

Figure 29.21 shows an example of synchronous suspend and resume operation. In this example, a higher-priority SG interrupts a lower-priority SG.

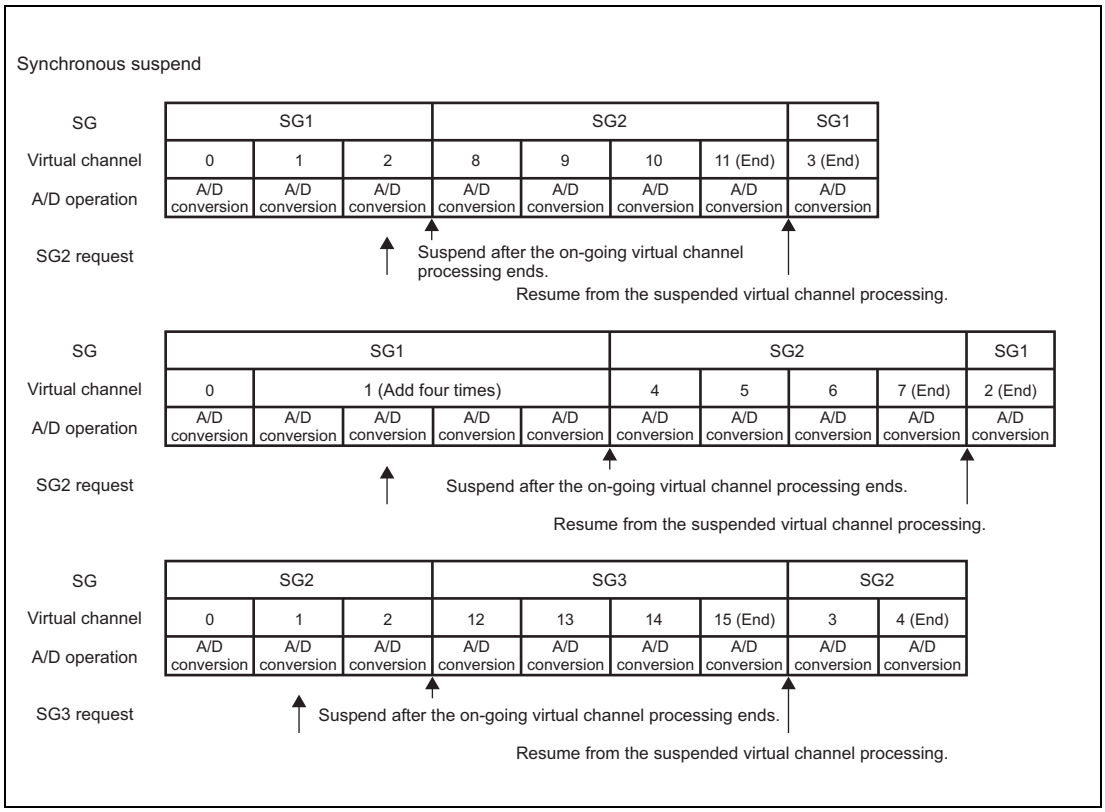


Figure 29.21 Example of Synchronous Suspend and Resume Operation

NOTE

Priority of scan groups is as follows.

Lower Higher

SG1 < SG2 < SG3 < PWM-Diag (SG4)

29.4.9 Example of Asynchronous Suspend and Resume Operation

Figure 29.22 shows an example of asynchronous suspend and resume operation. In this example, a higher-priority SG interrupts a lower-priority SG.

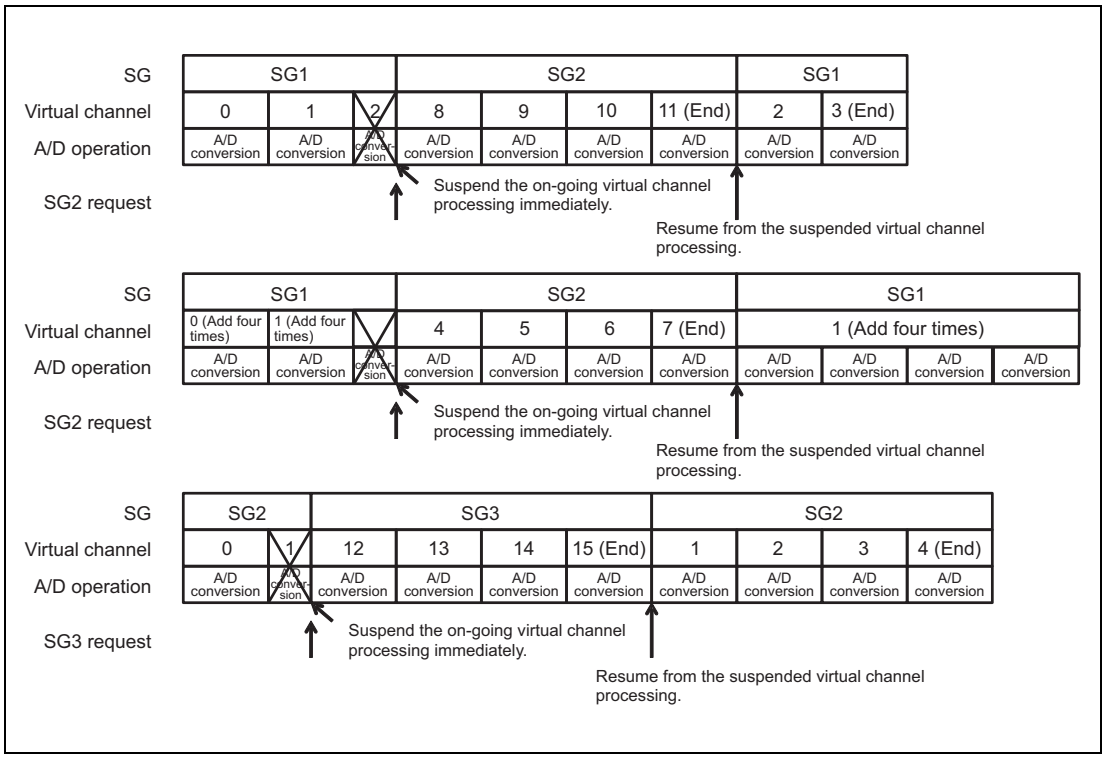


Figure 29.22 Example of Asynchronous Suspend and Resume Operation

NOTE

Priority of scan groups is as follows.

Lower Higher

SG1 < SG2 < SG3 < PWM-Diag (SG4)

29.4.10 Error Detecting Functions

ADCAn covers upper-limit error, lower-limit error, and overwrite error.

29.4.10.1 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.

29.4.10.2 Overwrite Error Detecting Function

If the ADCAnDIRj or ADCAnDRj of a virtual channel is not read while ADCAnDIRj.WFLG = 1 (AD converted value is stored) and the next AD converted value is written in the ADCAnDRj, an overwrite error is detected.

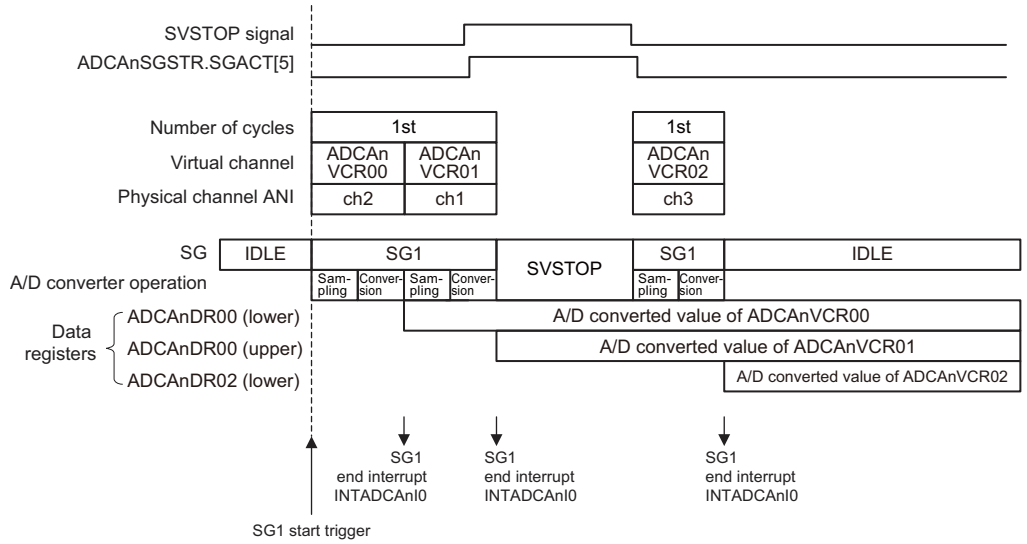
29.4.10.3 SVSTOP Operation

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.

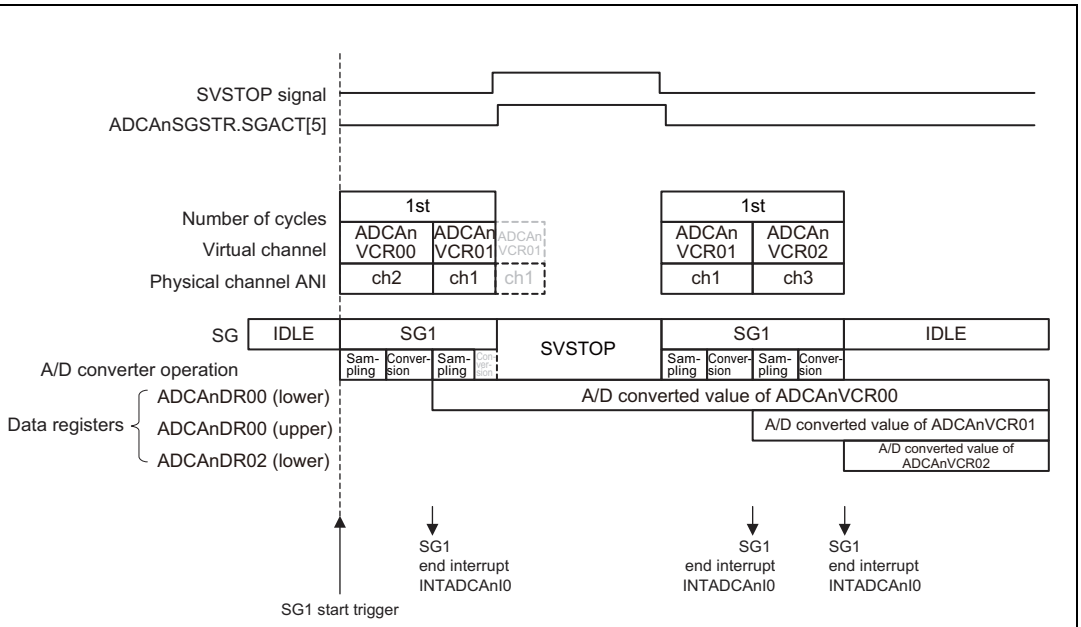
In operations for synchronous suspension, a new start trigger cannot be accepted over the time from when the high level is input to SVSTOP to the completion of conversion on the channel where conversion is currently proceeding. This time can be up to the time taken for one A/D conversion.

The following example illustrates a SVSTOP operation example.



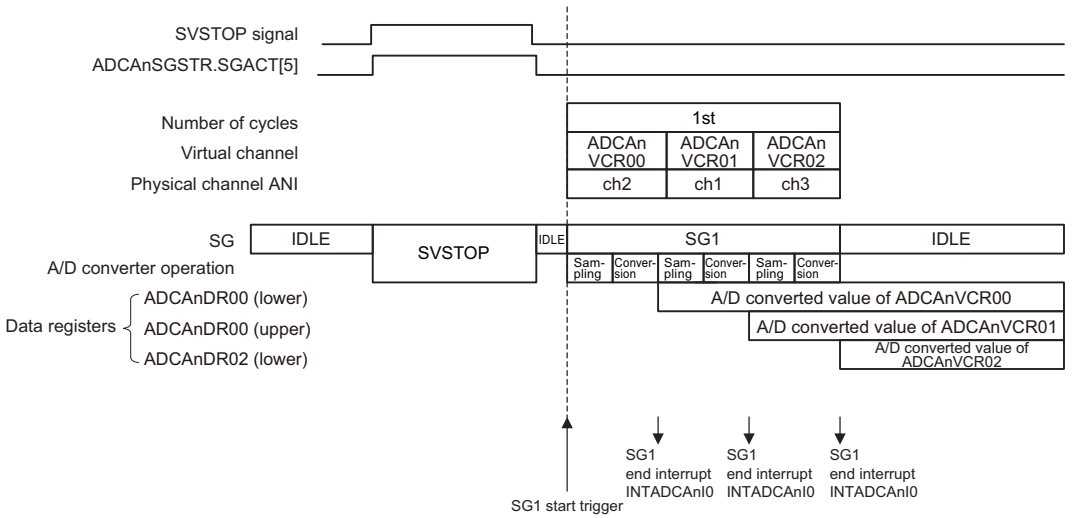
Note: An SG start trigger is accepted even during SVSTOP after the completion of conversion.

Figure 29.23 Example of SVSTOP Operation (ADCAnADCR.SUSMTD = 00 and ADCAnEMU.SVSDIS = 0)



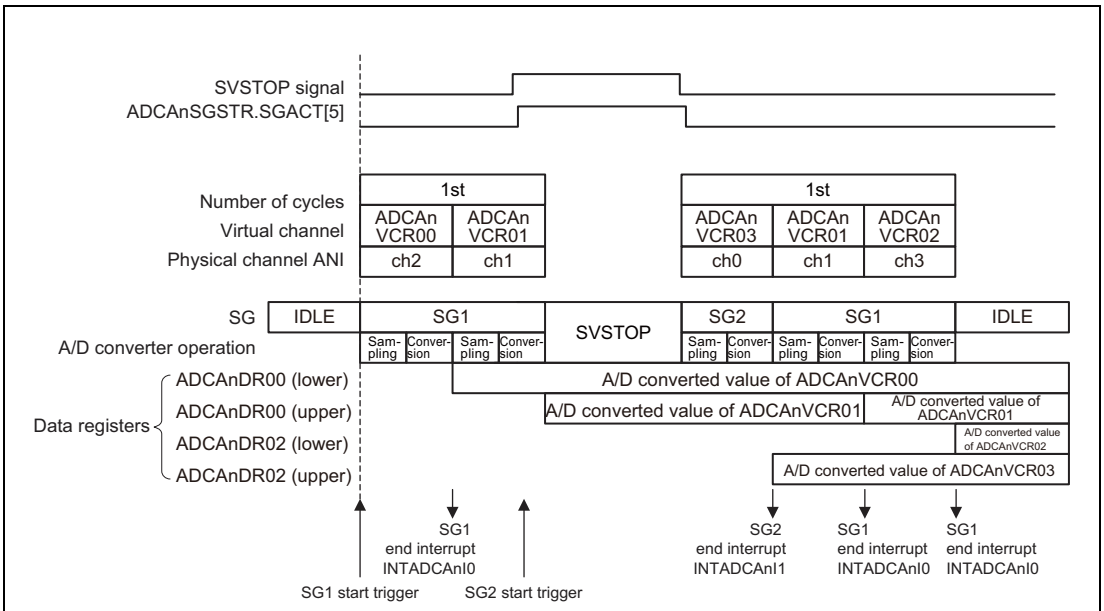
Note: An SG start trigger is accepted even during SVSTOP.

Figure 29.24 Example of SVSTOP Operation (ADCAnADCR.SUSMTD = 10 and ADCAnEMU.SVSDIS = 0)



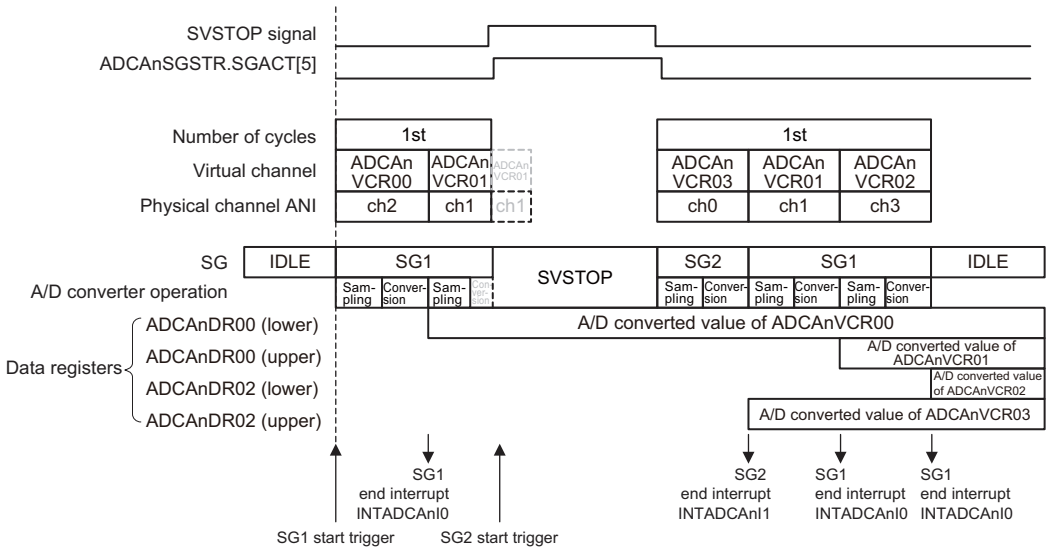
Note: An SG start trigger is accepted even during SVSTOP.

Figure 29.25 Example of SVSTOP Operation in the IDLE State (ADCAnADCR.SUSMTD = 00 and ADCAnEMU.SVSDIS = 0)



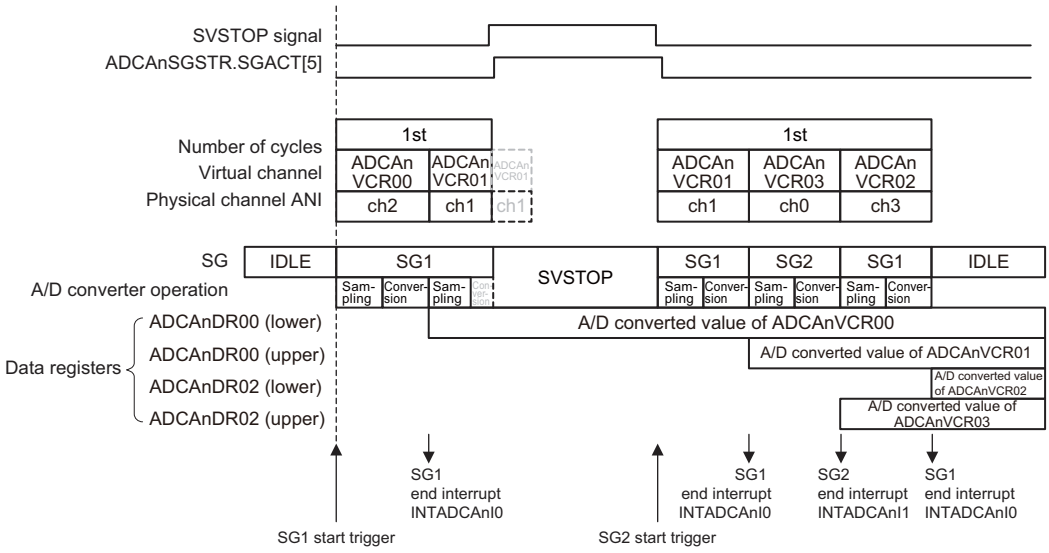
Note: Though the channel where conversion is currently proceeding is not stopped even after the acceptance of SVSTOP, an SG end interrupt is not generated.
This channel is converted again after SVSTOP has been cleared.
After the completion of conversion, the SG start trigger is accepted even during SVSTOP.

Figure 29.26 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 29.27 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 29.28 Conflict of SVSTOP Clear and High-Priority SG Start Trigger (ADCAnADCR.SUSMTD = 10, ADCAnEMU.SVSDIS = 0)

29.4.11 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 29.47, 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 the detail, see **Section Section 2, Pin Function**. More than one start trigger can be specified.

29.4.11.1 Stopping Scan Group by ADHALT

Setting ADCAnADHALTR.HALT (AD force halt trigger) to 1 forcibly halts the A/D conversion and clears the scan group x 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.

29.4.12 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 7, DMA**.

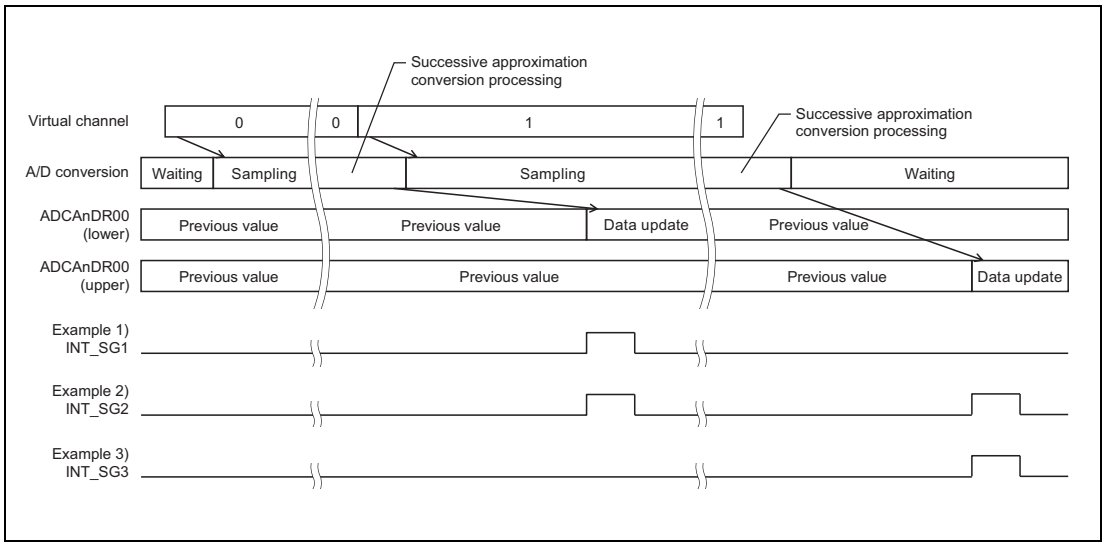


Figure 29.29 Example of a Scan Conversion End Interrupt Occurrence

29.4.13 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.

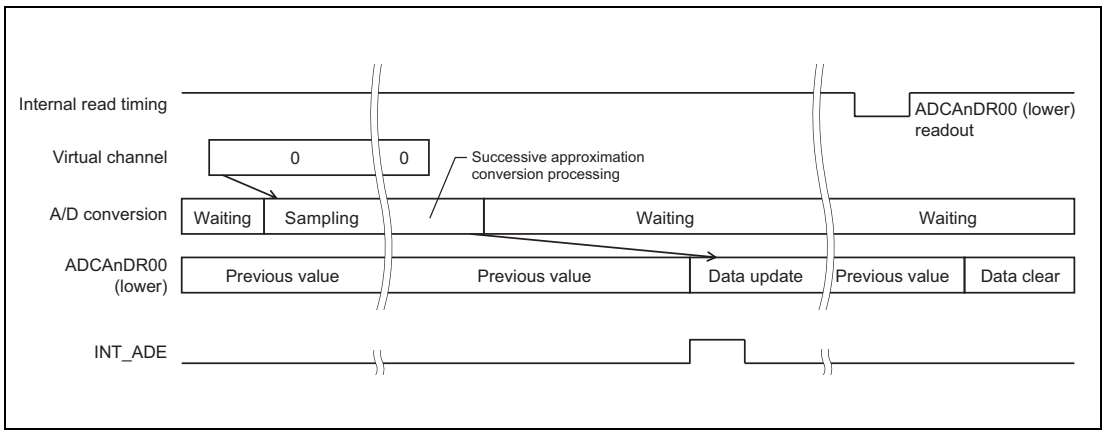


Figure 29.30 A/D Error Interrupt (Example: Overwrite Error)

29.5 Self-Diagnostic Function

To check the ADCAn function, the following self-diagnostic functions are available.

Section 29.5.1, Diagnosis of A/D Conversion Circuit

Section 29.5.2, Diagnosis of Channel Multiplexer

Section 29.5.3, Diagnosis of Open Pins

Section 29.5.4, 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.

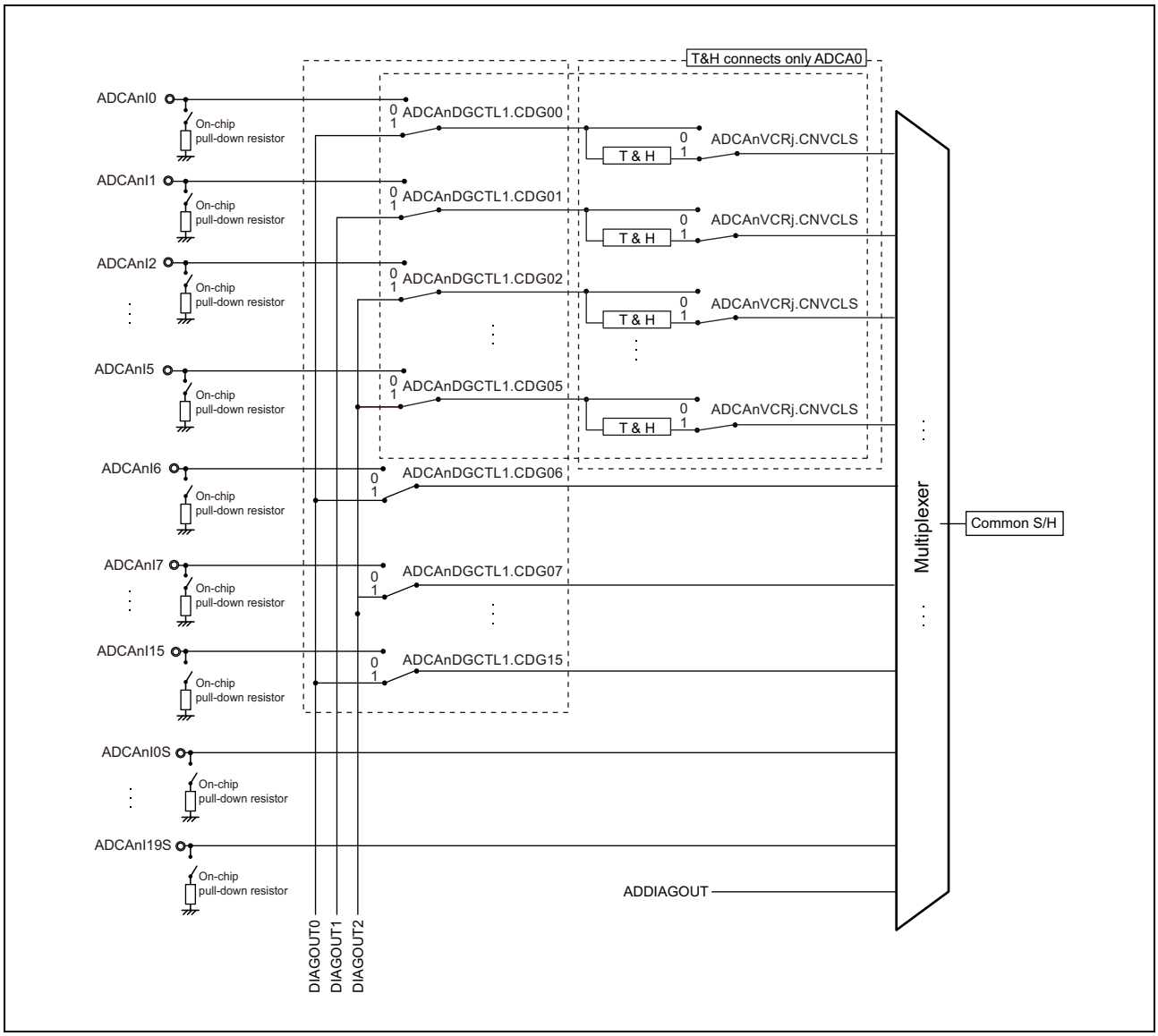


Figure 29.31 Overview of Self-Diagnostic Functions

NOTE

The functions in the dashed-line frames depend on the product.

29.5.1 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.

29.5.1.1 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.

29.5.2 Diagnosis of Channel Multiplexer

This function checks whether the path from the analog input to the analog 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 29.53 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.

29.5.2.1 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. 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.

29.5.3 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.

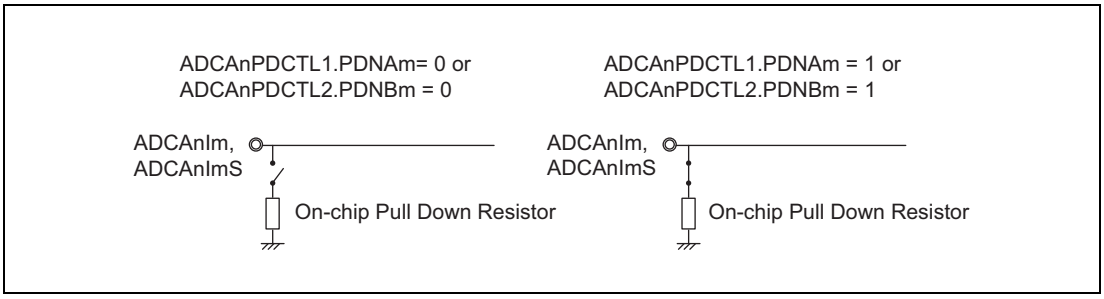


Figure 29.32 Setting of On-chip Pull Down Resistor

When there is a disconnection, the conversion result is almost 0 V and it indicates an open detection.

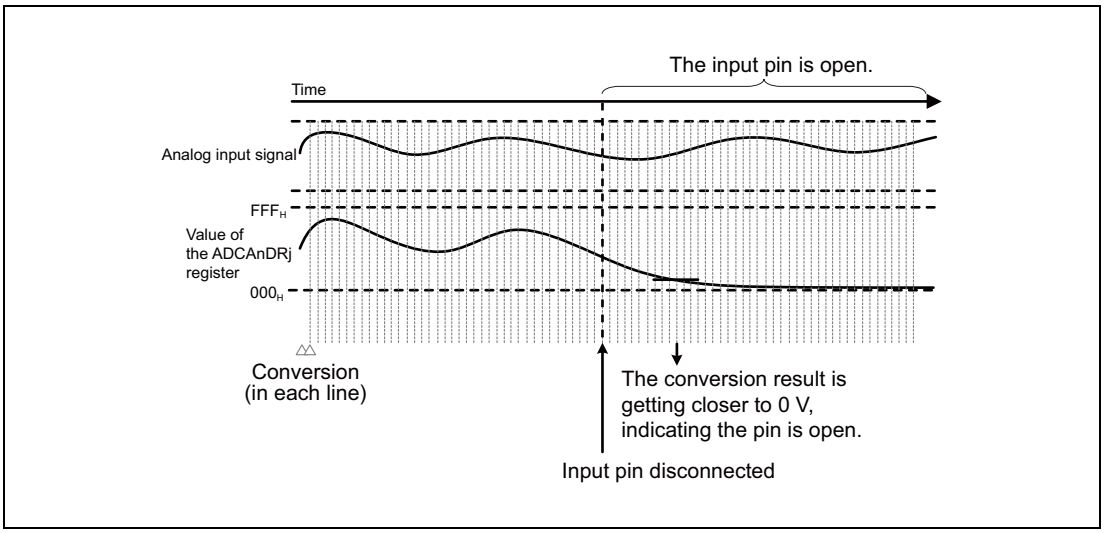


Figure 29.33 Analog Input Signal Disconnection Detection

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.**

29.5.3.1 Diagnostic procedure

1. Set PDNA bits of the ADCAnPDCTL1 or ADCAnPDCTL2 register 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.

29.5.4 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 registers 33 to 35 (ADCA0VCR33 to 35) are used exclusively for comparison of the potential conversion result using the T&H circuit and the one without using the T&H circuit to detect a failure of the T&H circuit.

For this diagnosis, with the A/D conversion trigger set to the hold end trigger (ADCA0THACR.HLDCTE = 1) and the hold start trigger (ADCA0THACR.HLDTE = 0), 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.

29.5.4.1 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.

29.5.4.2 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.

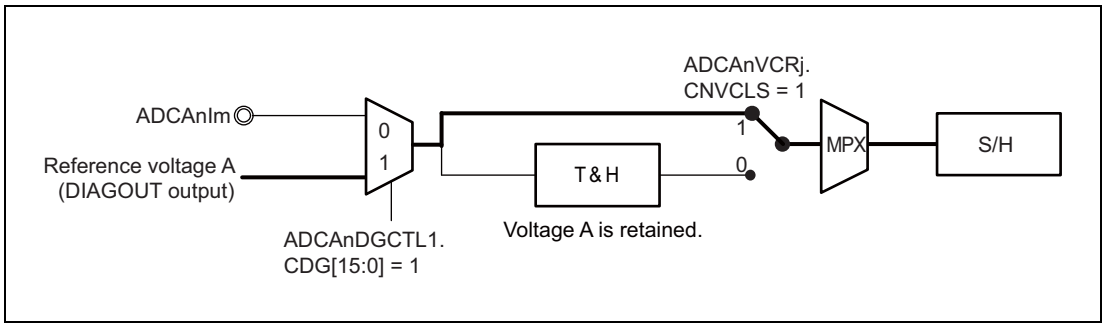


Figure 29.34 T&H Circuit Diagnostic Mechanisms (1)

(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.

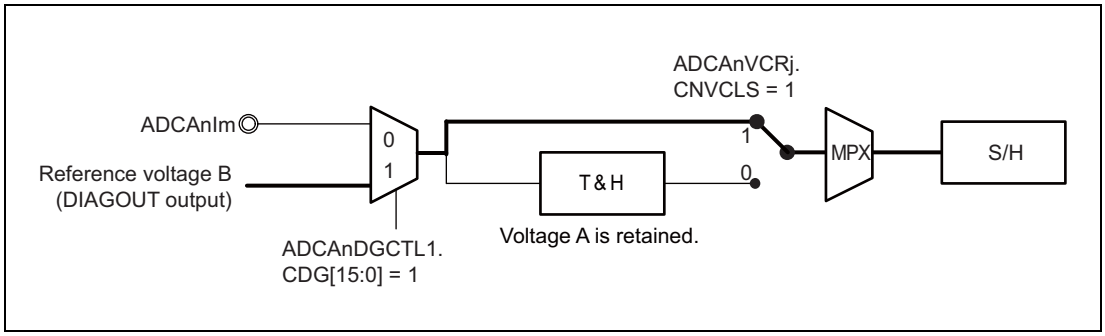


Figure 29.35 T&H Circuit Diagnostic Mechanisms (2)

(3) An A/D conversion is performed using the T&H circuit. The T&H circuit continues to hold the voltage A.

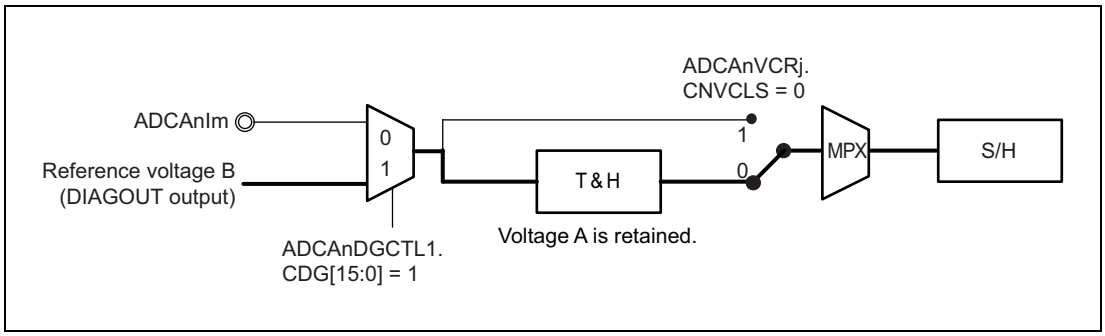


Figure 29.36 T&H Circuit Diagnostic Mechanisms (3)

(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.

29.6 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 29.37**).
- 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.

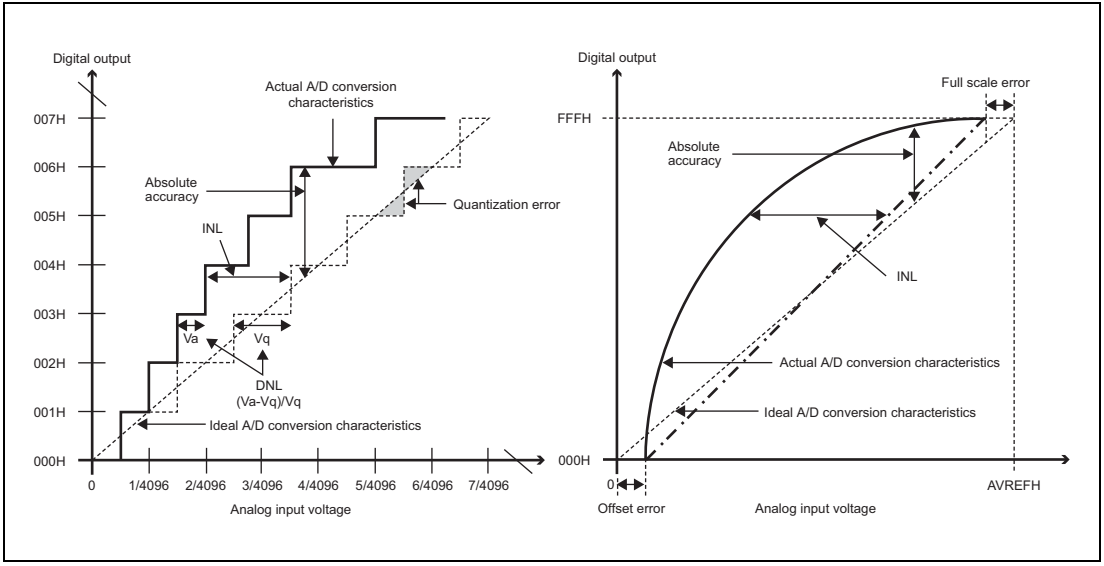


Figure 29.37 Definition of A/D Conversion Accuracy

29.7 Usage Notes

29.7.1 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.

29.7.2 Notes on Application Design

(1) Analog input pins (ADCAnIm, ADCAnImS)

- Ensure that the 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 AD conversion.
- We recommend avoiding the driving of large currents through input and output pins near the ADCAnIm and ADCAnImS pins and particularly keeping toggled signals away from these pins.
- If you are using the standby functions, set the ADCAnADHALTR.HALT, ADCAnSGCRx, 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 (selecting the LPS). For details, see **Section 12, 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 AD 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 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 AD 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.

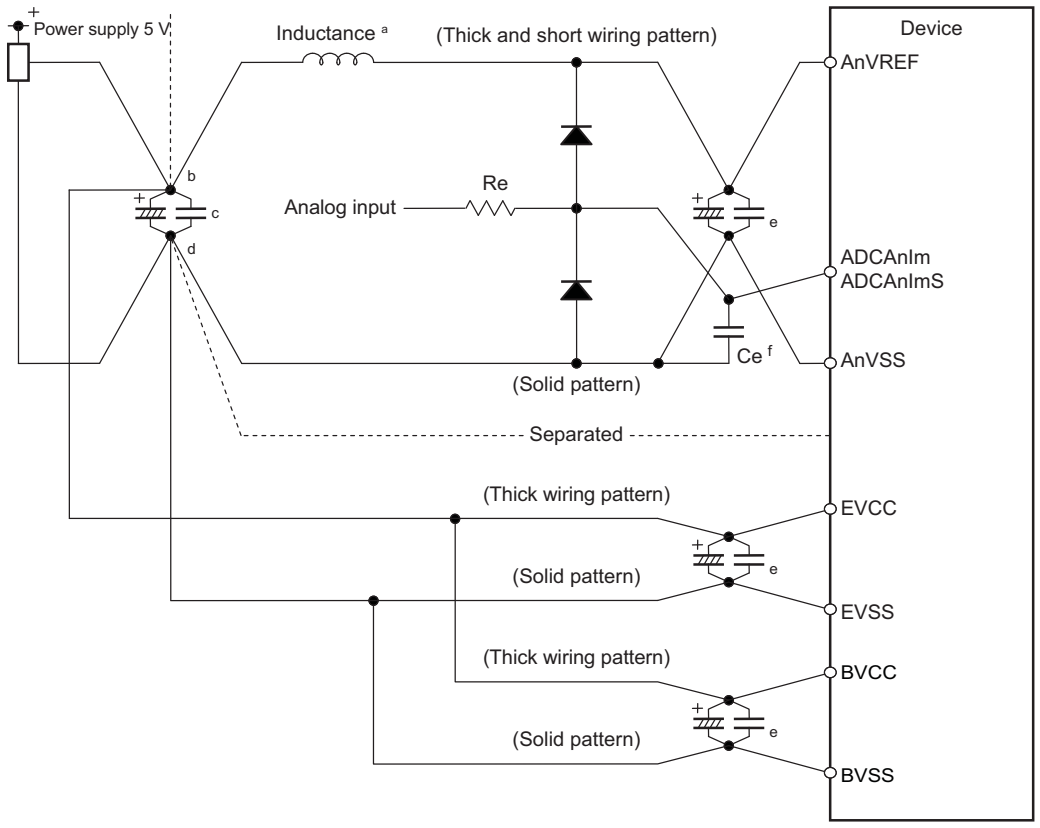
Examples of software handling are described below.

- Use averaged values from several rounds of AD conversion
- Execute AD 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.

(4) Alternative Input/Output

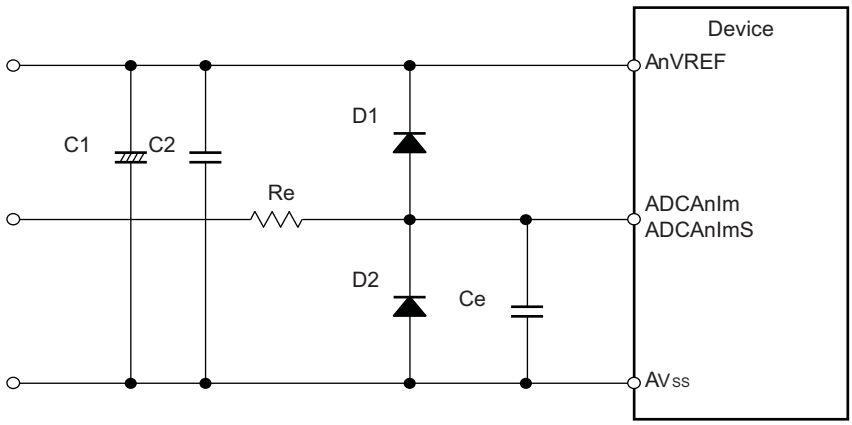
Analog input (ADCAnIm, ADCAnImS) pins can be used as port pins.

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 lower the precision of conversion.



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 various power supply pins.
f) Place capacitor Ce near the analog input pins of the device.

Figure 29.38 Example of Power Wiring



Note: C1: Product dependent
C2: Product dependent
Ce: Product dependent
Re: Product dependent
D1, D2: $V_F \leq 0.3$ V

Figure 29.39 Example of Noise Protection for Analog Input Circuit

Capacitor C1 is effective for low-frequency noise, and capacitors C2 and Ce are effective for high-frequency noise.

The voltage on an AnVREF pin is undefined immediately after switching from the stopped state to the start of conversion operations, and this may have the effect of reducing the accuracy of conversion. As a countermeasure for this situation, connect capacitors C1 and C2 to the AnVREF pins.

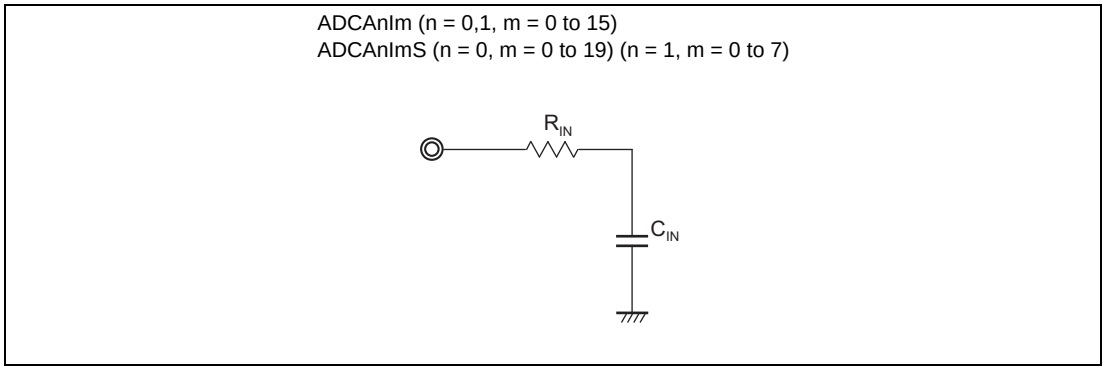


Figure 29.40 ADCAnIm, ADCAnImS Internal Equivalent Circuit

NOTE

The following table lists the reference values of circuit constants.

Pin	Condition	R_{IN} [kΩ]	C_{IN} [pF]
ADCA0I0-5	Channel T/H (Track & Hold) is in use	6.2	1.2
	Channel T/H (Track & Hold) is not in use	2.4	3.3
ADCA0I6-15		2.4	3.3
ADCA0I0S-3S, 5S-16S		3.6	8.4
ADCA0I4S, 17S-19S		5.4	8.4
ADCA1I0-15		2.4	3.8
ADCA1I0S-7S		3.0	6.2

Note 1. The above specification is not tested at the time of shipping inspection; that is, the values of R_{IN} and C_{IN} are not guaranteed. The values given above are defined as maximum values.

Section 30 Key Return (KR)

This section contains a generic description of the Key Return (KR) function.

The first part in this section describes the RH850/F1L 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.

30.1 Features of RH850/F1L KR

30.1.1 Number of Units and Channels

This microcontroller has the following number of KR units.

Table 30.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1					
Name	KRn (n = 0)					

The KR unit has the Key Return function of the following number of channels.

Table 30.2 KRn Unit Configurations and Channels

Unit Name (Channel Name) KRn	No. of Channels	RH850/F1L 64 pins (6 ch)	RH850/F1L 64 pins (8 ch)	RH850/F1L 80 pins (8 ch)	RH850/F1L 100 pins (8 ch)	RH850/F1L 144 pins (8 ch)	RH850/F1L 176 pins (8 ch)
KR0	8	—	√	√	√	√	√
	6	√	—	—	—	—	—

Table 30.3 Index

Index	Meaning
n	Throughout this section, the individual KR unit (Key Return Function unit) is identified by the index “n” (n = 0); for example, KRnKRM indicates the key return mode register.
m	Throughout this section, the individual KR channel is identified by the index “m” (m = 0 to 7); for example, KRnKRMm indicates the key input enable bit of KRnKRM (key return mode register).

30.1.2 Register Base Address

KRn base addresses are listed in the following table.

KRn register addresses are given as offsets from the base addresses.

Table 30.4 Register Base Address

Base Address Name	Base Address
<KR0_base>	FFF7 8000_H

30.1.3 Clock Supply

The KRn clock supply is shown in the following table.

Table 30.5 Clock Supply

Unit Name	Clock for the Unit	Supply Clock Name
KR0	PCLK	CPUCLK2

30.1.4 Interrupt Requests

KRn interrupt requests are listed in the following table:

Table 30.6 Interrupt Requests

Unit Interrupt Signal	Outline	Interrupt Number	DMA Trigger Number
KR0			
KRnTIKR	Key interrupt	82	—

30.1.5 Reset Sources

KRn reset sources are listed in the following table. KRn is initialized by these reset sources.

Table 30.7 Reset Sources

Unit Name	Reset Source
KR0	All reset sources (ISORES)

30.1.6 External Input/Output Signals

External input/output signals of KRn are listed below.

Table 30.8 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
KR0		
KRnTPKR7 to KRnTPKR0	Key input signal	KR0I7 to KR0I0

30.2 Overview

30.2.1 Functional Overview

The Key Return function has the following features:

A key interrupt request signal (KRnTIKR) 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).

30.2.2 Block Diagram

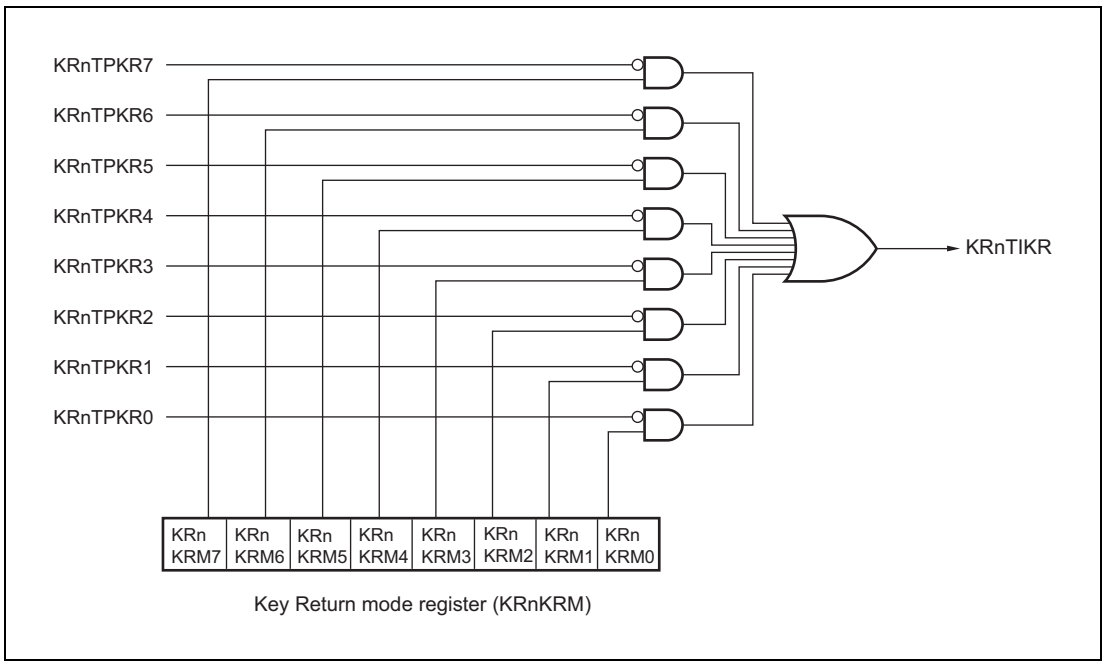


Figure 30.1 Block Diagram of the Key Return Function

30.3 Registers

30.3.1 List of Registers

KR registers are listed in the following table.

For details about <KRn_base>, see **Section 30.1.2, Register Base Address**.

Table 30.9 List of Registers

Module Name	Register Name	Symbol	Address
KRn	Key return mode register	KRnKRM	<KRn_base>

30.3.2 KRnKRM — Key Return Mode Register

This register enables/disables the key input signal detection.

Access: This register can be read/written in 8-bit or 1-bit units.

Address: <KRn_base>

Value after reset: 00_H This register is cleared by any reset.

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 30.10 KRnKRM Register Contents

Bit position	Bit name	Function
7 to 0	KRnKRMm	Enables/disables the key input signal detection. 0: Disabled 1: Enabled

30.4 Operation

30.4.1 Interrupt Request KRnTIKR

The interrupt request KRnTIKR is generated when a low level is input to the corresponding key input pin KRnTPKRm while input to the key input pin KRnTPKRm is enabled (KRnKRM.KRnKRMm = 1).

Figure 30.2 shows how the interrupt request is generated:

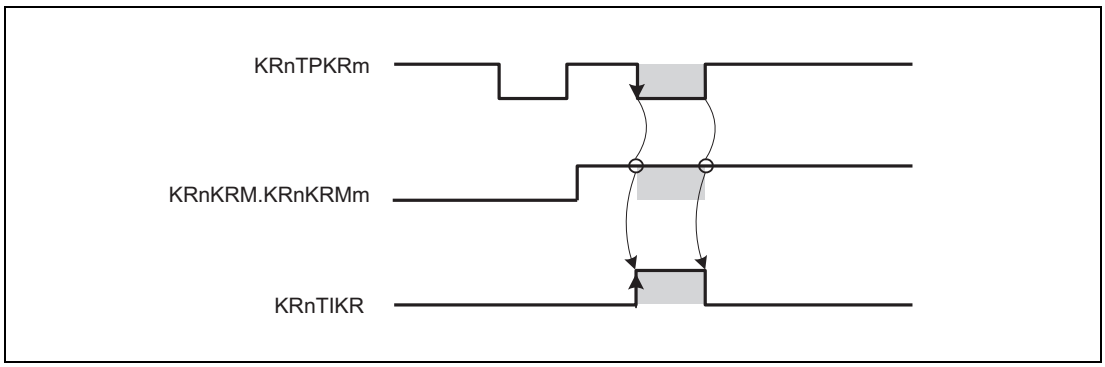


Figure 30.2 Interrupt Request Generation

CAUTIONS

1. **When any of the key input pins KRnTPKRm is low, even if one of the key input pins goes from high to low, KRnTIKR is not generated. KRnTIKR is only generated after all key input pins go to from high to low.**
2. **If the key input value changes at the same time the setting of KRnKRM.KRnTPKRm, unintended key interrupt request KRnTIKR might be generated.**
   Therefore, mask (forbid) KRnTIKR of the interrupt controller before changing KRnKRM.KRnTPKRm from 0 to 1, or from 1 to 0.

Section 31 Functional Safety

This section describes the overview of safety mechanism included in the RH850/F1L Series.

This microcontroller has been developed as a Safety Element out of Context (SEooC) in accordance to ISO26262.

As for the details on the development process and safety mechanism, please contact our sales representatives.

The following lists the failure detection functions provided by this microcontroller.

31.1 Overview

ECC and EDC

Detect failures of memories and data transfer paths and correct some types of failures.

For details, see the following sections.

- CSIHn RAM: **Section 15.7, Detection and Correction of Errors in CSIHn RAM**
- RS-CAN RAM: **Section 19.11, Detection and Correction of Errors in RS-CAN RAM**
- Code flash: **Section 35.9, ECC Error Detection and Correction for Code Flash Memory**
- Data flash: **Section 35.10, ECC Error Detection and Correction for Data Flash Memory**
- Local RAM: **Section 36.3, ECC Error Detection and Correction for Local RAM**

Memory Protection

Detects erroneous access to memories and peripheral circuits to protect the data in these elements against erroneous access.

For details, see **Section 3.1.1.4, Memory Management** and **Section 3.3.9, MPU Function Registers**.

Supply Voltage Monitor

Monitors the external and internal power supply voltages to detect an abnormal voltage.

For details, see **Section 9, Supply Voltage Monitor**.

Clock Monitor

Monitors the clock operation to detect an abnormal operation.

For details, see **Section 10.7, Clock Monitor A (CLMA)**.

Data CRC

Generates CRC to detect the data error.

For details, see **Section 32, 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 4, Write-Protected Registers**.

Section 32 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/F1L 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.

32.1 Features of RH850/F1L DCRA

32.1.1 Number of Units

This microcontroller has the following number of DCRA units.

Table 32.1 Number of Units

Product Name	RH850/F1L 48 pins	RH850/F1L 64 pins	RH850/F1L 80 pins	RH850/F1L 100 pins	RH850/F1L 144 pins	RH850/F1L 176 pins
Number of Units	1	1	4	4	4	4
Name	DCRAn (n = 0)	DCRAn (n = 0)	DCRAn (n = 0 to 3)	DCRAn (n = 0 to 3)	DCRAn (n = 0 to 3)	DCRAn (n = 0 to 3)

Table 32.2 Index

Index	Meaning
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.

32.1.2 Register Base Address

DCRAn base addresses are listed in the following table.

DCRAn register addresses are given as offsets from the base addresses.

Table 32.3 Register Base Address

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

32.1.3 Clock Supply

The DCRAn clock supply is shown in the following table.

Table 32.4 Clock Supply

Unit Name	Clock for the Unit	Clock Supply Name
DCRAn	PCLK	CPUCLK2

32.1.4 Reset Sources

DCRAn reset sources are listed in the following table. DCRAn is initialized by these reset sources.

Table 32.5 Reset Sources

Unit Name	Reset Source
DCRAn	All reset sources (ISORES)

32.2 Overview

32.2.1 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.

32.2.2 Block Diagram

The following picture shows the block diagram of the data CRC function A.

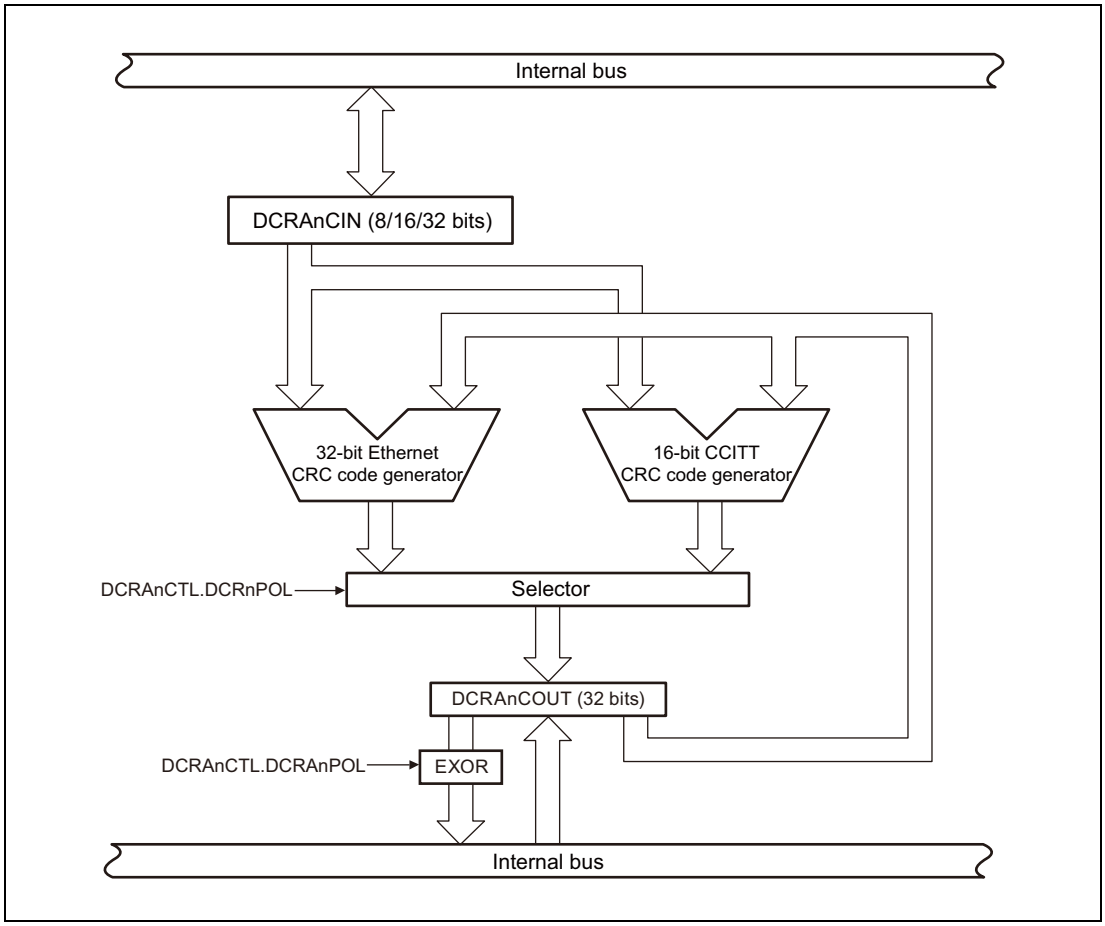
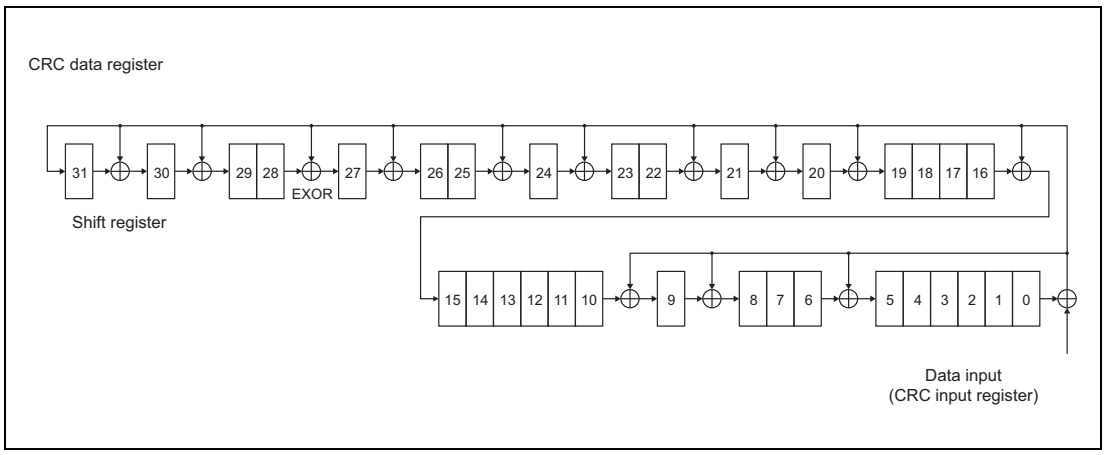


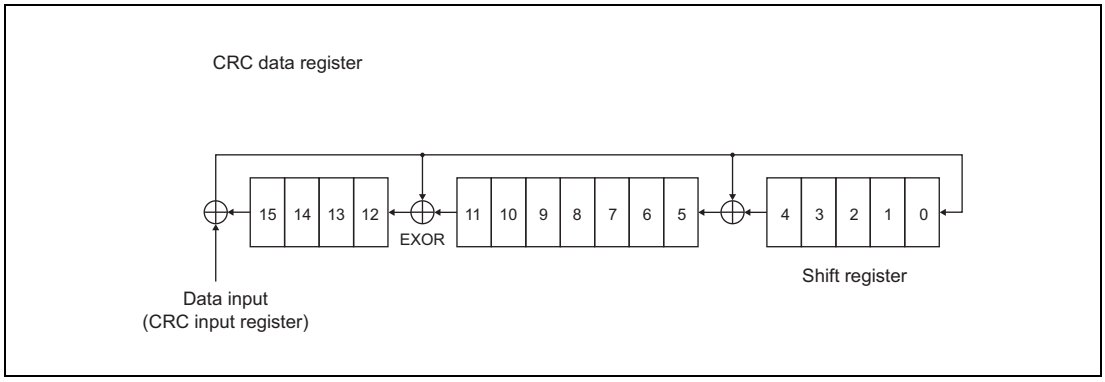
Figure 32.1 Block Diagram of Data CRC Function A

32.2.3 Operational Circuit

- 32-bit Ethernet



- 16-bit CCITT



32.3 Registers

32.3.1 List of Registers

DCRA registers are listed in the following table.

For details about <DCRAn_base>, see **Section 32.1.2, Register Base Address**.

Table 32.6 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

32.3.2 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/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 32.7 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]

32.3.3 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/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 32.8 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.

32.3.4 DCRAnCTL — CRC Control Register

This register controls the CRC generation process.

Access: This register can be read/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 32.9 DCRAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2, 1	DCRAnISZ[1:0]	Specifies 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. It means that 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, if the CRC input bit width is 8 bits (DCRAnISZ = 10). 1: 16-bit CCITT CRC polynomial generation. The byte order of the DCRAnCIN register is MSB (most significant bit) first. It means that 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, if the CRC input bit width is 8 bits (DCRAnISZ = 10).

CAUTION

- If the CRC generation method (DCRAnCTL.DCRAnPOL) is changed, the DCRAnCOUT register must be initialized by setting the initial start value.
- The CRC bit width (DCRAnCTL.DCRAnISZn) must be set according to the data block bit width. Changing the CRC bit width is not allowed during processing of a data block (a data block consists of N bytes, half-words or words). 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.

32.4 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.

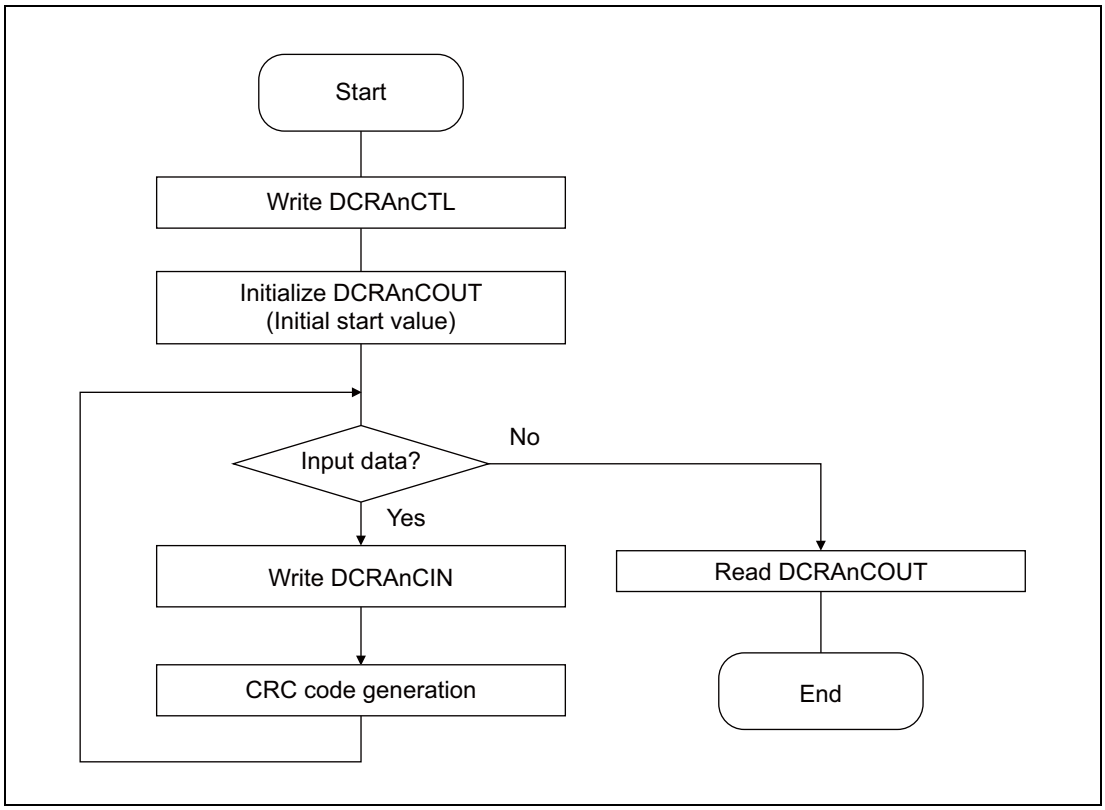


Figure 32.2 Flowchart of Data CRC Function A

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 32.10 Setting Example of Initial Start Values (When Read at a Reset)

	Initial Start Value	EXOR Value	DCRA0COUT 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 33 Intelligent Cryptographic Unit (ICUSB)

As for the intelligent cryptographic unit (ICUSB), the separate document is provided.

In addition, since ICUSB is mounted on the limited number of products, please contact our sales representatives for more information on the ICUSB mounted products.

Section 34 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 the product with the 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.

34.1 Overview of RH850/F1L OCD

34.1.1 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 to any execution address. Of the 12 breakpoints, a maximum of four breakpoints can be specified to any access (access address, access data).

(4) Software break

Software break point can be specified to 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 accuracy of measurement, see the user's manual of the debugger.

(11) Mask function

Masking the following factors are possible.

- All reset sources except for a POC reset and a wakeup reset
- $\overline{\text{MEMC0WAIT}}$

(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.

34.2 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 34.1 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 34.2 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
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, 1
Clocked serial interface G (CSIG)	CSIGnEMU.CSIGnSVSDIS 0: Stops during break 1: Continues during break	0, 1
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 34.3 Modules that Stop when EPC.SVSTOP = 1

Module
LIN / UART interface (RLIN3)
Low-Power Sampler (LPS)

34.3 Registers

34.3.1 EPC — Emulation Peripheral Control Register

This register stops operation of peripheral functions (timer, serial interface, and A/D converter) in debug mode (SVSTOP), and inhibits interference to special sequence access from a debugger using register operation (SVACCESS).

Access: Accessing from the user program is prohibited.

Address: —

Value after reset: 00_H

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

Table 34.4 EPC Register Contents

Bit Position	Bit Name	Function
7	SVACCESS	Inhibits interferences to special sequence access from a debugger using register operation. 0: User program is accessing 1: Debugger program is accessing
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. As for the setting of the debugger, see the user's manual of the debugger.

34.4 Caution on Using On-Chip Debugging

34.4.1 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 number of write/erase count of flash memory cannot be guaranteed.

Section 35 Flash Memory

This section describes the flash memory mounted on RH850/F1L.

The first part in this section describes the characteristics of the mounted flash memory and the characteristics specific to RH850/F1L, such as the memory map, flash memory programming, and ECC.

35.1 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, and WDTA, CVM, and external bus clocks.
- Support for the error detection/correction function (ECC) in the code flash memory and data flash memory
  The ECC function is included, which can detect 2-bit errors and detect/correct 1-bit errors.
- An interrupt is acceptable in self-programming mode.

For code flash sizes and data flash sizes of each product, see the following sections.

- **Section 3.2.4.1, Code Flash Area**
- **Section 3.2.4.2, Data Flash Area**

35.2 Structure of Memory

35.2.1 Mapping of Code Flash Memory

Figure 35.1 illustrates the mapping of the code flash memory for the 2-MB device. The user area of the code flash memory of the RH850/F1L is divided into 8- 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.

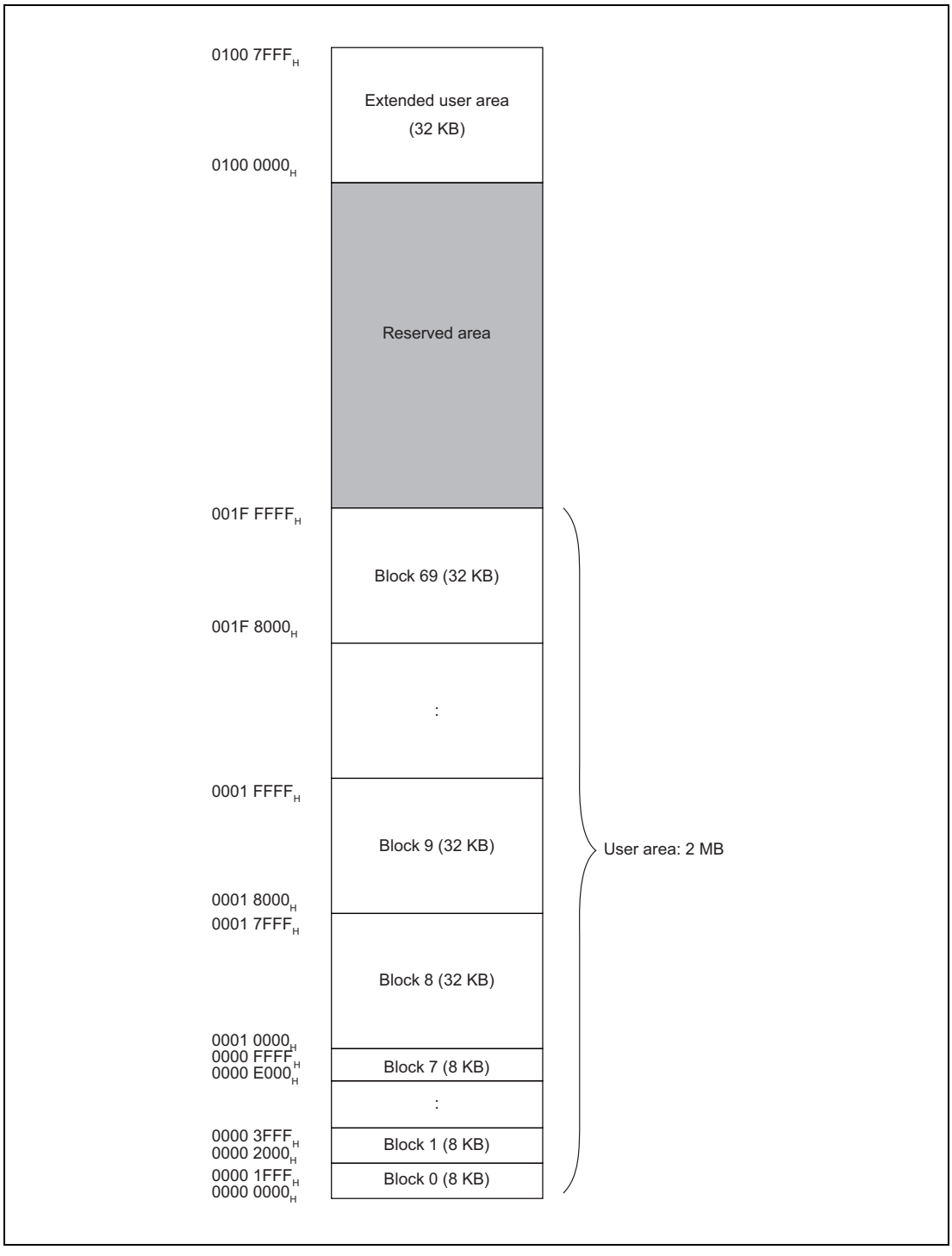


Figure 35.1 Mapping of the F1L-176 Pins/2 MB Code Flash Memory

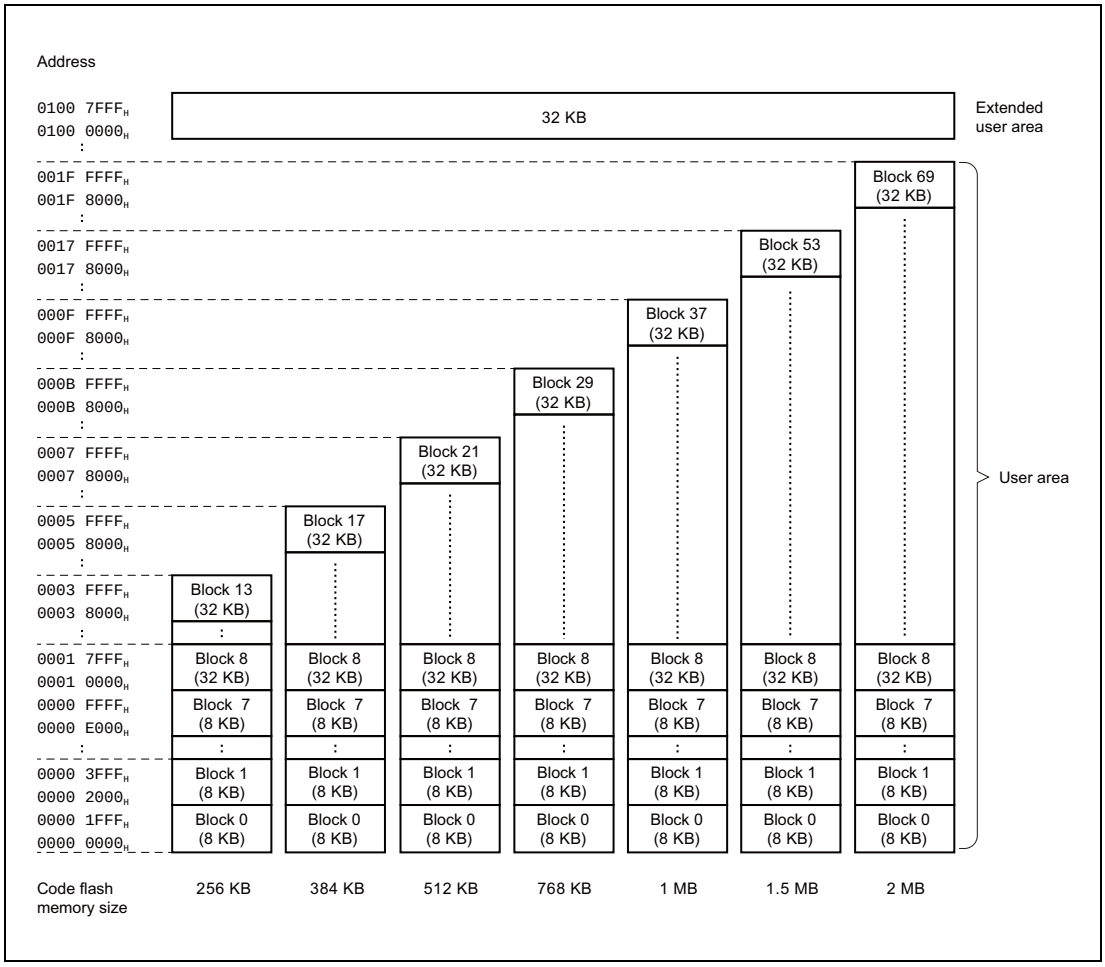


Figure 35.2 Mapping of Code Flash Memory

35.2.2 Mapping of Data Flash Memory

The data area of the data flash memory in the RH850/F1L is divided into 64-byte blocks, with each being a unit for erasure. **Figure 35.3** shows the mapping of the data flash memory.

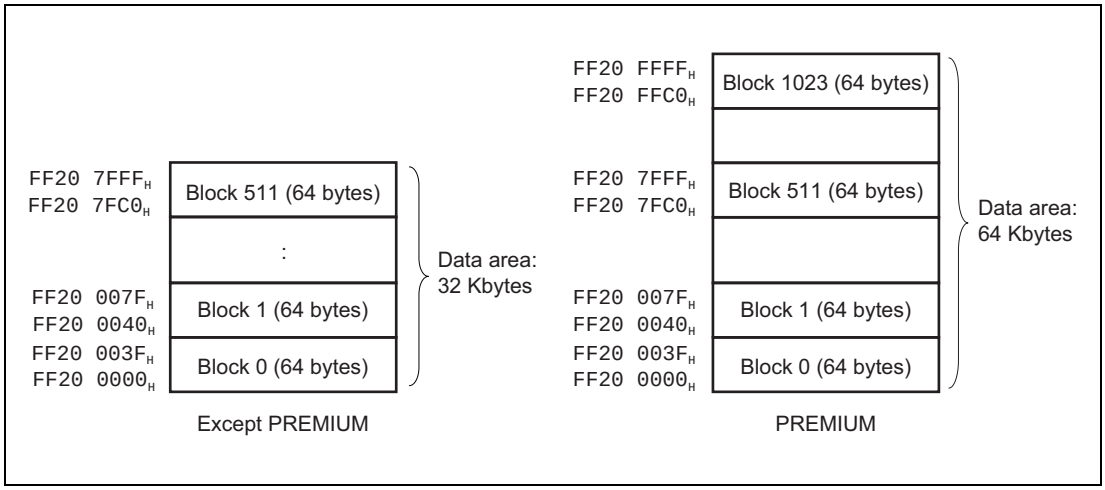


Figure 35.3 Mapping of the Data Flash Memory

35.3 Operating Modes Associated with Flash Memory

Figure 35.4 is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, see **Section 5, Operating Modes**.

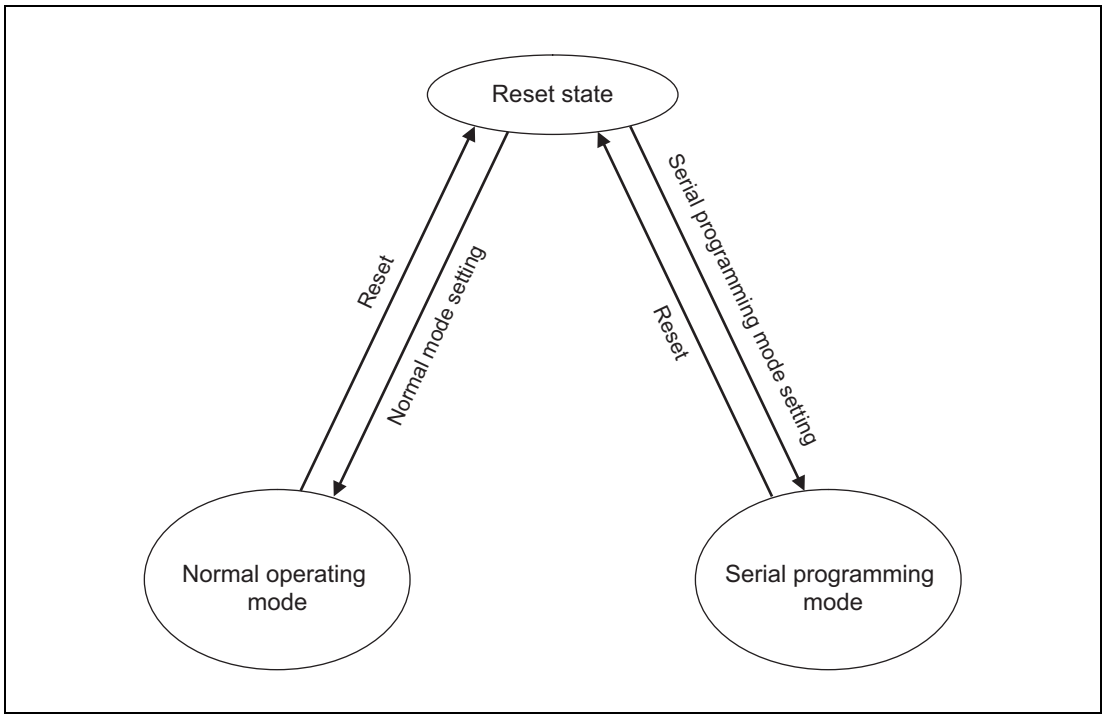


Figure 35.4 Mode Transition Associated with Flash Memory

Table 35.1 shows the flash memory area which is programmable and erasable in each mode and the boot program after reset release.

Table 35.1 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	Firmware program for serial programming

35.4 Functions

35.4.1 Overview of Functions

The flash memory of the RH850/F1L 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 35.2** gives an overview of the methods of programming and the corresponding operating modes.

Table 35.2 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 executing 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 being programmed. 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 35.3 lists the functions of the flash memory. A dedicated flash memory programmer commands realize serial programming, while reading of the flash memory by a library function or the user program realizes self-programming.

Table 35.3 Basic Functions at a Glance

Function	Description in Overview	Level of Support (√: Supported, Δ: 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.	√	Δ (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.	√	Δ (Only when setting is prohibited after being permitted)
Protection settings	Lock bits for all blocks of code flash memory are provided and the value of the reset vector is variable.	Δ (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/F1L.	√	√
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 35.4** and **Table 35.5**.

Table 35.4 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 frash 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 frash memory programmer	The connection of a dedicated frash memory programmer for serial programming is prohibited. Since execution of the configuration clearing command is also prohibited when the connection of a dedicated frash 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 35.5 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 frash 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 35.6**.

Table 35.6 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 FLMD 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 35.5**, 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.

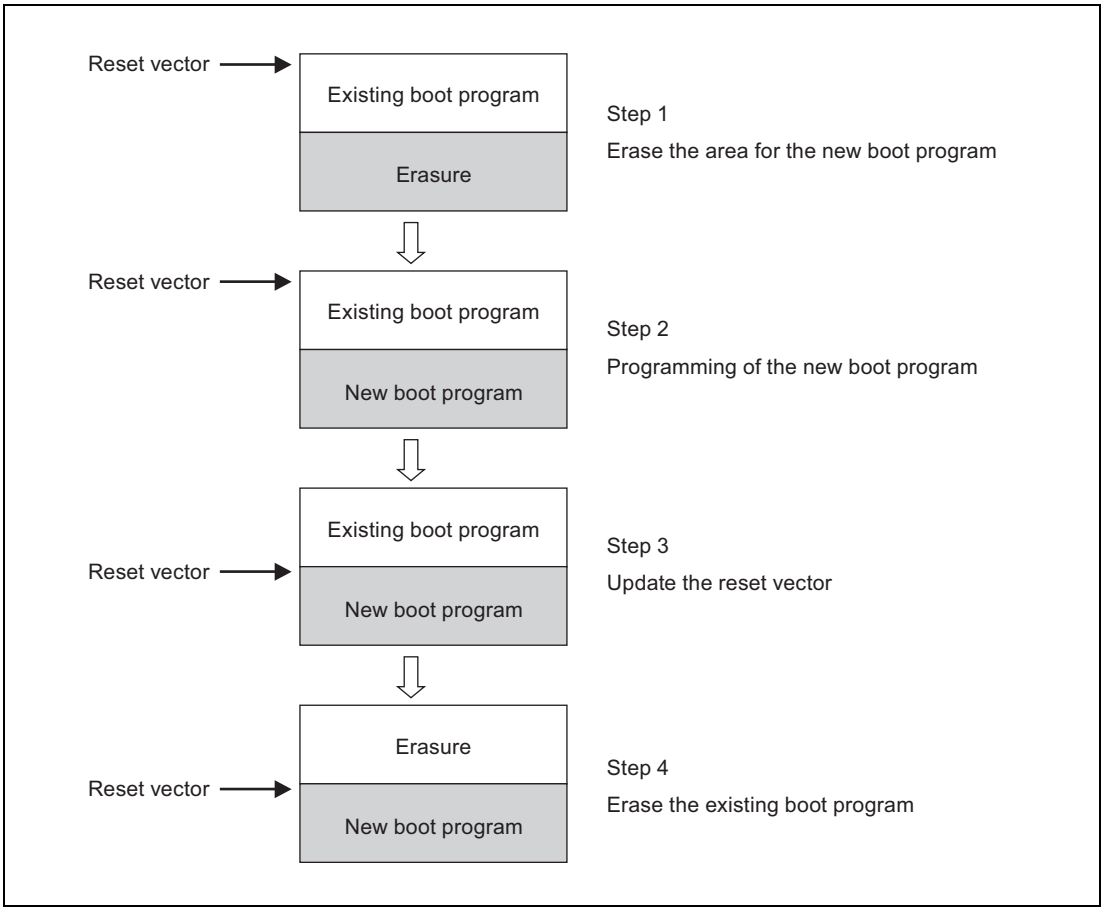


Figure 35.5 Utilizing the Variable Reset Vector Function to Update the Boot Program

35.5 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.

35.5.1 Environments for Programming

The recommended environments for handling the flash memory of the microcontroller with data are described below.

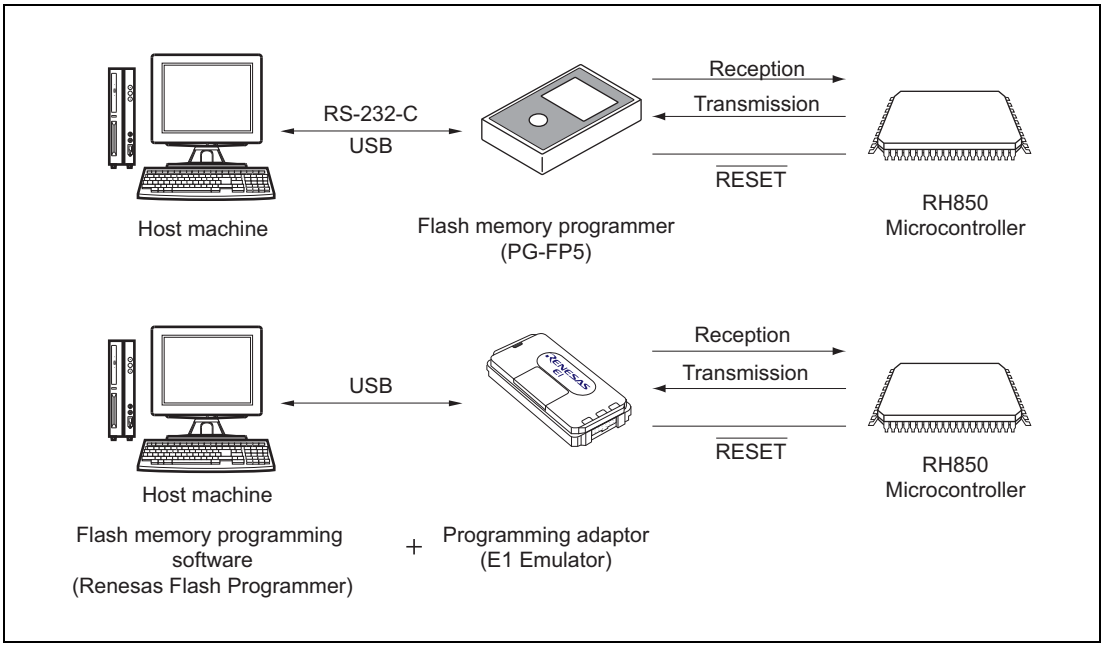


Figure 35.6 Environments for Handling Programs of the Flash Memory

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*.

35.6 Communication Modes

35.6.1 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.

- FLSCI3RXD, FLSCI3TXD/JP0_0: Receive data input/transmit data output

35.6.2 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.

- FLSCI3RXD/JP0_0: Receive data input
- FLSCI3TXD/JP0_1: Transmit data output

35.6.3 Synchronous Flash Programming Interface CSI

The synchronous serial programming interface CSI is connected to the flash memory programmer with the following ports.

- FLSCI3RXD/JP0_0: Receive data input
- FLSCI3TXD/JP0_1: Transmit data output
- FLSCI3SCKI/JP0_2: Serial clock input

The flash memory programmer outputs the serial data clock SCK, and the microcontroller operates as a slave.

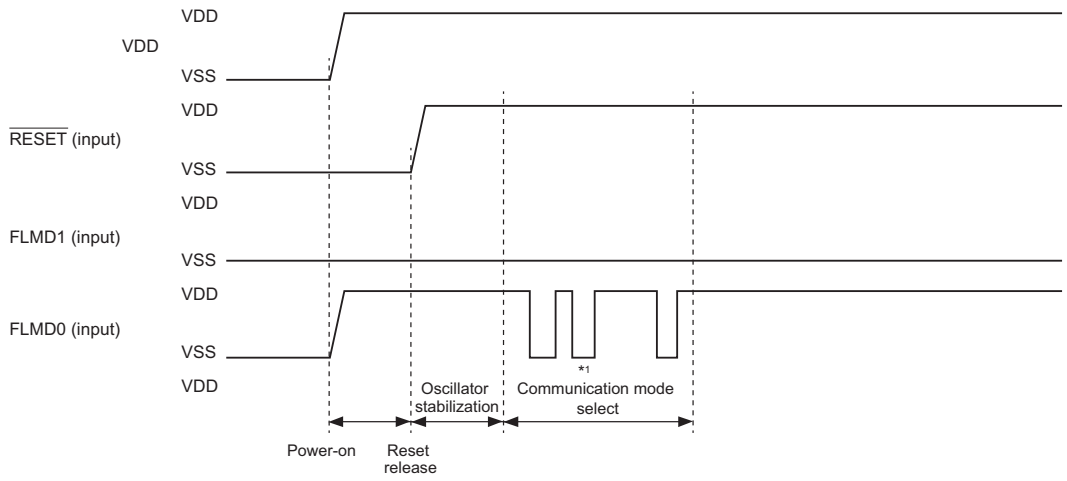
NOTE

For details on flash programming software, Renesas Flash Programmer, see the *Renesas Flash Programmer Flash Programming Software User's Manual*.

35.6.4 Selection of Communication Method

In RH850/F1L, communication method can be selected by pulse input to the FLMD 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 35.7 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 35.7 Selection of Communication Method

35.7 Self Programming

35.7.1 Outline

The RH850/F1L 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.

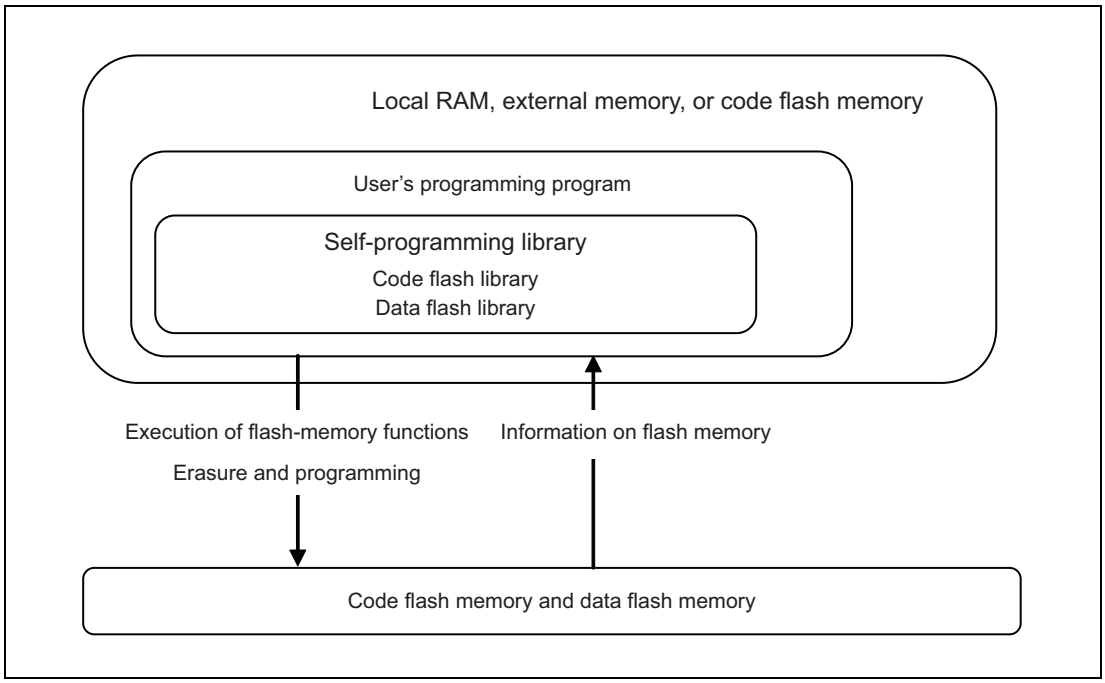


Figure 35.8 Schematic View of Self Programming

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.

35.7.2 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 35.7 Conditions under which Background Operation is Usable

Range for Writing	Range for Reading
Data flash memory	Code flash memory

35.7.3 Enabling of Self-Programming

The self-programming function can be activated in normal operating mode.

Erasure and programming of the 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 35.7.3.1, FLMDCNT Register**.

35.7.3.1 FLMDCNT Register

This register specifies the internal pull-up or pull-down of the FLMD0 pin.

Writing to this register is protected by a special sequence of instructions by using the protection command register FLMDPCMD. For details, see **Section 4, Write-Protected Registers**.

Access: This register can be read/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 35.8 FLMDCNT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	FLMDPUP	FLMD0 Pin Software Control 0: Pull-down selected 1: Pull-up selected

35.8 Reading Flash Memory

35.8.1 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.

35.8.2 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. Once the setting for the number of cycles is made, data can simply be read out through access to addresses in the data flash memory.

Values read from data flash memory that has been erased but not yet been programming again are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

35.8.2.1 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 and written in 8-bit units.

Address: FFC5 A010$_H$

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 35.9 EEPRDCYCL Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, 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 ≤ 20 MHz	20 MHz < FCPUCLK ≤ 40 MHz	40 MHz < FCPUCLK ≤ 60 MHz	60 MHz < FCPUCLK ≤ 80 MHz	80 MHz < FCPUCLK ≤ 96 MHz
0000	1	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0001	2	√	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0010	3	√	√	Setting prohibited	Setting prohibited	Setting prohibited
0011	4	√	√	√	Setting prohibited	Setting prohibited
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 = (4 + Number of wait cycles) × 2 / 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.

35.8.2.2 PRDNAMEn — Product Name Storage Register (n = 1 to 3)

This register stores the product name. The product model 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 model name respectively.

Access: This register 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 35.11** to **Table 35.14**.

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 35.10 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)

Table 35.11 to **Table 35.14** list registers related to product information.

Table 35.11 List of Registers Related to Product Information (ECO)

Product Model Name	PRDNAME1	PRDNAME2	PRDNAME3
R7F701006	3746 3752	3030 3130	2020 2036
R7F701007	3746 3752	3030 3130	2020 2037
R7F701008	3746 3752	3030 3130	2020 2038
R7F701009	3746 3752	3030 3130	2020 2039
R7F701010	3746 3752	3130 3130	2020 2030
R7F701011	3746 3752	3130 3130	2020 2031
R7F701012	3746 3752	3130 3130	2020 2032
R7F701013	3746 3752	3130 3130	2020 2033
R7F701014	3746 3752	3130 3130	2020 2034
R7F701015	3746 3752	3130 3130	2020 2035
R7F701016	3746 3752	3130 3130	2020 2036

Table 35.11 List of Registers Related to Product Information (ECO)

Product Model Name	PRDNAME1	PRDNAME2	PRDNAME3
R7F701017	3746 3752	3130 3130	2020 2037
R7F701018	3746 3752	3130 3130	2020 2038
R7F701019	3746 3752	3130 3130	2020 2039
R7F701020	3746 3752	3230 3130	2020 2030
R7F701021	3746 3752	3230 3130	2020 2031
R7F701022	3746 3752	3230 3130	2020 2032
R7F701023	3746 3752	3230 3130	2020 2033
R7F701024	3746 3752	3230 3130	2020 2034
R7F701025	3746 3752	3230 3130	2020 2035
R7F701026	3746 3752	3230 3130	2020 2036
R7F701027	3746 3752	3230 3130	2020 2037
R7F701028	3746 3752	3230 3130	2020 2038
R7F701029	3746 3752	3230 3130	2020 2039
R7F701030	3746 3752	3330 3130	2020 2030
R7F701032	3746 3752	3330 3130	2020 2032
R7F701033	3746 3752	3330 3130	2020 2033
R7F701034	3746 3752	3330 3130	2020 2034

Table 35.12 List of Registers Related to Product Information (ADVANCED)

Product Model Name	PRDNAME1	PRDNAME2	PRDNAME3
R7F701040	3746 3752	3430 3130	2020 2030
R7F701041	3746 3752	3430 3130	2020 2031
R7F701042	3746 3752	3430 3130	2020 2032
R7F701043	3746 3752	3430 3130	2020 2033
R7F701044	3746 3752	3430 3130	2020 2034
R7F701045	3746 3752	3430 3130	2020 2035
R7F701046	3746 3752	3430 3130	2020 2036
R7F701047	3746 3752	3430 3130	2020 2037
R7F701048	3746 3752	3430 3130	2020 2038
R7F701049	3746 3752	3430 3130	2020 2039
R7F701050	3746 3752	3530 3130	2020 2030
R7F701051	3746 3752	3530 3130	2020 2031
R7F701052	3746 3752	3530 3130	2020 2032
R7F701053	3746 3752	3530 3130	2020 2033

Table 35.13 List of Registers Related to Product Information (PREMIUM)

Product Model Name	PRDNAME1	PRDNAME2	PRDNAME3
R7F701054	3746 3752	3530 3130	2020 2034
R7F701055	3746 3752	3530 3130	2020 2035
R7F701056	3746 3752	3530 3130	2020 2036
R7F701057	3746 3752	3530 3130	2020 2037

Table 35.14 List of Registers Related to Product Information (Gateway)

Product Model Name	PRDNAME1	PRDNAME2	PRDNAME3
R7F701002	3746 3752	3030 3130	2020 2032
R7F701003	3746 3752	3030 3130	2020 2033

35.9 ECC Error Detection and Correction for Code Flash Memory

35.9.1 Outline of ECC Functions for Code Flash Memory

Table 35.15 gives an outline of the ECC functions for the code flash memory.

Table 35.15 Outline of the ECC Functions for the Code Flash Memory

Item	Outline of Functions
Detection and correction of ECC errors	ECC error detection and correction can be enabled or disabled.When enabled, the following settings are also selectable. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection Neither detection or correction occurs if this function is disabled. In the initial state, 1-bit errors are detected and corrected, 2-bit errors are detected, and error notification proceeds.
Error notification	Notification proceeds in response to an ECC error being found. ECC errors • Enabling or disabling of error notification in the case of detection of ECC 2-bit errors is selectable. • Enabling or disabling of error notification in the case of detection of ECC 1-bit errors is selectable. In the initial state, error notification is enabled for 2-bit errors and disabled for 1-bit errors. As the error notification, an ECC 2-bit error is output as a signal (a source) and an ECC 1-bit error is output as a signal (a source).
Error status	Monitoring for the detection of ECC 2-bit errors and for the detection of ECC 1-bit errors is available. The ECC 1-bit error status is set only in the situation where no error status is 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. The error status also functions as the enable bit for the capture address.
Address capture	In the situation where no error status is set, the address at which the first ECC error occurred is captured. In addition, when the retained address source is an ECC 1-bit error or ECC 2-bit error, the address is also captured. The error signal for an ECC 2-bit error and the error signal for an ECC 1-bit error serve as triggers for address capture. These enable the bits for the error status.
Errant instruction execution suppression	When an ECC 2-bit error is detected during the instruction fetch, an SYSERR exception is generated.

35.9.2 Interrupt Request

The following table shows interrupt requests by ECC for the code flash memory.

Table 35.16 Interrupt Requests by ECC for Code Flash Memory

Unit Interrupt Signal	Outline	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of code flash memory	INTECCSDFLI0	—
—	ECC 2-bit error interrupt of code flash memory	SYSERR	—

35.9.3 Registers

35.9.3.1 List of Registers

This table below lists the registers for the code flash ECC.

Table 35.17 List of Code Flash ECC Registers

Register Name	Abbreviation	Address
Code flash ECC control register	CFECCCTL	FFC6 2000_H
Code flash first error status register	CFFSTERSTR	FFC6 2004_H
Code flash error status clear register	CFFSTSTC	FFC6 2024_H
Code flash error overflow status register	CFOVFSTR	FFC6 2028_H
Code flash error overflow status clear register	CFOVFSTC	FFC6 $202C_H$
Code flash error notification control register	CFERRINT	FFC6 2030_H
Code flash first error address register	CFFSTEADR	FFC6 2034_H
Code flash test control register	CFTSTCTL	FFC6 2054_H

35.9.3.2 CFECCCTL — Code Flash ECC Control Register

This register enables or disables the ECC error detection/correction and 1-bit error correction. Set the PROT1 and PROT0 bits to 01_B when writing to this register.

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

Address: FFC6 2000_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 35.18 CFECCCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	These bits enable or disable modification of the ECCDIS and SEDDIS 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 to these bits, 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 this bit, 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.

35.9.3.3 CFFSTERSTR — Code Flash First Error Status Register

This register monitors an error which occurs first.

Detecting an ECC 1-bit error sets the SEDF bit and detecting an ECC 2-bit error sets the DEDF bit while ECC error detection/correction is enabled.

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

Address: FFC6 2004$_H$

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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF	SEDF
Value after reset	0	0	0	0	0	0	0	0	0	0	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 35.19 CFFSTERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF	ECC 2-Bit Error Monitor Flag This bit is set to 1 when an ECC 2-bit error occurs, regardless of the SEDF or DEDF state. 0: An internal reset or a pin reset has been generated. The FSTERRCLR bit in code flash error status clear register has been set. 1: An ECC 2-bit error has occurred.
0	SEDF	ECC 1-Bit Error Monitor Flag This bit is set to 1 when an ECC 1-bit error occurs while both SEDF and DEDF are 0. 0: An internal reset or a pin reset has been generated. The FSTERRCLR bit in code flash error status clear register has been set. 1: An ECC 1-bit error has occurred while both SEDF and DEDF are 0.

35.9.3.4 CFFSTSTC — Code Flash Error Status Clear Register

This register clears error flags in the code flash error status register.

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

Address: FFC6 2024$_H$

Value after reset: 00$_H$

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

Table 35.20 CFFSTSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	FSTERRCLR	Clear SEDF and DEDF Flags in CFFSTERSTR Writing 1 to this bit clears the SEDF and DEDF flags in the CFFSTERSTR register.

35.9.3.5 CFOVFSTR — Code Flash Error Overflow Status Register

This register monitors occurrence of a code flash error overflow.

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

Address: FFC6 2028$_H$

Value after reset: 0000 0000$_H$

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

Table 35.21 CFOVFSTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF	Error Overflow Flag This bit is set when an ECC error occurs (an overflow occurs) while the error address registers is valid. • If the retained address source is SEDF: When a 1-bit error occurs in the same address, an overflow does not occur. When a 1-bit error occurs in a different address, an overflow occurs. When a 2-bit error occurs, an overflow occurs. • If the retained address source is DEDF: When a 1-bit error occurs, an overflow occurs. When a 2-bit error occurs in the same address, an overflow does not occur. When a 2-bit error occurs in a different address, an overflow occurs.

35.9.3.6 CFOVFSTC — Code Flash Error Overflow Status Clear Register

This register is used to clear the code flash error overflow flag. The flag is cleared by writing 1 to the ERROVFCLR bit.

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

Address: FFC6 202C$_H$

Value after reset: 00$_H$

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

Table 35.22 CFOVFSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	ERROVFCLR	Error Overflow Flag Clear Writing 1 to this bit clears the ERROVF flag in the CFOVFSTR register. This bit is always read as 0.

35.9.3.7 CFERRINT — Code Flash Error Notification Control Register

This register enables or disables error notification signal generation when an ECC 1-bit is detected.

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

Address: FFC6 2030$_H$

Value after reset: 02$_H$

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

Table 35.23 CFERRINT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	SEDIE	ECC 1-Bit Error Notification Control This bit controls error notification on 1-bit error detection when ECC error detection/correction is enabled. 0: ECC 1-bit error notification is disabled. 1: ECC 1-bit error notification is enabled.

35.9.3.8 CFFSTEADR — Code Flash First Error Address Register

This register holds the address where an ECC error has occurred.

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

Address: FFC6 2034$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	CFFSTEADR[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
	CFFSTEADR[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 35.24 CFFSTEADR Register Contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24 to 4	CFFSTEADR [24:4]	First Error Occurrence Address These bits are read-only bits to monitor the address where the first error has occurred. The address is updated as follows: • If the retained address source is SEDF: When a 1-bit error occurs in the same address, the address is not overwritten. When a 1-bit error occurs in a different address, the address is not overwritten. When a 2-bit error occurs, the address is overwritten. • If the retained address source is DEDF: When a 1-bit error occurs, the address is not overwritten. When a 2-bit error occurs in the same address, the address is not overwritten. When a 2-bit error occurs in a different address, the address is not overwritten.
3 to 0	Reserved	When read, the value after reset is returned.

35.9.3.9 CFTSTCTL — Code Flash Test Control Register

This register is used for the ECC test (self-diagnosis). The ECC bits can be read after ECC test mode is set by setting ECCTST to 1.

During this mode, the ECC check bits are read out instead of flash data.

Set the PROT1 and PROT0 bits to 01_B when writing to this register.

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

Address: FFC6 2054_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	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST
Value after 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 35.25 CFTSTCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	These bits enable or disable modification of the ECCTST bit. 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 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	ECCTST	ECC Test Mode Writing 1 to this bit sets ECC test mode. 0: Normal mode 1: ECC test mode

35.9.4 ECC Test Function for the Code Flash

Through appropriate register setting, the code flash data and ECC bits can be read out.

(1) Reading out code flash data

1. Set the ECCDIS bit in the code flash ECC control register to 1 to disable ECC error detection and correction.
2. When ECCDIS = 1, neither error detection nor correction is executed when the code flash is read; the data output from the code flash is read just as it is.

(2) Reading ECC bits

1. Set the ECCDIS bit in the code flash ECC control register to 1 to disable ECC error detection and correction.
2. Set the ECCTST bit in the code flash test control register to 1 to set test mode.
3. When the code flash is read, the ECC bits are read via bits 8 to 0 (the upper bits are undefined).

(3) Canceling this mode

1. Set the ECCDIS bit in the code flash ECC control register to 0 to enable ECC error detection and correction.
2. Set the ECCTST bit in the code flash test control register to 0 to set normal mode.

35.10 ECC Error Detection and Correction for Data Flash Memory

35.10.1 Outline of ECC Functions for Data Flash Memory

Table 35.26 gives an outline of the ECC functions for the data flash memory.

Table 35.26 Outline of the ECC Functions for the Data Flash Memory

Item	Outline of Functions
Detection and correction of ECC errors	ECC errors detection and correction can be enabled or disabled. When enabled, the following settings are also selectable. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection Neither detection or correction occurs if this function is disabled. In the initial state, 1-bit errors are detected and corrected, 2-bit errors are detected, and error notification proceeds.
Error notification	Notification proceeds in response to an ECC error being found. • Enabling or disabling of error notification in the case of detection of ECC 2-bit errors is selectable. In the initial state, error notification is enabled for 2-bit errors.
Error status	Monitoring for the detection of ECC 2-bit errors and for the detection of ECC 1-bit errors is available. The module has status registers for setting to indicate the occurrence of a first error in a situation where no error status setting has been made and for setting to indicate the occurrence of a second error. A register for clearing the error status is provided. The error status includes dual bits for enabling captured addresses.

35.10.2 Interrupt Request

The following table shows interrupt requests by ECC for the data flash memory.

Table 35.27 Interrupt Requests by ECC for Data Flash Memory (Reading by the CPU)

Unit Interrupt Signal	Outline	Name	DMA Trigger Number
—	ECC 2-bit error interrupt of data flash memory	INTECCDEEP0	—

35.10.3 Registers

35.10.3.1 List of Registers

The following table lists the registers for ECC for the data flash memory.

Table 35.28 List of Data Flash ECC Registers

Register Name	Abbreviation	Address
Data flash ECC control register	DFECCCTL	FFC6 6000$_H$
Data flash error status register	DFERSTR	FFC6 6004$_H$
Data flash error status clear register	DFERSTC	FFC6 6008$_H$
Data flash error notification control register	DFERRINT	FFC6 6014$_H$
Data flash test control register	DFTSTCTL	FFC6 601C$_H$

35.10.3.2 DFECCCTL — Data Flash ECC Control Register

This register enables or disables the ECC error detection/correction and 1-bit error correction. Set the PROT1 and PROT0 bits to 01$_B$ when writing to this register.

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

Address: FFC6 6000$_H$

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 35.29 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 to these bits, 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 this bit, 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.

35.10.3.3 DFERSTR — Data Flash Error Status Register

This register monitors an error occurrence.

Detecting an ECC 1-bit error sets the SEDF bit and detecting an ECC 2-bit error sets the DEDF bit while ECC error detection/correction is enabled.

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

Address: FFC6 6004_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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF	SEDF
Value after reset	0	0	0	0	0	0	0	0	0	0	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 35.30 DFERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF	ECC 2-Bit Error Monitor Flag This bit is set to 1 when an ECC 2-bit error occurs while both SEDF and DEDF are 0. 0: An internal reset or a pin reset has been generated. The ERRCLR bit in data flash error status clear register is set. 1: An ECC 2-bit error has occurred while both SEDF and DEDF are 0.
0	SEDF	ECC 1-Bit Error Monitor Flag This bit is set to 1 when an ECC 1-bit error occurs while both SEDF and DEDF are 0. 0: An internal reset or a pin reset has been generated. The ERRCLR bit in data flash error status clear register is set. 1: An ECC 1-bit error has occurred while both SEDF and DEDF are 0.

35.10.3.4 DFERSTC — Data Flash Error Status Clear Register

This register clears error flags in the data flash error status register.

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

Address: FFC6 6008$_H$

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 35.31 DFERSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	ERRCLR	Clear SEDF and DEDF Flags in DFERSTR Writing 1 to this bit clears the SEDF and DEDF flags in the DFERSTR register.

35.10.3.5 DFERRINT — Data Flash Error Notification Control Register

This register enables or disables error notification signal generation when an ECC 2-bit error is detected.

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

Address: FFC6 6014$_H$

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 35.32 DFERRINT Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	DEDIE	ECC 2-Bit Error Notification Control This bit controls error notification on 2-bit error detection when ECC error detection/correction is enabled. 0: ECC 2-bit error notification is disabled. 1: ECC 2-bit error notification is enabled.
0	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.

35.10.3.6 DFTSTCTL — Data Flash Test Control Register

This register is used for the ECC test (self-diagnosis). The ECC bits can be read after ECC test mode is set by setting ECCTST to 1.

Set the PROT1 and PROT0 bits to 01_B when writing to this register.

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

Address: FFC6 601C$_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST
Value after 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 35.33 DFTSTCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	These bits enable or disable modification of the ECCTST bit. 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 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	ECCTST	ECC Test Mode Writing 1 to this bit sets ECC test mode. 0: Normal mode 1: ECC test mode

35.10.4 ECC Test Function for the Data Flash

Through appropriate register setting, the data flash data and ECC bits can be read out.

(1) Reading out data flash data

1. Set the ECCDIS bit in the data flash ECC control register to 1 to disable ECC error detection and correction.
2. When ECCDIS = 1, neither error detection nor correction is executed when the data flash is read; the data output from the data flash is read just as it is.

(2) Reading ECC data

1. Set the ECCDIS bit in the data flash ECC control register to 1 to disable ECC error detection and correction.
2. Set the ECCTST bit in the data flash test control register to 1 to set test mode.
3. When the data flash is read, the ECC bits are read via bits 6 to 0 (the upper bits are undefined).

(3) Canceling this mode

1. Set the ECCDIS bit in the data flash ECC control register to 0 to enable ECC error detection and correction.
2. Set the ECCTST bit in the data flash test control register to 0 to set normal mode.

35.11 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.

35.11.1 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 WDTA1 and WDTA0
- Frequency division for the external memory controller
- Enabling the high voltage monitor
- Enabling the low voltage monitor

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
	—	—	—	—	—	—	—	MEMC DIV	—	—	CVM_HD_EN	CVM_LD_EN	—	—	—	—

Table 35.34 Option Byte Setting (1/2)

Bit Position	Bit Name	Function
31	Reserved	When writing to this bit, 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 mode. 10: JP0 is used for LPD 1-pin mode. 11: JP0 is used for Nexus I/F.
28, 27	Reserved	When writing to these bits, write “1”.
26	WDT1_3	Specifies the activation code method of WDTA1. 0: Fixed activation code 1: Variable activation code
25	Reserved	When writing to this bit, write “1”.

Table 35.34 Option Byte Setting (2/2)

Bit Position	Bit Name	Function
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 to this bit, 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 9	Reserved	When writing to these bits, write “1”.
8	MEMCDIV	Specifies the frequency division for the external memory controller 0: CPUCLK/2 1: CPUCLK/4
7, 6	Reserved	When writing to these bits, 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 to these bits, write “1”.

35.11.2 OPBT0 — Option Byte 0 Register

Access: This register can only be read in 32-bit units in debug mode.

Value after reset: User defined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	OPJTAG[1:0]		—	—	WDT1_3	—	WDT1_1	WDT1_0	WDT0_3	—	WDT0_1	WDT0_0	WDT_2	WDT_1	WDT_0
Value after reset	1	0/1	0/1	1	1	0/1	1	0/1	0/1	0/1	1	0/1	0/1	0/1	0/1	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
	—	—	—	—	—	—	—	MEMCDIV	—	—	CVM_HD_EN	CVM_LD_EN	—	—	—	—
Value after reset	1	1	1	1	1	1	1	0/1	1	1	0/1	0/1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 35.35 OPBT0 Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned.
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 mode. 10: JP0 is used for LPD 1-pin mode. 11: JP0 is used for Nexus I/F.
28, 27	Reserved	When read, the value after reset is returned.
26	WDT1_3	Specifies the activation code method of WDTA1. 0: Fixed activation code 1: Variable activation code
25	Reserved	When read, the value after reset is returned.
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 read, the value after reset is returned.
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 9	Reserved	When read, the value after reset is returned.
8	MEMCDIV	Specifies the frequency division for the external memory controller 0: CPUCLK/2 1: CPUCLK/4
7, 6	Reserved	When read, the value after reset is returned.
5	CVM_HD_EN	High Voltage Monitor Enable 0: Disable high voltage detection 1: Enable high voltage detection

Table 35.35 OPBT0 Register Contents (2/2)

Bit Position	Bit Name	Function
4	CVM_LD_EN	Low Voltage Monitor Enable 0: Disable low voltage detection 1: Enable low voltage detection
3 to 0	Reserved	When read, the value after reset is returned.

35.12 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 undefined data that are read out becoming the source of faulty operation, 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 programming again

Note that reading from an area of code flash memory that has been erased but not yet been programming 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.

(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 a reset due to the signal on the $\overline{\text{RESET}}$ pin during programming and erasure, wait for at least 20 μs 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 generation 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 prove 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.

- Have the operating voltage from the power supply go beyond the allowed range.
- Change the frequency of the peripheral clock.

Section 36 RAM

This section describes the local RAM mounted on RH850/F1L and the error correction function (ECC) of the local RAM.

36.1 Features

- RH850/F1L includes three types of the local RAM:
  - Primary local RAM
    The primary local RAM is RAM area that can be accessed with speed. Values are not retained in this area in DEEPSTOP mode.
  - Secondary local RAM
    The secondary local RAM is a RAM area that can be accessed only after the system has waited for one clock cycle compared to the primary local RAM. Values are not retained in this area in DEEPSTOP mode.
  - Retention RAM
    The retention RAM is a RAM area that can be accessed only after the system has waited for one clock cycle compared to the primary local RAM. Values are retained in this area in DEEPSTOP mode.
    In addition, even though the power-supply voltage (REGVCC) falls below the POC voltage, if the RAM retention voltage is V_{VLVI} or greater, the RAM data is retained.
- Error detection/correction function (ECC) in the local RAM
  The ECC function is included, which can detect 2-bit errors and detect/correct 1-bit errors.

The size of the RAM mounted on each product is shown below.

- **Section 3.2.4.3, Primary Local RAM Area**
- **Section 3.2.4.4, Secondary Local RAM Area**
- **Section 3.2.4.5, Retention RAM Area**

36.2 Memory Configuration

Figure 36.1 shows the memory map of the local RAM.

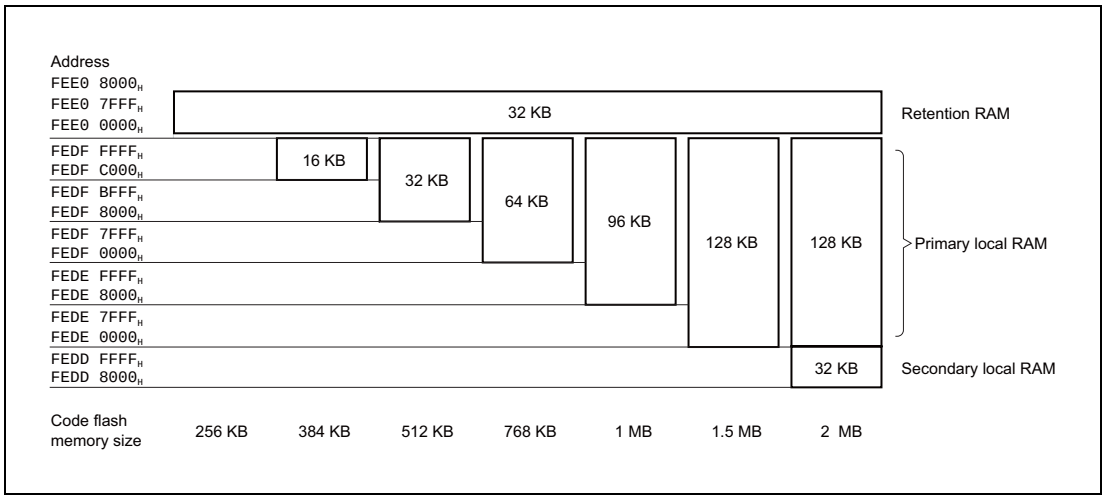


Figure 36.1 Memory Map of the Local RAM

36.3 ECC Error Detection and Correction for Local RAM

36.3.1 Outline of ECC Functions for Local RAM

Table 36.1 gives an outline of the ECC functions for the local RAM.

Table 36.1 List of the ECC Functions for the Local RAM

Item	Outline of Functions
ECC error detection/correction	The ECC error detection/correction can be enabled or disabled. When enabled, either of the following two options is selectable. • ECC error detection/correction (2-bit error detection and 1-bit error detection/ correction) • ECC error detection (2-bit error detection and 1-bit error detection) When disabled, errors are not detected or corrected. In the initial setting, the ECC function is enabled, and 1-bit error detection/correction and 2-bit error detection/notification are selected.
Error notification	When an ECC error occurs, the error is notified. • Error notification can be enabled or disabled when an ECC 2-bit error is detected. • Error notification can be enabled or disabled when an ECC 1-bit error is detected. In the initial setting, 2-bit error notification is enabled, and 1-bit error notification is disabled.
Error status	Monitoring for the detection of ECC 2-bit errors and for the detection of ECC 1-bit errors is available. The ECC 1-bit error status is set only in a situation where no error status setting has been made. The ECC 2-bit error status is set even when the ECC 1-bit error status has been set. A register for clearing the error status is provided.
Address capture	In the situation where no error status is set, the address at which the first ECC error occurred is captured. In addition, when the retained address source is an ECC 1-bit error, the address of ECC 2-bit error is also captured. The error signal for an ECC 2-bit error and the error signal for an ECC 1-bit error serve as triggers for address capture. These enable the bits for the error status.
Others	Executing an instruction is stopped when an ECC 2-bit error occurs during the instruction fetch.

CAUTION

When ECC error detection/correction for the embedded RAM is enabled, initialize the RAM with the 32-bit length access size before the RAM is used. If the RAM before initialization is accessed, 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 detected.

36.3.2 Interrupt Request

The following table shows interrupt requests by ECC for the local RAM.

Table 36.2 Interrupt Requests by ECC for Local RAM

Unit Interrupt number	Function	Name	DMA Trigger Number
—	ECC 1-bit error interrupt of local RAM	INTECCRAM	—
—	ECC 2-bit error interrupt of local RAM	SYSERR	—

36.4 Registers

36.4.1 List of Registers

The following table lists the ECC-related registers for the local RAM.

Table 36.3 List of RAM ECC Registers

Register Name	Symbol	Address
Local RAM ECC control register	LRECCCTL	FFC6 3000_H
Local RAM first error status register	LRFSTERSTR	FFC6 3004_H
Local RAM error status clear register	LRSTCLR	FFC6 3024_H
Local RAM error overflow status register	LROVFSTR	FFC6 3028_H
Local RAM error overflow status clear register	LROVFSTC	FFC6 $302C_H$
Local RAM first error address register 0	LRFSTEADR0	FFC6 3030_H
Local RAM error notification control register	LRERRINT	FFC6 $30B0_H$
Local RAM test control register	LRTSTCTL	FFC6 $30B4_H$
Local RAM test data read buffer 0	LRTDATBF0	FFC6 $30B8_H$

36.4.2 LRECCCTL — Local RAM ECC Control Register

The LRECCCTL register specifies whether to enable or disable ECC error detection/correction and whether to enable or disable 1-bit error correction. When writing to the LRECCCTL register, PROT1 and PROT0 must be 01_B.

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

Address: FFC6 3000_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	—	—	—	—	—	—	—	—	—	—	—	—	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 36.4 LRECCCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	These two bits specify whether updating the ECCDIS and SECDIS bits 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 writing to the LRECCCTL register, (PROT1,PROT0) must be (0,1).
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	SECDIS	1-Bit Error Correction Disable This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. When writing to this bit, (0,1) should be written to (PROT1,PROT0) at the same time. The initial setting is that the 1-bit error 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.
0	ECCDIS	ECC Disable This bit specifies whether to enable or disable the ECC error detection/correction function. When writing to this bit, (0,1) should be written to (PROT1,PROT0) at the same time. The initial setting is that the ECC error detection/correction function is enabled. 0: ECC error detection/correction function is enabled. 1: ECC error detection/correction function is disabled. Even when the error detection/correction function is disabled, the encode function is still enabled.

36.4.3 LRFSTERSTR — Local RAM First Error Status Register

LRFSTERSTR is a register for monitoring the first error.

If the ECC error detection/correction is enabled, the SEDF0 bit is set when an ECC 1-bit error is detected, and the DEDF0 bit is set when an ECC 2-bit error is detected.

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

Address: FFC6 3004$_H$

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 36.5 LRFSTERSTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	DEDF0	Local RAM ECC 2-Bit Error Monitor Flag This bit is set when an ECC 2-bit error occurs regardless of the SEDF0 and DEDF0 settings. • Clearing condition: Internal reset or pin reset occurs. The FSTERRCLR0 bit in the local RAM error status clear register is set. • Setting condition: An ECC 2-bit error occurs.
0	SEDF0	Local RAM ECC 1-Bit Error Monitor Flag This bit is set when an ECC 1-bit error occurs while both SEDF0 and DEDF0 are 0. • Clearing condition: Internal reset or pin reset occurs. The FSTERRCLR0 bit in the local RAM error status clear register is set. • Setting condition: An ECC 1-bit error occurs while both SEDF0 and DEDF0 are 0.

36.4.4 LRSTCLR — Local RAM Error Status Clear Register

LRSTCLR is a register used for clearing the error flag in the local RAM error status register.

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

Address: FFC6 3024_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	FSTERR CLR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.6 LRSTCLR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	FSTERRCLR0	First Error Status Register Clear Write 1 to this bit to clear the error flag in the first error status register.

36.4.5 LROVFSTR — Local RAM Error Overflow Status Register

LROVFSTR is a register used for monitoring the occurrence of local RAM error overflow. The ERROVF0 flag is cleared by setting 1 to the ERROVFCLR0 bit in the LROVFSTC register.

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

Address: FFC6 3028_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVF0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 36.7 LROVFSTR Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	ERROVF0	Error Overflow Flag This bit is set when an ECC error occurs while error address registers are enabled. • When the retained address source is SEDF 1-bit error occurs at the same address → Overflow does not occur 1-bit error occurs at a different address → Overflow occurs 2-bit error occurs → Overflow occurs • When the retained address source is DEDF 1-bit error occurs → Overflow occurs 2-bit error occurs at the same address → Overflow does not occur 2-bit error occurs at a different address → Overflow occurs

36.4.6 LROVFSTC — Local RAM Error Overflow Status Clear Register

LROVFSTC is a register used for clearing the local RAM error overflow flag.

The error overflow flag is cleared when the ERROVFCLR0 bit is set to 1.

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

Address: FFC6 302C$_H$

Value after reset: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ERROVF CLR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 36.8 LROVFSTC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	ERROVFCLR0	Error Overflow Flag Clear Write 1 to this bit to clear the ERROVF0 flag. This bit is always read as 0.

36.4.7 LRFSTEADR0 — Local RAM First Error Address Register 0

LRFSTEADR0 retains the address of an ECC error occurrence. This register retains an internal address. Add the base address FEC0 0000_H to convert it to the actual address.

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

Address: FFC6 3030_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	LRFSTEADR0[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
	LRFSTEADR0[15:2]														—	—
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 36.9 LRFSTEADR0 Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 2	LRFSTEADR0 [21:2]	Address of the First Error Occurrence These bits are read only and used to monitor the address of the first error occurrence. • When the retained address source is SEDF 1-bit error occurs at the same address → The address is not overwritten. 1-bit error occurs at a different address → The address is not overwritten. 2-bit error occurs →The address is overwritten • When the retained address source is DEDF 1-bit error occurs →The address is not overwritten. 2-bit error occurs at the same address →The address is not overwritten. 2-bit error occurs at a different address →The address is not overwritten.
1, 0	Reserved	When read, the value after reset is returned.

36.4.8 LRERRINT — Local RAM Error Notification Control Register

The LRERRINT register is used for specifying whether to enable or disable error notification signals generated when an ECC 1-bit error is detected.

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

Address: FFC6 30B0$_H$

Value after reset: 02$_H$

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

Table 36.10 LRERRINT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	SEDIE	ECC 1-Bit Error Notification Enable This bit controls whether to notify an error when a 1-bit error is detected while ECC error detection/correction is enabled. 0: ECC 1-bit error notification is disabled. 1: ECC 1-bit error notification is enabled.

36.4.9 LRTSTCTL — Local RAM Test Control Register

This register is used for an ECC test (self diagnosis). After the ECC test mode is enabled by setting ECCTST = 1, arbitrary data can be written to the ECC bits. The DATSEL bit is used to select the RAM data or the ECC bits.

When writing to the LRTSTCTL register, PROT1 and PROT0 must be 01_B.

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

Address: FFC6 $30B4_H$

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	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 36.11 LRTSTCTL Register Contents

Bit Position	Bit Name	Function
15	PROT1	These two bits specify whether updating the ECCTST and DATSEL bits 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 writing to the LRTSTCTL register, (PROT1, PROT0) must be (0,1).
14	PROT0	
13 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	ECCTST	ECC Test Mode This bit specifies the ECC test mode. 0: Normal mode 1: ECC test mode
0	DATSEL	Data Selection This bit is enabled when ECCTST = 1. This bit specifies the RAM bit which can be accessed when writing. 0: RAM data is selected. 1: ECC bits are selected.

36.4.10 LRTDATBF0 — Local RAM Test Data Read Buffer 0

In test mode (ECCTST = 1), the ECC bits can be read. While ECCTST = 1 in the local RAM test control register, reading the RAM reads out the data of the ECC bits stored in the RAM, and the data is stored in this buffer.

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

Address: FFC6 30B8$_H$

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
	—	—	—	—	—	—	—	—	—	LRTDATBF0[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 36.12 LRTDATBF0 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	When read, the value after reset is returned.
6 to 0	LRTDATBF0 [6:0]	These bits are enabled when ECCTST = 1 (test mode) in the local RAM test control register. When reading the RAM, the data of the ECC bits is stored in LRTDATBF06 to LRTDATBF00.

Section 37 Boundary Scan

This section contains a generic description of boundary scan.

The RH850/F1L has the JTAG interface and provides the boundary scan function.

37.1 Overview

Boundary scan is a test method defined in the IEEE standard 1149.1, which is used to test the connection between the devices mounted on the printed-circuit board. The boundary scan of the RH850/F1L conforms to IEEE Std 1149.1-2001.

37.2 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 six commands.

- BYPASS mode
  Test mode conforming to the IEEE 1149.1
- EXTEST mode
  Test mode conforming to the IEEE 1149.1
- SAMPLE/PRELOAD mode
  Test mode conforming to the IEEE 1149.1
- IDCODE mode
  Test mode conforming to the IEEE 1149.1

Figure 37.1 shows a block diagram of the JTAG interface.

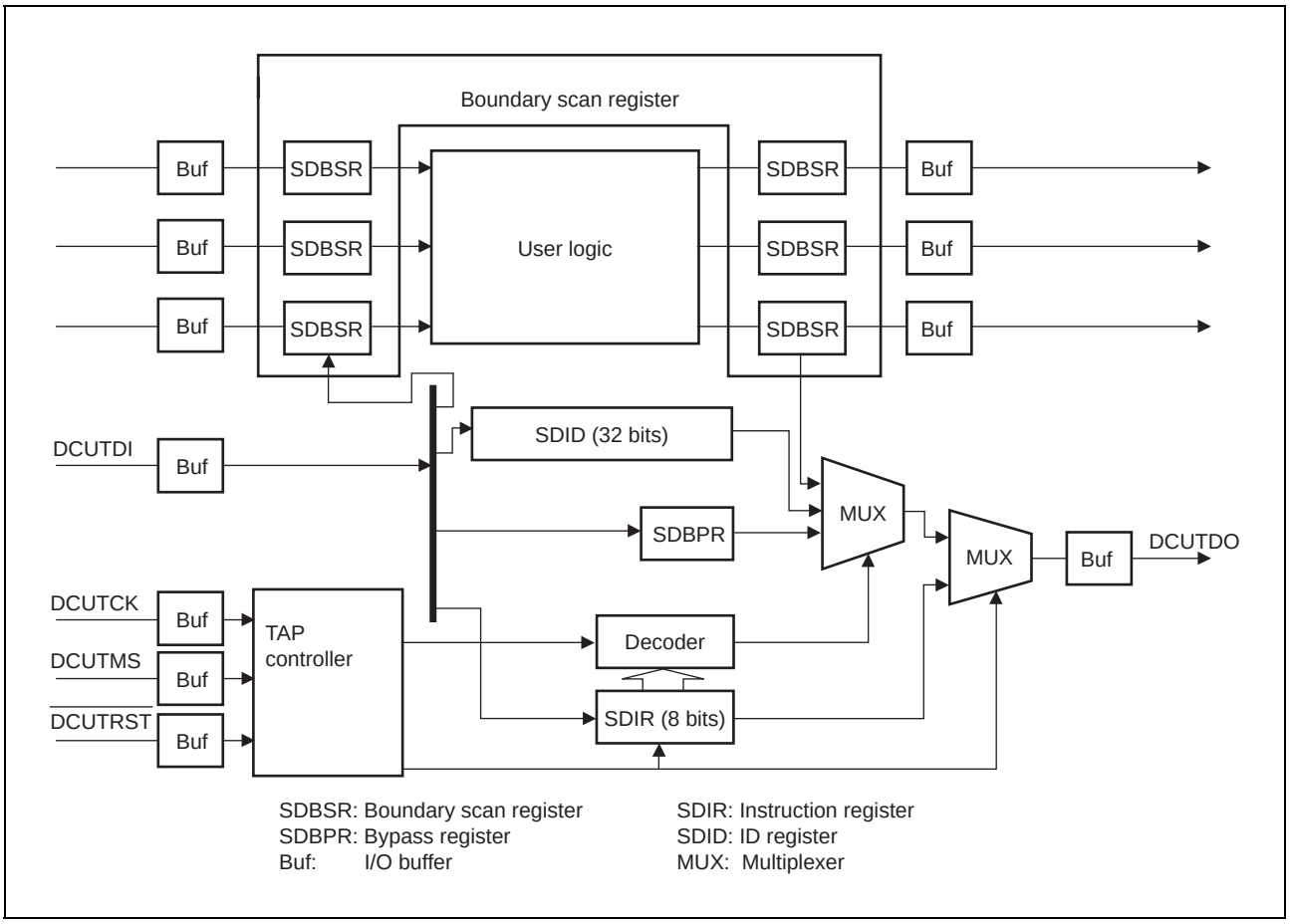


Figure 37.1 Block Diagram of JTAG Interface

37.3 External Input/Output Pins

There are five JTAG control signals: DCUTCK, DCUTDI, DCUTMS, DCUTDO, and $\overline{\text{DCUTRST}}$.

Table 37.1 shows the pin configuration.

Table 37.1 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 37.2, 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.

37.4 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 37.2 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.

Commands 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 37.3 shows the serial transfer types possible with the JTAG interface registers.

Table 37.3 Serial Transfer Types Possible with JTAG Interface Registers

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.

37.4.1 Instruction Register (SDIR)

SDIR is an 8-bit register that holds a boundary scan command. SDIR is initialized by a low-level input of $\overline{\text{DCUTRST}}$ or in the TAP Test-Logic-Reset state. Operation is not guaranteed when any reserved command is set in this register.

Table 37.4 Boundary Scan Commands

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

37.4.2 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 command is set, but cannot be accessed from the CPU.

For the read values, see **Table 37.2, Register Configuration**.

37.4.3 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 asserted $\overline{\text{DCUTRST}}$.

37.4.4 Boundary Scan Register (SDBSR)

SDBSR is a shift register for controlling the external I/O 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 $\overline{\text{DCUTRST}}$.

37.5 Operation

37.5.1 TAP Controller

Table 37.2 shows the state transition of the TAP controller. Transition is caused according to the DCUTMS value at the rising edge of DCUTCK.

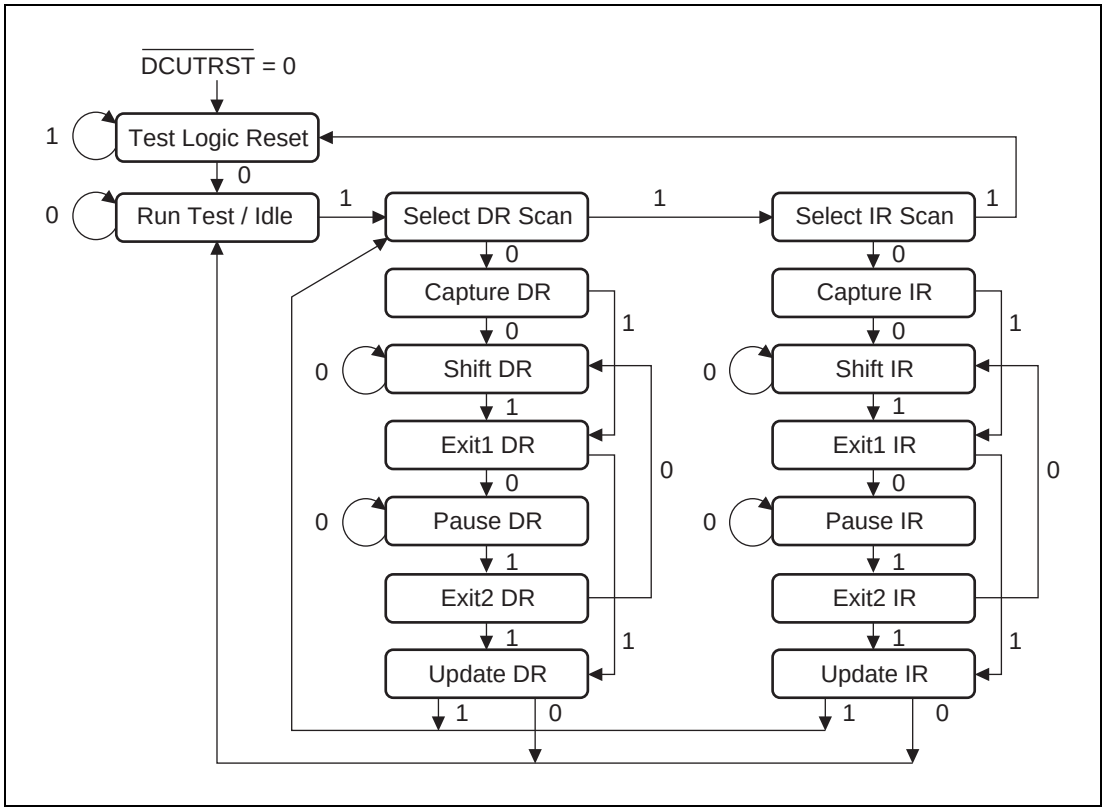


Figure 37.2 TAP Controller State Transition Diagram

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.

37.5.2 Supported Commands

37.5.2.1 BYPASS

The BYPASS command is a standard command indispensable to bypass register operation. This command shortens the shift path to achieve high-speed serial data transfer of other devices on the printed-circuit board. During execution of this command, the test circuit has no effect on the system circuit.

37.5.2.2 SAMPLE/PRELOAD

The SAMPLE/PRELOAD command 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 command, 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 command 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 command. If the EXTEST command is executed without PRELOAD operation, the 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 with the EXTEST command.

37.5.2.3 EXTEST

The EXTEST command is used to test the external circuits when this device is mounted on the printed-circuit board. When this command is executed, the output pin is used to output the test data (previously set with the SAMPLE/PRELOAD command) 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 command 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 command, it is not used for testing the external circuits (replaced through shift operation).

37.5.2.4 IDCODE

The IDCODE command 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.

37.5.3 Pins Subjected to Boundary Scan

All pins excluding such as external clock input pins or power supply pins are subjected to boundary scan.

The pins which are not subjected to boundary scan are listed in **Table 37.5**.

Table 37.5 Pins not Subjected to Boundary Scan

Type	Pins
JTAG interface	DCUTCK, DCUTDI, DCUTDO, DCUTMS, $\overline{\text{DCUTRST}}$
Power supply pins	REGVCC, AWOVCL, AWOVSS, ISOVCL, ISOVSS EVCC, BVCC*1, EVSS, BVSS*1
Power supply pins (A/D converter)	A0VREF, A1VREF*1, A0VSS, A1VSS*1
Clock signals	X1, X2, XT1*1, IP0_0/XT2*1

Note 1. Only available for 144-pin and 176-pin devices.

The following signals are only sampled in boundary scan mode.

Table 37.6 Pins Subjected to Boundary Scan (Only Sampling)

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 37.7 Pins Subjected to Boundary Scan (Only General I/O Pins)

Function	Pin Name
ADCA0 input	P8_0-12, P9_0-6, AP0_0-15
ADCA1 input	P18_0-7, 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

37.6 Usage Notes

1. Once a command is set, it is not modified until another command is issued again. To continuously issue the same commands, insert a command that has no effect on chip operation (such as BYPASS mode) between the desired commands.
2. To start the system in boundary scan mode, negate $\overline{\text{DCUTRST}}$ while $\overline{\text{RESET}}$ is high.
3. For the maximum clock frequency that can be input to DCUTCK, see the Electrical Characteristics section in the Data Sheet document.
4. If the number of serially transferred bits exceeds the number of bits of the register connected between DCUTDI and DCUTDO, the data that is input from DCUTDI is output from DCUTDO after the serial data equal to the number of register bits are output.
5. If the serial transfer sequence is corrupted, be sure to reset $\overline{\text{DCUTRST}}$. Here, start the transfer over again 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 38 Power Supply and Power Domains

This section describes the power supply and power domains of the RH850/F1L.

38.1 Function

The internal circuits are separated into two independent power domains, the Always-On area (AWO) and the Isolated area (ISO).

The Always-On area remains powered in all operating modes and standby mode.

The power supply of the Isolated area can be turned off depending on the standby mode to reduce the overall power consumption.

For each power domain, a separate 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 of its 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 separated I/O ports.

38.1.1 Power Supply Pins

The table below lists all power supply pins and what they are used for.

Table 38.1 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	(176-pin devices) • Port groups JP0, P0, P1, P2, P8, P9, P20 (144-pin devices) • Port groups JP0, P0, P1, P8, P9, P20 (80-pin and 100-pin devices) • Port groups JP0, P0, P8, P9, P10, P11 (48-pin and 64-pin devices) • Port groups JP0, P0, P8, P9, P10
	EVSS	
	BVCC	(144-pin and 176-pin devices) • 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: For the voltage range, see the Electrical Characteristics of the data sheet.

Note 1. Stabilized power supply

38.1.2 Block Diagram of Power Domains

The figure below shows the overview of power supply circuit.

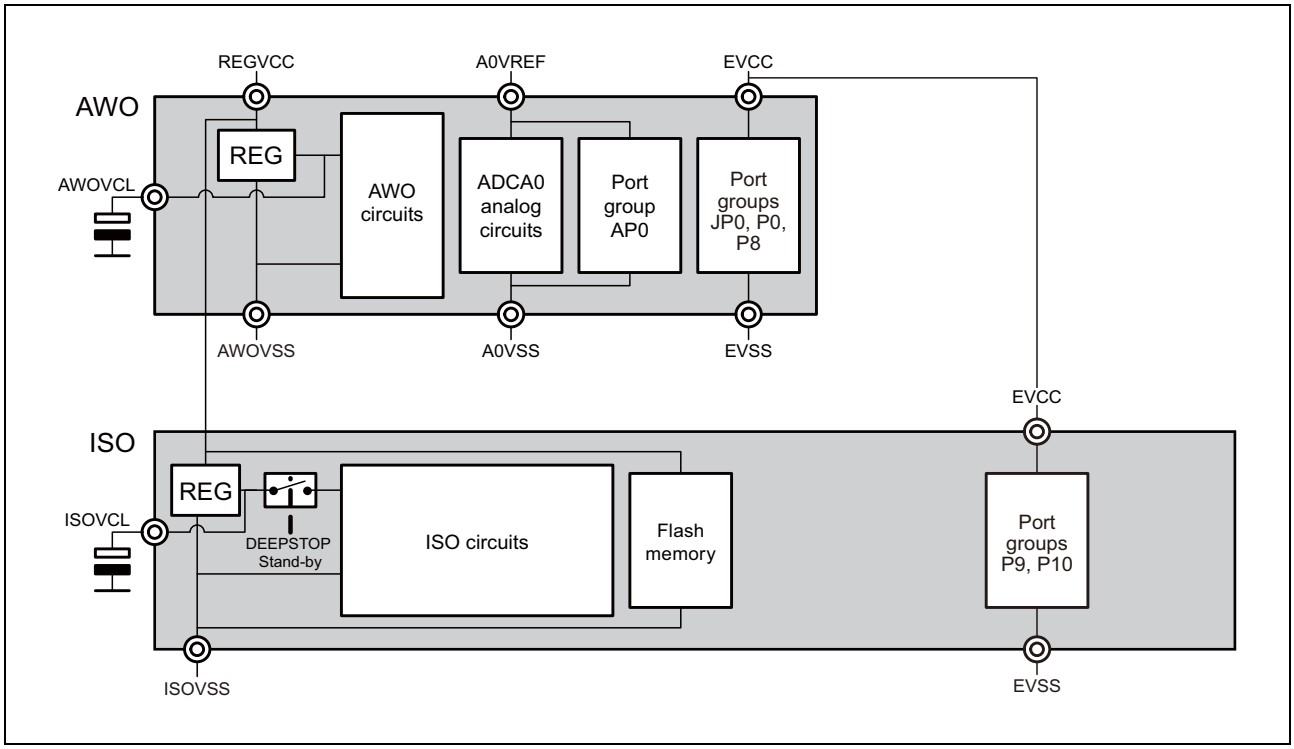


Figure 38.1 Overview of Power Supply Circuit (48-pin and 64-pin)

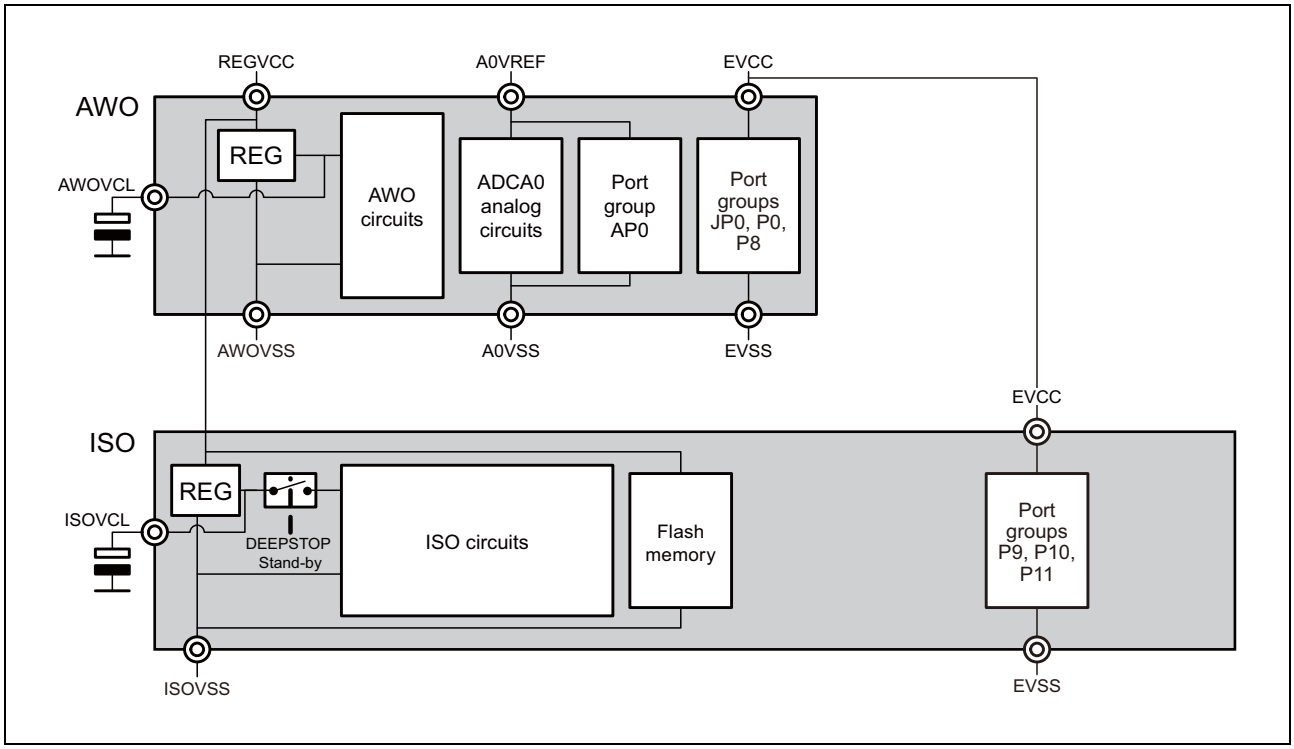


Figure 38.2 Overview of Power Supply Circuit (80-pin and 100-pin)

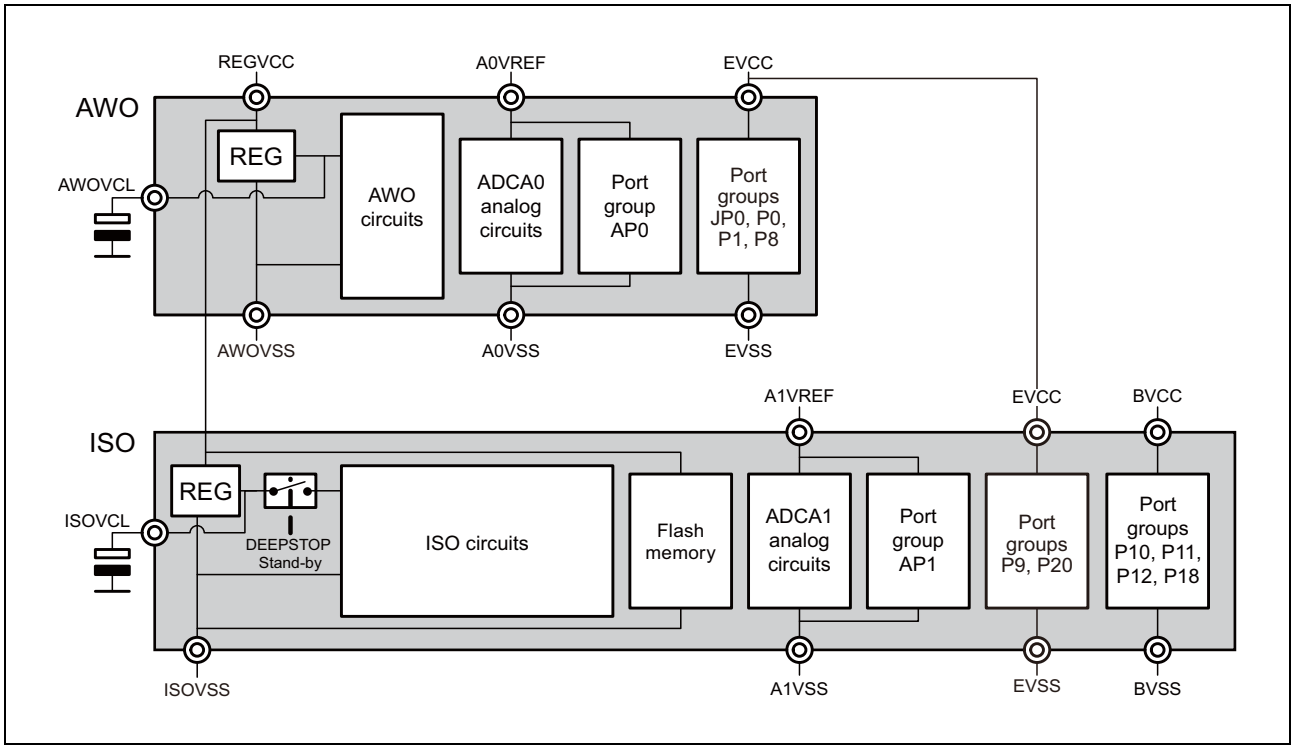


Figure 38.3 Overview of Power Supply Circuit (144-pin)

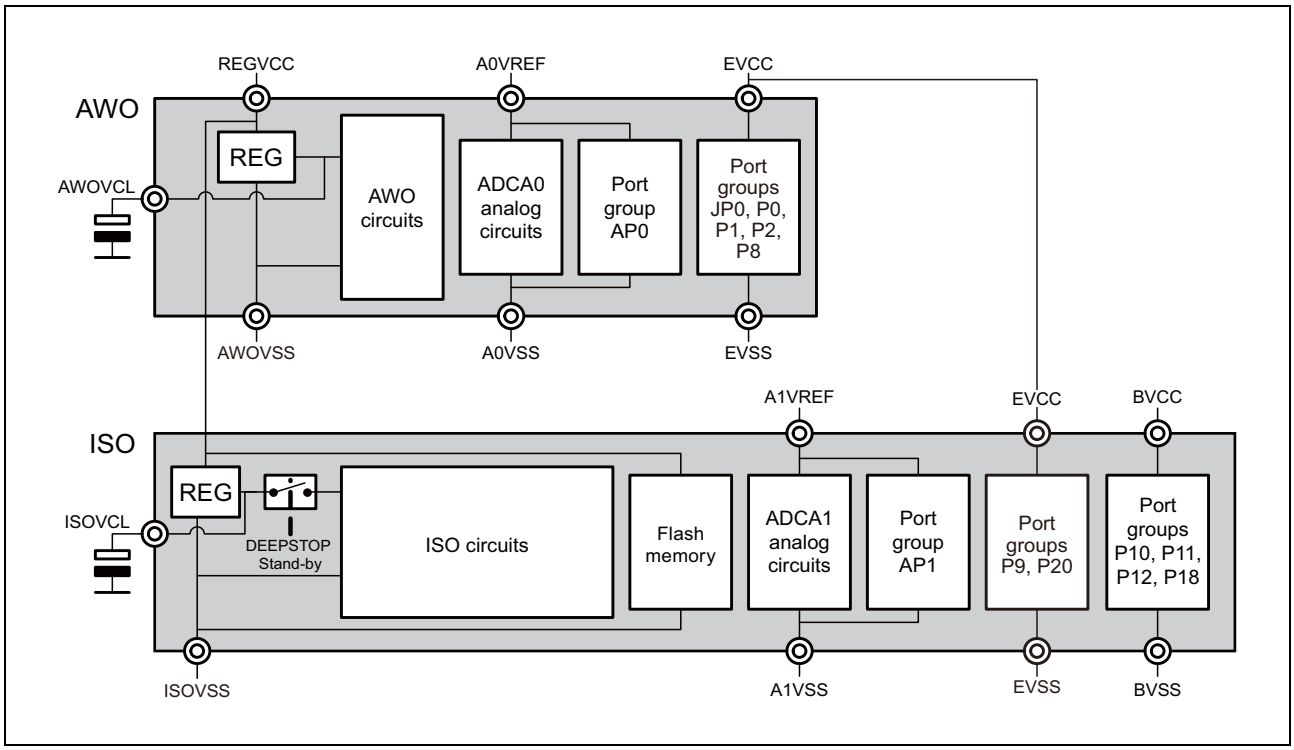


Figure 38.4 Overview of Power Supply Circuit (176-pin)

38.1.3 Power Domains Arrangement

The table below lists the microcontroller functional modules for each power domain.

Table 38.2 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, P8, AP0, IP0
ISO area	• CPU subsystem • Code flash, data flash, primary local RAM, secondary local RAM • PLL, CLMA2 • WDAT1, DCRAn, TAUDn, TAUBn, TAUJ1, OSTMn, PWM-Diag, CSIGn, CSIHn, RSCANn, RLIN2m, RLIN3n, RIICn, ADCA1, Motor Control, ENCAn, KRn • Port groups P9, P10, P11, P12, P18, P20, AP1

Appendix A. Pin List

Table A.1 Pin List (1/4)

Pin Name	IO	Function	Input					Output	Other				Device					
			CMOS	SHMT1	SHMT2	SHMT4	Analog	Freq. [MHz]	Pull-up	Pull-down	Hi-Z control during reset	I/O hold control in DeepSTOP mode	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
$\overline{\text{RESET}}$	I	System control			√								√	√	√	√	√	√
A0VREF	—	Power for ADCA0											√	√	√	√	√	√
A0VSS	—	GND for ADCA0											√	√	√	√	√	√
A1VREF	—	Power for ADCA1															√	√
A1VSS	—	GND for ADCA1															√	√
AP0_0	IO	Analog port (AP0)	√				√	Low		*4	√		√	√	√	√	√	√
AP0_1	IO	Analog port (AP0)	√				√	Low		*4	√		√	√	√	√	√	√
AP0_2	IO	Analog port (AP0)	√				√	Low		*4	√		√	√	√	√	√	√
AP0_3	IO	Analog port (AP0)	√				√	Low		*4	√		√	√	√	√	√	√
AP0_4	IO	Analog port (AP0)	√				√	Low		*4	√		√	√	√	√	√	√
AP0_5	IO	Analog port (AP0)	√				√	Low		*4	√		√	√	√	√	√	√
AP0_6	IO	Analog port (AP0)	√				√	Low		*4	√		√	√	√	√	√	√
AP0_7	IO	Analog port (AP0)	√				√	Low		*4	√		√	√	√	√	√	√
AP0_8	IO	Analog port (AP0)	√				√	Low		*4	√			√	√	√	√	√
AP0_9	IO	Analog port (AP0)	√				√	Low		*4	√			√	√	√	√	√
AP0_10	IO	Analog port (AP0)	√				√	Low		*4	√				√	√	√	√
AP0_11	IO	Analog port (AP0)	√				√	Low		*4	√					√	√	√
AP0_12	IO	Analog port (AP0)	√				√	Low		*4	√					√	√	√
AP0_13	IO	Analog port (AP0)	√				√	Low		*4	√					√	√	√
AP0_14	IO	Analog port (AP0)	√				√	Low		*4	√					√	√	√
AP0_15	IO	Analog port (AP0)	√				√	Low		*4	√					√	√	√
AP1_0	IO	Analog port (AP1)	√				√	Low		*4	√	√					√	√
AP1_1	IO	Analog port (AP1)	√				√	Low		*4	√	√					√	√
AP1_2	IO	Analog port (AP1)	√				√	Low		*4	√	√					√	√
AP1_3	IO	Analog port (AP1)	√				√	Low		*4	√	√					√	√
AP1_4	IO	Analog port (AP1)	√				√	Low		*4	√	√					√	√
AP1_5	IO	Analog port (AP1)	√				√	Low		*4	√	√					√	√
AP1_6	IO	Analog port (AP1)	√				√	Low		*4	√	√					√	√
AP1_7	IO	Analog port (AP1)	Ö				√	Low		*4	√	√					√	√
AP1_8	IO	Analog port (AP1)	√				√	Low		*4	√	√						√
AP1_9	IO	Analog port (AP1)	√				√	Low		*4	√	√						√
AP1_10	IO	Analog port (AP1)	√				√	Low		*4	√	√						√
AP1_11	IO	Analog port (AP1)	√				√	Low		*4	√	√						√
AP1_12	IO	Analog port (AP1)	√				√	Low		*4	√	√						√
AP1_13	IO	Analog port (AP1)	√				√	Low		*4	√	√						√
AP1_14	IO	Analog port (AP1)	√				√	Low		*4	√	√						√
AP1_15	IO	Analog port (AP1)	√				√	Low		*4	√	√						√
AWOVCL	—	Power for AWO area											√	√	√	√	√	√
AWOVSS	—	GND for AWO area											√	√	√	√	√	√
BVDD	—	Power for IO															√	√
BVSS	—	GND for IO															√	√
EVCC	—	Power for IO											√	√	√	√	√	√

Table A.1 Pin List (2/4)

Pin Name	IO	Function	Input					Output	Other				Device					
			CMOS	SHMT1	SHMT2	SHMT4	Analog	Freq. [MHz]	Pull-up	Pull-down	Hi-Z control during reset	I/O hold control in DeepSTOP mode	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
EVSS	—	GND for IO											√	√	√	√	√	√
FLMD0	I	System control		√					√*5	√*5			√	√	√	√	√	√
IP0_0	I	IP0/SOSC	*8														√	√
ISOVCL	—	Power for ISO area											√	√	√	√	√	√
ISOVSS	—	GND for ISO area											√	√	√	√	√	√
JP0_0	IO	JTAG port (JP0)				√		Low	√	√	√*1		√	√	√	√	√	√
JP0_1	IO	JTAG port (JP0)				√		Low	√	√	√*1		√	√	√	√	√	√
JP0_2	IO	JTAG port (JP0)				√		Low	√	√	√*1		√	√	√	√	√	√
JP0_3	IO	JTAG port (JP0)				√		Low	√	√	√*1		√	√	√	√	√	√
JP0_4	IO	JTAG port (JP0)				√		Low	√	√	√*1		√	√	√	√	√	√
JP0_5	IO	JTAG port (JP0)				√		Low	√	√	√*1		√	√	√	√	√	√
JP0_6	IO	JTAG port (JP0)				√		Low	√	√	√*1						√	√
P0_0	IO	Digital port (P0)		√		√		Low	√	√	*2		√	√	√	√	√	√
P0_1	IO	Digital port (P0)		√		√		Low	√	√	√		√	√	√	√	√	√
P0_2	IO	Digital port (P0)		√		√		Low / Fast *7	√	√	√		√	√	√	√	√	√
P0_3	IO	Digital port (P0)		√		√		Low / Fast *7	√	√	√		√	√	√	√	√	√
P0_4	IO	Digital port (P0)		√		√		Low	√	√	√			√	√	√	√	√
P0_5	IO	Digital port (P0)		√		√		Low / Fast	√	√	√			√	√	√	√	√
P0_6	IO	Digital port (P0)		√		√		Low / Fast	√	√	√			√	√	√	√	√
P0_7	IO	Digital port (P0)		√		√		Low / Fast	√	√	√				√	√	√	√
P0_8	IO	Digital port (P0)				√		Low	√	√	√				√	√	√	√
P0_9	IO	Digital port (P0)		√		√		Low	√	√	√				√	√	√	√
P0_10	IO	Digital port (P0)				√		Low	√	√	√				√	√	√	√
P0_11	IO	Digital port (P0)		√		√		Low	√	√	√				√	√	√	√
P0_12	IO	Digital port (P0)		√		√		Low	√	√	√				√	√	√	√
P0_13	IO	Digital port (P0)		√		√		Low / Fast	√		√					√	√	√
P0_14	IO	Digital port (P0)				√		Low / Fast	√		√					√	√	√
P1_0	IO	Digital port (P1)		√		√		Low	√		√						√	√
P1_1	IO	Digital port (P1)				√		Low	√		√						√	√
P1_2	IO	Digital port (P1)		√		√		Low	√		√						√	√
P1_3	IO	Digital port (P1)				√		Low	√		√						√	√
P1_4	IO	Digital port (P1)		√		√		Low	√		√						√	√
P1_5	IO	Digital port (P1)				√		Low	√		√						√	√
P1_6	IO	Digital port (P1)		√		√		Low	√		√						√	√
P1_7	IO	Digital port (P1)				√		Low	√		√						√	√
P1_8	IO	Digital port (P1)		√		√		Low	√		√						√	√
P1_9	IO	Digital port (P1)				√		Low	√		√						√	√
P1_10	IO	Digital port (P1)		√		√		Low	√		√						√	√
P1_11	IO	Digital port (P1)				√		Low	√		√						√	√
P1_12	IO	Digital port (P1)		√		√		Low	√		√							√
P1_13	IO	Digital port (P1)				√		Low	√		√							√
P1_14	IO	Digital port (P1)		√		√		Low	√		√							√
P1_15	IO	Digital port (P1)				√		Low	√		√							√
P2_0	IO	Digital port (P2)		√		√		Low	√		√							√
P2_1	IO	Digital port (P2)				√		Low	√		√							√

Table A.1 Pin List (3/4)

Pin Name	IO	Function	Input					Output	Other				Device					
			CMOS	SHMT1	SHMT2	SHMT4	Analog	Freq. [MHz]	Pull-up	Pull-down	Hi-Z control during reset	I/O hold control in DeepSTOP mode	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P2_2	IO	Digital port (P2)		√		√		Low	√		√							√
P2_3	IO	Digital port (P2)				√		Low	√		√							√
P2_4	IO	Digital port (P2)		√		√		Low	√		√							√
P2_5	IO	Digital port (P2)				√		Low	√		√							√
P2_6	IO	Digital port (P2)				√		Low	√		√							√
P8_0	IO	Digital port (P8)				√	√	Low	√	√*4	√		√	√	√	√	√	√
P8_1	IO	Digital port (P8)				√	√	Low	√	√*4	√		√	√	√	√	√	√
P8_2	IO	Digital port (P8)				√	√	Low	√	√*4	√			√	√	√	√	√
P8_3	IO	Digital port (P8)				√	√	Low	√	√*4	√			√	√	√	√	√
P8_4	IO	Digital port (P8)				√	√	Low	√	√*4	√			√	√	√	√	√
P8_5	IO	Digital port (P8)				√	√	Low	√	√*4	√			√	√	√	√	√
P8_6	IO	Digital port (P8)				√	√	Low	√	√*4	√			√	√	√	√	√
P8_7	IO	Digital port (P8)				√	√	Low	√	*4	√					√	√	√
P8_8	IO	Digital port (P8)				√	√	Low	√	*4	√					√	√	√
P8_9	IO	Digital port (P8)				√	√	Low	√	*4	√					√	√	√
P8_10	IO	Digital port (P8)				√	√	Low	√	*4	√					√	√	√
P8_11	IO	Digital port (P8)				√	√	Low	√	*4	√					√	√	√
P8_12	IO	Digital port (P8)				√	√	Low	√	*4	√					√	√	√
P9_0	IO	Digital port (P9)				√	√	Low	√	√*4	√	√	√	√	√	√	√	√
P9_1	IO	Digital port (P9)				√	√	Low	√	√*4	√	√	√	√	√	√	√	√
P9_2	IO	Digital port (P9)				√	√	Low	√	√*4	√	√		√	√	√	√	√
P9_3	IO	Digital port (P9)				√	√	Low	√	√*4	√	√		√	√	√	√	√
P9_4	IO	Digital port (P9)				√	√	Low	√	√*4	√	√			√	√	√	√
P9_5	IO	Digital port (P9)				√	√	Low	√	√*4	√	√			√	√	√	√
P9_6	IO	Digital port (P9)				√	√	Low	√	√*4	√	√			√	√	√	√
P10_0	IO	Digital port (P10)		√		√		Low / Fast	√	√	√	√	√	√	√	√	√	√
P10_1	IO	Digital port (P10)				√		Low / Fast	√	√	*6	√	√	√	√	√	√	√
P10_2	IO	Digital port (P10)		√		√		Low / Fast	√	√	*6	√	√	√	√	√	√	√
P10_3	IO	Digital port (P10)		√		√		Low / Fast	√	√	√	√	√	√	√	√	√	√
P10_4	IO	Digital port (P10)		√		√		Low / Fast	√	√	√	√	√	√	√	√	√	√
P10_5	IO	Digital port (P10)				√		Low / Fast	√	√	√	√	√	√	√	√	√	√
P10_6	IO	Digital port (P10)		√		√		Low / Fast	√	√	√	√	√	√	√	√	√	√
P10_7	IO	Digital port (P10)				√		Low / Fast	√	√	√	√	√	√	√	√	√	√
P10_8	IO	Digital port (P10)				√		Low / Fast	√	√	*3	√	√	√	√	√	√	√
P10_9	IO	Digital port (P10)		√		√		Low / Fast	√	√	√	√	√	√	√	√	√	√
P10_10	IO	Digital port (P10)				√		Low / Fast	√	√	√	√	√	√	√	√	√	√
P10_11	IO	Digital port (P10)		√		√		Low / Fast	√	√	√	√		√	√	√	√	√
P10_12	IO	Digital port (P10)				√		Low / Fast	√	√	√	√		√	√	√	√	√
P10_13	IO	Digital port (P10)		√		√		Low / Fast	√	√	√	√		√	√	√	√	√
P10_14	IO	Digital port (P10)				√		Low / Fast	√	√	√	√		√	√	√	√	√
P10_15	IO	Digital port (P10)		√		√		Low / Fast	√	√	√	√			√	√	√	√
P11_0	IO	Digital port (P11)				√		Low / Fast	√	√	√	√			√	√	√	√
P11_1	IO	Digital port (P11)		√		√		Low / Fast	√	√	√	√			√	√	√	√
P11_2	IO	Digital port (P11)				√		Low / Fast	√	√	√	√			√	√	√	√
P11_3	IO	Digital port (P11)		√		√		Low / Fast	√	√	√	√			√	√	√	√

Table A.1 Pin List (4/4)

Pin Name	IO	Function	Input					Output	Other				Device					
			CMOS	SHMT1	SHMT2	SHMT4	Analog	Freq. [MHz]	Pull-up	Pull-down	Hi-Z control during reset	I/O hold control in DeepSTOP mode	48 pins	64 pins	80 pins	100 pins	144 pins	176 pins
P11_4	IO	Digital port (P11)				√		Low / Fast	√	√	√	√			√	√	√	√
P11_5	IO	Digital port (P11)		√		√		Low / Fast	√		√	√				√	√	√
P11_6	IO	Digital port (P11)		√		√		Low / Fast	√		√	√				√	√	√
P11_7	IO	Digital port (P11)				√		Low / Fast	√		√	√				√	√	√
P11_8	IO	Digital port (P11)				√		Low / Fast	√		√	√					√	√
P11_9	IO	Digital port (P11)		√		√		Low / Fast	√		√	√					√	√
P11_10	IO	Digital port (P11)				√		Low / Fast	√		√	√					√	√
P11_11	IO	Digital port (P11)				√		Low / Fast	√		√	√					√	√
P11_12	IO	Digital port (P11)		√		√		Low	√		√	√					√	√
P11_13	IO	Digital port (P11)		√		√		Low / Fast	√		√	√					√	√
P11_14	IO	Digital port (P11)				√		Low / Fast	√		√	√					√	√
P11_15	IO	Digital port (P11)		√		√		Low / Fast	√		√	√					√	√
P12_0	IO	Digital port (P12)				√		Low / Fast	√		√	√					√	√
P12_1	IO	Digital port (P12)		√		√		Low / Fast	√		√	√					√	√
P12_2	IO	Digital port (P12)				√		Low / Fast	√		√	√					√	√
P12_3	IO	Digital port (P12)		√		√		Low	√		√	√						√
P12_4	IO	Digital port (P12)				√		Low	√		√	√						√
P12_5	IO	Digital port (P12)				√		Low	√		√	√						√
P18_0	IO	Digital port (P18)				√	√	Low	√	*4	√	√					√	√
P18_1	IO	Digital port (P18)				√	√	Low	√	*4	√	√					√	√
P18_2	IO	Digital port (P18)				√	√	Low	√	*4	√	√					√	√
P18_3	IO	Digital port (P18)				√	√	Low	√	*4	√	√					√	√
P18_4	IO	Digital port (P18)				√	√	Low	√	*4	√	√						√
P18_5	IO	Digital port (P18)				√	√	Low	√	*4	√	√						√
P18_6	IO	Digital port (P18)				√	√	Low	√	*4	√	√						√
P18_7	IO	Digital port (P18)				√	√	Low	√	*4	√	√						√
P20_0	IO	Digital port (P20)		√		√		Low	√		√	√						√
P20_1	IO	Digital port (P20)				√		Low	√		√	√						√
P20_2	IO	Digital port (P20)		√		√		Low	√		√	√						√
P20_3	IO	Digital port (P20)				√		Low	√		√	√						√
P20_4	IO	Digital port (P20)		√		√		Low	√		√	√					√	√
P20_5	IO	Digital port (P20)				√		Low	√		√	√					√	√
REGVCC	—	Power for VREG												√	√	√	√	√
X1	—	MOSC												√	√	√	√	√
X2	—	MOSC												√	√	√	√	√
XT1	—	SOSC															√	√

Note 1. During on-chip debug and LPD mode, this pin does not become Hi-Z.
Note 2. Low level output. This pin does not become Hi-Z.
Note 3. During serial programming mode and boundary scan mode, this pin is used as FLMD1 mode pin.
Note 4. Pull-down resistor for ADC diagnostic purpose. Control via ADC self-diagnostic register.
Note 5. Pull-up/Pull-down resistor for flash memory control. Control via FLMCNT register.
Note 6. Hi-Z state is not used in boundary scan mode.
Note 7. Set fast mode if the load capacitance of CSIH is 100 pF.
Note 8. See data sheet for input characteristics of PI0_0.

RH850/F1L User's Manual: Hardware

Publication Date: Rev.0.11 Oct 05, 2012
Rev.1.10 Jun 23, 2014

Published by: Renesas Electronics Corporation

Renesas Electronics Corporation

http://www.renesas.com

SALES OFFICES

Refer to "http://www.renesas.com/" for the latest and detailed information.

California Eastern Laboratories, Inc.
4590 Patrick Henry Drive, Santa Clara, California 95054, U.S.A.
Tel: +1-408-919-2500, Fax: +1-408-988-0279

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 D üsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, 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 Rd., Putuo District, Shanghai, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, 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 906, 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 Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Gu, Seoul, 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 2.0

RH850/F1L

Renesas Electronics Corporation

R01UH0390EJ0110